# CS 3421 Final Processor Project

Fall 2018

Due: 14 December 2018

In a nutshell, you must implement the full single-cycle processor architecture shown in Figure 1 in Verilog. It should be able to execute simple programs. This is a tiered assignment. Successful completion of all tiers will give you an automatic 'A' on the Final Exam. The rest of this document details the requirements of the assignment.

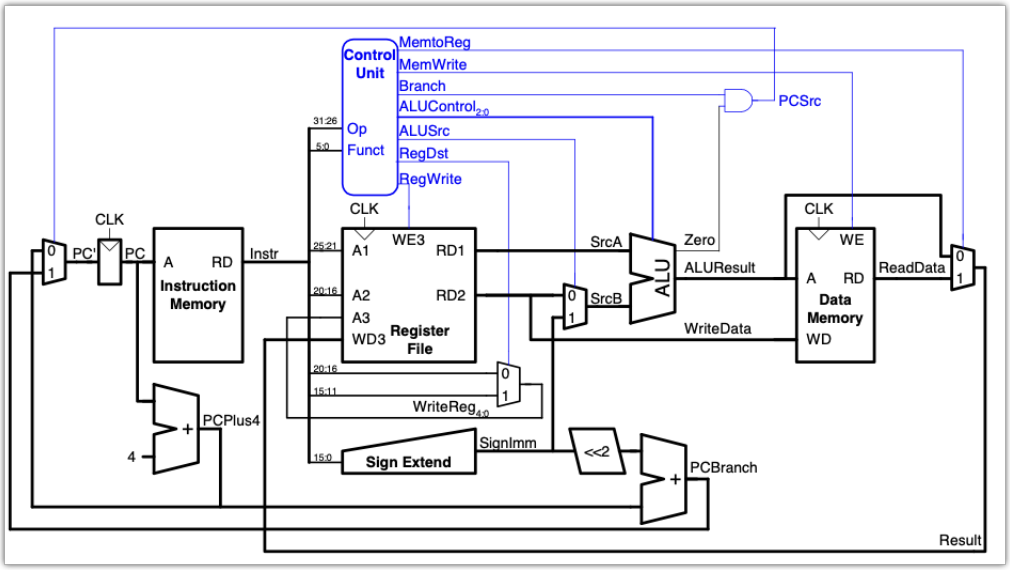


Figure 1: The Simple Single-Cycle Architecture

## Basic Requirements

In this most basic and essential version of the assignment, you will implement the single-cycle architecture for the minimal set of the following instructions:

- The R-type instructions `and`, `or`, `add`, `sub`, and `slt`

- The `lw` and `sw` memory-accessing instructions
- The single `beq` branch instruction

The structure of your Verilog modules must be substantially the same as the diagram. That is, there should be explicit instantiations of instruction memory, a sign-extending module, the ALU you created earlier in the semester and so on. You must be the author of every module, even if you are borrowing code from the slides. That means that you must write the mux code, you must write the sign-extender code and you must even implement the shifter module using the design shown in Lecture 7. The only modules you can fall back on a high-level behavioral model (e.g., a `case` statement) are the multiplexers and the Control Unit.

At least one sample program will be posted for you to test your design with. It is strongly encouraged that you perform additional tests to be sure your design is functional. A final, ultra-secret test program will be run against your design as the last step in the grading process.

All designs should be done so that they simulate properly with ModelSim.

The posted test program is worth 75 points. The ultra-secret test program is worth 125 points.

## Tier 2: Added Functionality

Tier 2 is meant for those who simply want to raise their homework grade in the class. All additional points from this tier of the assignment will simply be added to your Homework score (which, as a reminder, is 50% of your overall grade).

All of the Basic Requirements remain. The additional requirements here are to add the `addi` and `j` instructions to the single-cycle architecture. This should be done in the same manner as the Basic Requirements, explicitly instantiating the additional hardware and updating the `case` statements in the Control Unit.

A sample program will be run against your design to verify functionality of these new instructions. If successful, 75 points will be added to your grade for this assignment.

## Tier 3: Final Exam Replacement

At tier 3, a 100% successful implementation of the processor will earn you an automatic 'A' on the Final Exam. To complete tier 3, you need to implement several more instructions. These should not complicate the core datapath very much, do not require any changes to the ALU, but will definitely require additional control logic. The required instructions, their opcodes and behavior are specified in the table below.

All instructions must be implemented properly and be successfully tested before the Final Exam will be waived.

| Instruction | Opcode | Operation |
|---|---|---|
| `addiu` | `001001` | Adds `rt` and a zero-extended immediate |
| `andi` | `001100` | Ands `rt` and a zero-extended immediate |
| `bne` | `000101` | Branch on not equal |
| `ori` | `001101` | Ors `rt` and a zero-extended immediate |
| `slti` | `001010` | Performs `SLT` with a sign-extended immediate value |

Figure 2: Required Instructions for Tier 3

## Submission

Submit only your Verilog files for all modules. Explicitly email me if you want to be considered for Tier 3.