Simple Combinational Circuits

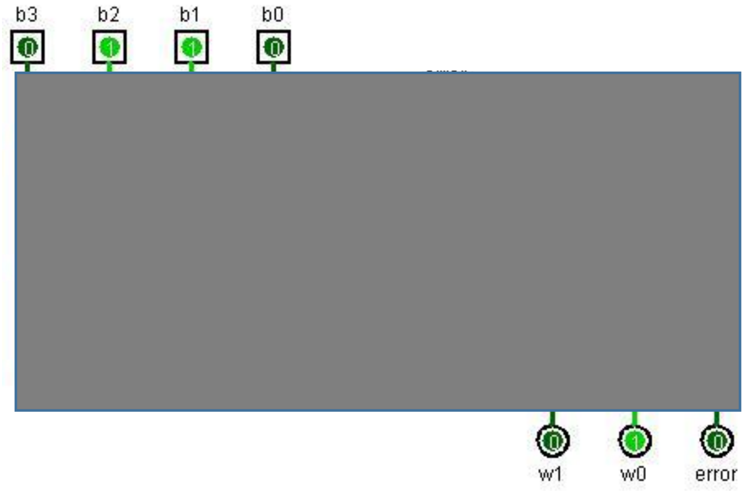
For this project, you will be using Logisim 2.7.1[1] to create some simple combinational circuits. You should familiarize yourself with Logisim by working your way through the built-in tutorial. You might want to review the CS 2505 notes on data representation before starting to design your solution.

Q1: Finding Low Set Bit in Nybble [32 points]

Create a simple circuit that returns the position of the lowest-order set bit (1) in a given nybble.

The bits of a nybble are indexed from low- to high-significance as: $b_3\ b_2\ b_1\ b_0$. So, for the nybble 1010, your circuit must return the value 1. But, what about the nybble 0000? There's no valid index to be returned, so you must add an extra output to signal the user than an error has occurred.

The circuit will have 4 1-bit input pins and 3 1-bit output pins (in order to take advantage of Logisim's ability to generate a truth table). The interface must be laid out as shown below:



In this example, the values of w1 and w0 indicate that the lowest set bit is in position 01, which is correct.

Note: you may only use elements from the Wiring and Gates Libraries in Logisim. You should probably consider creating subcircuits rather than a single, monolithic solution.

You must also create a plain text file showing your analysis that led to your circuit design; most likely this will consist of a truth table, and the associated Boolean expressions. You are not required to simplify the Boolean expressions, but doing so will make implementing the circuit easier (the reference solution uses a total of 14 logic gates and 3 subcircuits).

The correctness of your solution will be determined by using Logisim to generate a truth table (16 rows), and comparing that to the truth table produced from the reference solution. If you don't lay out the input and output pins as shown, then Logisim may generate the truth table columns in a different order, and you will lose points. In Logisim, go to Project/Analyze Circuit to generate the truth table. The column headings must be (order matters):

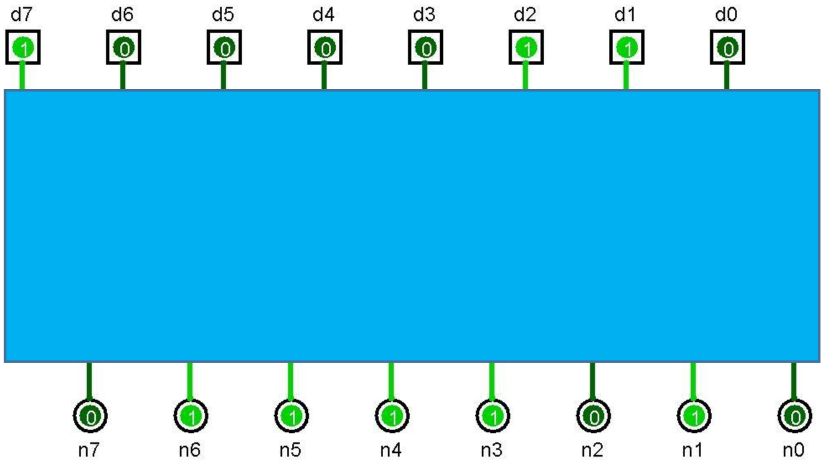
```
   b3  b2  b1  b0  |  w1  w0  error
```

If not, your circuit interface is incorrect.

Save your implementation of the circuit in a Logisim file named `lowOne.circ`, and name your text file `lowOne.txt`. The use of other names for the files will irritate the person who grades your solution, and you may be charged for that irritation. The grading will allocate 24 points for your circuit and 8 points for your analysis.

Q2: Absolute Value Circuit for int8_t [36 points]

Create a circuit that computes the absolute value of signed 8-bit integers, represented in 2's complement form. The interface must be laid out as shown below:



The input here represents the value -122_{10}, and the output represents the value 122_{10}.

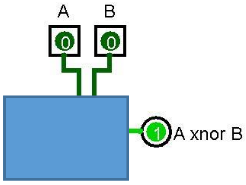
You may only use elements of the Wiring and Gates Logisim Libraries, and multiplexors. In this case, you should consider how to accomplish the computation from a high level; creating a truth table (256 rows!) and Boolean expressions would be unproductive.

The correctness of your solution will be determined by using Logisim to generate a truth table (256 rows), and comparing that to the truth table produced from the reference solution. If you don't lay out the input and output pins as shown, then Logisim may generate the truth table rows in a different order, and you will lose points.

Save your implementation of the circuit in a Logisim file named `absval8.circ`. The use of other names for the files will irritate the person who grades your solution, and you may be charged for that irritation.

Q3: XNOR Gate via Transistors [32 points]

Create an XNOR gate by using transistors, pull resistors, and any other elements in Logisim's Wiring library. You may not use any Logisim components from the other libraries. The interface must be laid out as shown below:

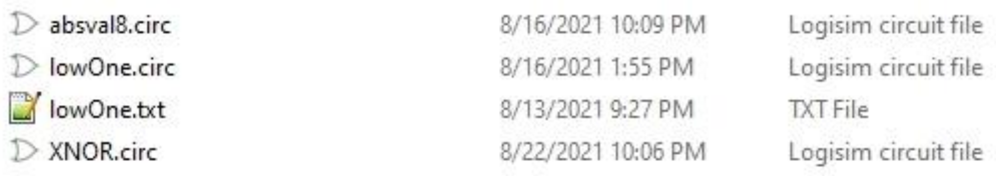


The correctness of your circuit will be determined by using Logisim to generate a truth table (4 rows), and comparing that to the truth table for the XNOR gate. If you don't lay out the input and output pins as shown, then Logisim may generate the truth table rows in a different order, and you will lose points.

Save your implementation of the circuit in a Logisim file named `XNOR.circ`. The use of another name for the file will irritate the person who grades your solution, and you may be charged for that irritation.

What to submit

You will submit a single, uncompressed, flat zip file containing your text file and your three Logisim files. When you display the contents of the file, it should look something like this (with NO directory information):



absval8.circ	8/16/2021 10:09 PM	Logisim circuit file
lowOne.circ	8/16/2021 1:55 PM	Logisim circuit file
lowOne.txt	8/13/2021 9:27 PM	TXT File
XNOR.circ	8/22/2021 10:06 PM	Logisim circuit file

If you worked with a partner, **only one of you should make a submission**, and be sure the text file contains a copy of the Partners Template (see the Assignments page), including the PID and name of each partner. Otherwise, only the partner who made the submission will receive credit.

The *Student Guide* and other pertinent information, such as the link to the proper submit page, will be found at:

`http://www.cs.vt.edu/curator/`

Notes

[1] You can download Logisim from the SourceForge project page at:

https://sourceforge.net/projects/circuit/

I recommend going to the Files page, then to the folder for 2.7.x/2.7.1, and then select the file

```
logisim-generic-2.7.1.jar
```

which you can run by double-clicking on it, or using the command "`java -jar logisim-generic-2.7.1.jar`" from a command line (both assuming you've properly installed Java on your machine). This is the version we will use for grading.

Windows users can try the file `logisim-win-2.7.1.exe`, but I found it complained unless I had a terribly out-of-date version of Java installed.

Do not make the error of using Logisim-evolution, which is a somewhat incompatible extension of Logisim.