TCSS 372, Fall 2018 HW3, Simple Pipeline Machine

Due: by midnight Wednesday. Nov. 7 (or with 10% late penalty, Friday. Nov. 9).

In this assignment you will build a simple version of the pipeline machine outlined in Chapter 4. There is a starting circuit that you must use as the basis for the assignment. It is a complete solution to HW2, i.e. it is a single cycle machine. You must:

- add the 4 major pipeline registers: IF/ID, ID/EX, EX/MEM, MEM/WB
- run data and control signals through the pipeline registers as in Figure 4.51
- implement forwarding as shown in Figure 4.57 (A logisim subcircuit is required.)

Your components should be laid out in a way that closely resembles these diagrams.
Each component should be clearly located in one of the 5 stages: IF, ID, EX, MEM, WB

You must NOT implement the following advanced pipeline features:
- Stalling or insertion of "bubbles" into the pipeline (so no Hazard Detection Unit)
- Faster branching (where branch decisions are made in the ID stage as in Figure 4.62)
- Flushing pipeline registers after a branch instruction branches in the MEM stage.

The starting circuit includes the following functional units which you completed in HW2:
- Instruction Splitter, Control, Register File, ALU Control, ALU, Data Memory.
  You may not modify the inside of any of these 6 functions units.

Because your pipeline has no forwarding and cannot stall or flush, you will have to be careful what kind of programs you try to run on it. At least two public test programs will be available and both are written to work in a machine that lacks these features. Test programs will use 0 as base register value!

There are some strict rules for this assignment. They are meant to discourage plagiarism. In summary, you must
- use the starting circuit
- not change the internals of the 6 functional units
- not implement stalling, faster branching, or flushing after a branch

If you don't follow these rules, I will assume you found a similar circuit on the internet and turned that in, in which case, no credit will be awarded.

You may want to add the pipeline registers one at a time from left to right, starting with IF/ID. The advantage of this approach is that you can test how the first pipeline register works before building the second etc. There will be a lot of wires. In class we will see a new feature of Logisim called tunnels which you may find useful. Don't use more than a few tunnels as they make a circuit harder to understand. You must create the pipline registers as subcircuits and modify the drawing of the subcircuit so it looks like the tall skinny rectangles of the textbook. The implementation of pipeline registers is extremely simple. They contain one or more registers. Inputs to the registers should be on the left and outputs on the right. Clock connections will be neeeded.

Just as in HW2, all registers and the memory should be connected directly to the clock, but this time there is one exception. The Register File must connect to the NEGATED clock. This allows a register to be written during the first half of a cycle and the new value to be read during the second half.