Project Description

For your final project, you will have to implement a pipelined processor. The processor will implement the same instructions as the single-cycle one:

- arithmetic: add, addi, addu, sub, subu
- logical: and, or, sll
- data transfer: lw, sw
- conditional branch: beq, bne, bgtz, slt, sltu

The processor has to include forwarding logic, but other than that, there is no grading for performance, so you are free to choose any optimization you want to include.

You can find the necessary files (FinalGroupProject_VHDL and _Verilog) under Files->Project_files. These files contain the VHDL and Verilog components you can use during the implementation (lib), and the three applications that the processor will be running (data). Later, we will also include a single cycle processor implementation (SingleCycleProcessor). In case you cannot finish the first project, you can work on this design for your final project. Please note that you cannot use any components that are not included in the lib directory. You can build your pipelined processor using the single cycle processor that will be available or your own implementation from the first group project.

Test Files

Similar to the first group project, three test programs are included. The format of these input files is

address / data_at_address;

Thus, the input file has one address-data pair per line in hex. Comment lines are allowed, and must begin with a #. There are three test files attached to the assignment, so you don't need to write your own test files. Here are the test programs:

- a sort program – sort_corrected_branch.dat
- a summation program – unsigned_sum.dat
- a simple transaction simulator – bills_branch.dat

The test files are extracted from a MIPS simulator called SPIM. The simulator assumes that code begins at address 0x00400020, so make sure that the program counter is loaded with this initial value.

THINGS TO TURN IN:

You should turn in a report describing and justifying your design decisions and describing the problems you have encountered and how you solved them. In addition to this report, please submit:

1. All your VHDL/Verilog codes (datapath, register file, ALU, etc.)
2. Traces of running the above programs, demonstrating that the correct result was obtained (i.e., you have to figure out what the program did, and show the memory location(s) where the results are stored).

Please note that you will do a demo of your processor.

Please, submit your report electronically through Canvas before the deadline or bring a hard copy to your demo.

Deadline:

Typically, students will continue with the groups they have formed for the first group project. If you would like to change your group, please inform the instructor as soon as possible.

The demos will be held on **Thursday, December 13th** throughout the day. There will be a signup sheet for determining the exact time of your demo. Your project reports are due the end of the same day.