

COMP 273 Assignment 1

Prepared by Prof. M. Langer and T.A.s Nurmemet (Noor) Abudukelimu and Josh Romoff.

Posted: Mon. Feb. 1, 2016

Due: Wed. Feb. 10, 2016 at 23:59

General Instructions

- Please use the mycourses discussion boards for clarification questions, rather than emailing the prof or TA. Only email the prof/TA when your question might give away too much information or if you are sure that it is not of general interest.
- TA office hours and locations will be posted on myCourses. Only Noor and Josh are handing A1.

nurmemet.abudukelimu@mail.mcgill.ca

joshua.romoff@mail.mcgill.ca

- **Late assignments** will be accepted up to only 3 days late, and will be penalized by 20 percent per day. If you submit one minute late, we reserve the right to treat this as equivalent to submitting 23 hours and 59 minutes late, etc.

Verify that your submission was indeed submitted correctly. If it was not (e.g. you submitted for another course) then your assignment will be considered late.

- **Bonus points:** If you suspect there are ambiguities or minor errors with the assignment, then notify us immediately. If you are correct in identifying problems, you will be eligible for bonus points.
- **Plagiarism:** We encourage you to discuss the assignment with each other, but the discussion should be public in the sense that anyone should be able to listen in. You are not allowed to tell each other how to do the assignment, although some help is permitted e.g. if you are stuck on a small detail. Under no circumstances should you show solutions to each other. Any strongly suspected cases of plagiarism will be reported to the Faculty of Science disciplinary officer.

Your Task: Build a sequential multiplier circuit using Logisim

Create an 8-bit unsigned sequential multiplier following the algorithm seen in class. The input (“multiplicand A” and “multiplier B”), output (“product”), START and DONE pins are defined in the circuit file that you are provided. Note that the product should have 16 bits.

The following specifications must be met for you to receive full points:

The START pin is used to control the two stages of the computation, namely loading the inputs and carrying out the product. When START goes high (1), all registers must be cleared and the two 8-bit inputs A and B must be loaded at the next rising edge of clock. (All bits of A and B should be loaded simultaneously.) Then, when START goes low (0), the circuit should start executing multiplication at the next rising edge of clock and DONE should go high after eight more clock cycles, namely when the multiplication is over and the 16-bit product result is ready. Use a counter to determine this conditions. When DONE goes high, all lines and registers except those of the clock should stop changing. DONE should remain high until START is set again. A few more details: Intermediate results can be displayed at the output, but final result should remain displayed as long as DONE is high. There is no need to clear registers when DONE goes high.

Your circuit will be automatically tested with all combinations of A and B. For our automatic testing scheme to work, it is crucial that START behave as described above. Note that you should be able to restart the circuit by poking START twice (setting and resetting) after a multiplication is done, or even in the middle of an operation. Test your circuit extensively before submitting to make sure this restart condition is satisfied.

You may use any combinational circuits (e.g. gates, multiplexers, comparators, ...) or sequential circuits (registers, a counter, a shift register, ...) from the default Logisim library, except obviously you are not allowed to use a Multiplier circuit. Note that the Logisim shift register is right shift only. The algorithm requires that you left shift the multiplicand. To do so you will need to build your own left shift register. You should use a subcircuit to do so.

Use the provided file to create your circuit. The input and output pins are defined there and must remain as they are. You can change the pin positions on the canvas and change the wires, but the pins must remain there. Moreover, the filename and circuit name must remain unchanged to allow for automated testing.

Include your name and student number as a text comment in your main Logisim circuit.

Use a zip archive to bundle your submitted files since this is assumed by the script that the TA’s use to organize the submissions. Do not use other types of archives. *Do not include any directory structure in your submitted zip file* e.g. in MS Windows, select the file(s), right click, and choose ‘send to compressed folder’. You may include a README.TXT file in the zip file if you wish to explain anything about your submission and add a comment to your mycourses submission.

Downloading your submission from the server and verify that it was correctly submitted. You will not receive marks for work that is incorrectly submitted.

Tips

You will need to learn the basics of Logisim before starting this assignment. Go through the tutorial and the Help menu, and consult the documentation at <http://www.cburch.com/logisim/docs.html>

Logisim uses terminology and notation that sometimes differ from the lectures or were not discussed in the lectures. You will need to become familiar with terms such as “pin”, “probe”, “splitter”. Part of the fun of using Logisim is discovering what these various elements are and how they are used.

Start with some simple examples. Build a few basic combinational circuits. Then build your own shift left register, which you can use as a subcircuit in your solution.

Use the whole canvas, rather than cramming your whole circuit in a small fraction of the space.

For various circuit elements such as a right shift register, multiplexor, counter, note that the input and output lines have specific locations and values and note that the attributes that can be edited.

To add in debugging, use probes to read values on lines. Similarly, add text comments to indicate what your circuit elements are doing, just as you would add comments to a program.

Note the difference between a “pin” and a “constant”. When you define a subcircuit e.g. your shift left register, and insert an instance of this subcircuit into your main circuit, the pin values of the subcircuit will be accessible, whereas constants in the subcircuit will not be accessible.

Do not use any three-state pins in your circuit. This can cause peculiar problems with testing, and it wastes the TAs time. You don't want the TAs to be unhappy when they are grading your assignment.

Save your file periodically. Logisim is not bullet proof. It can happen that an error occurs when entering keyboard or mouse inputs, e.g. dragging elements, and you can't save your file. In that case, you will need to revert to a previous version. If you don't use any sort of version control system or cloud backup, now is time to start.

You may occasionally draw wires incorrectly without realizing it, and find yourself wasting time staring at a circuit that seems correct but is not behaving properly. For example you might think you have attached two wires to two inputs slots of a circuit element, but in fact the wires might be joined together as one wire and not touching the circuit element at all. Try deleting the wires and adding them again, and/or moving the circuit element to make sure there are no hidden wires underneath.

Get started early, and have fun!