Student Designed CPU Simulation

Part	Due Date
Project Meeting Last Date	Wednesday, March 12th
ISA Document	Wednesday, April 09th
Hardware and Control Unit Document	Monday, April 28th
Initial Project Demonstration	Wed/Fri, May 07th & 09th
Final Project Submission	Tuesday, May 20th
Final Project Demonstration	Wednesday, May 21st

You will be designing and implementing the simulation of your own CPU using the C++ *arch* package.

1. Goal

During the quarter you will be exposed to a number of issues and solutions that computer architects have to consider when designing a CPU. The goal of this project is to allow you to explore specific areas of computer architecture. Areas that you may consider are:

Micro-coded C.U.	Out-of-order execution
Pipelining	Superscaler
Branch prediction	VLIW
Instruction pre-fetching	Data pre-fetching
Cache behavior	etc.

2. Teams

For this project, you may, if you wish, work in teams of two people. You are not required to work in teams. If you choose to work in a team, I would expect the amount of work produced to be greater than what I expect out of a person working alone. The grading for team projects will differ in that a part of the grade will come from the evaluation files (see below) that you and your teammate submits. The due date for team submissions is the same as that for non-team submissions.

If you choose to work as a team, only one copy of your solution code should be submitted. The names of **both** team members should be in **each and every** file submitted. Be sure to distinguish between the primary author and the contributing author in code files.

3. Project Components

This project has six due dates specified above. These intermediate due dates exist to make sure that the teams are making good progress on the project. In addition, they give me a chance to provide feedback on the scope and direction of your projects.

I understand that changes may need to be done with decisions that you make early, and I have reviewed. If this happens you should feel free to make changes to your design. Keep in mind that I will be the final grader on the project, and if you make major changed to your project, it is possible that I may not feel that the project is of the correct size and scope. So, if you make major changes to your project you should make sure to discuss them with me. In addition, at the end of the project each team will be required to submit an Addendum document which should document any and all changes you made to the previously submitted documents.

3.1. Project Meeting

Each team is required to schedule a meeting with me by Wednesday, March 12th to discuss their project. At this meeting the team will go over their thoughts on the area(s) that they will be focusing on in their project. The team will then have to go through a brief description of the ISA and hardware they will be designing.

It is my goal with this meeting to give the teams feedback on the scope, scale and viability of their project. In addition, the meeting will allow me to give the team hints and insight about implementing their project using the *arch* package.

3.2. ISA Document

The ISA document should describe all the external characteristics of your proposed system. This should include (at the very least):

* Instruction format, with a complete description of how the instructions will be encoded.
* A list of all the instructions in your system, including opcode, the mnemonic, a description, and any information about encoding
* Information about the memory space, address size, bits-per-addressable-unit, width of the bus to memory (how many units get transfered per load/store)
* Name and description of all the program visible registers
* Description of all address modes supported, complete with how the address modes will be encoded.

3.2.1. Submitting the ISA Document

Two copies of your ISA Document are due in class on Wednesday, April 09th. You should make sure that your document contains the names of both member of the team (if applicable).

3.3. Hardware and Control Unit Document

This document contains two parts:

3.3.1. Hardware Description

This is a complete description of the programmer visible, and non-visible, internal, components of your system. You should include a brief description of each component and their use. In addition, you should include a diagram showing the connection of all the components.

3.3.2. Control Unit Description

The description of the implementation of the Control Unit for your instruction set in RTL. You should include the RTL for handling the fetch and decode processes as well as the complete RTL to execute each of the instructions within your ISA.

3.3.3. Submitting the Hardware and Control Unit Document

Two copies of your Hardware and Control Unit Document are due in class on Monday, April 28th.

3.4. Initial Project Demonstration

Each team will schedule a time during class on Wed/Fri, May 07th & 09th where they will demo and discuss the current state of their project. This is not meant as a final demo of system, instead I am looking for the teams to show satisfactory progress on the project. At this point I would like to see the projects able to handle input (possibly having the fetch/decode stages working) in addition to demo-ing the trace output the the project will use. If the team is implementing their own *arch* package objects, then I would like to see the running of some debug code for the component so I could see the state that that work is in.

In addition, at this time I would like each team to put together an informal document (not hand-written) that describes the work being handled by each member of the team. Also this document should contain some kind of schedule or plan of their remaining work, and a list of who is assigned responsibility for each unfinished tasks.

3.5. Final Project

3.5.1. Simulator Requirements

3.5.1.1. Simulation Code Name

You should name the source file that contains your main function `CPU.cpp`.

3.5.1.2. Tracing

Your simulation should produce a tracing of your programs execution. This tracing can be as simple as the output produced by the previous programming assignments, or something more complex to highlight the focus of your project (for pipelining you might want to specify the instruction, the result, and the cycle number that the instruction exited each stage of the pipeline).

3.5.1.3. Command Line Arguments

Your program should always take at least 1 command line argument, the name of the obj file for the program to execute. You can add more arguments as long as you completely document them in the **README** file supplied with your submission.

3.5.2. Addition files

3.5.2.1. Object Files

You will need to submit a number of *arch* object files that exercise your simulator. This should include simple files, that tests the individual instructions in your ISA, as well complex programs that demonstrate to me that your simulator works as expected. Included in the complex programs should be examples that highlight the topic that your simulation was build to explore.

You will need to provide a description in the **README** file for each *arch* object file that you submit. This description should include, at the very least, information about what the object file is testing/highlighting and some kind of description on how you can interpret the results.

3.5.2.2. README file

You must submit a text file named **README** with your project. This file should contain your name(s), and a description of the project. It should also contain a list of all the files submitted describing what the files contain, and why they were submitted. In addition, you should include any notes you think would be useful while I grade the assignment.

3.5.2.3. Addendum Documents

You should submit a text file named **ADDENDUM** that documents any changes that you made to the any of the previous documents.

If you would like more formatting features than can be provided by a text file, you can submit this as a printout. In this case you will still need to submit an **ADDENDUM** text file, but this can just point out that the document was submitted in hardcopy.

3.5.3. Handing In the Assignment

By the end of the day (i.e. 11:59:59pm on the CS system clock) on Tuesday, May 20th Submit your solution using the command below; you must submit your source files (all `.cpp` and `.h` files you write) plus the `README` and `ADDENDUM` files. The try script will generate a `Makefile` (using `~csci453/pub/misc/header.mak` and the makemake program as discussed in class) and recompile your submission for testing.

To submit your solution, use a command such as

```
try  grd-453  project-135 CPU.cpp README ADDENDUM PROGRAM_FILES
```

where *OTHER_PROGRAM_FILES* contains the names of all the other program files (`.cpp` and `.h` files) you are submitting as your solution.

NOTE:

I do not want any C++ object or executable files from you.

Do not submit a directory, just submit all files pertaining to the project.

4. Final Presentation/Demo

During the final exam period, each team will present their project to the class. This presentation should be about 10-15 min long, and should include some background on the scope of the project, and a demo of your simulation.

5. Teams

In addition to submitting the solution, **BOTH TEAM MEMBERS MUST** submit a team evaluation by 11:59:59pm on Wednesday, May 21st. This is a text file named **EVAL** which contains your name, your teammate's name, an explanation of the work done by each of you (i.e., who worked on what part of the solution. This should include design work, specific documents, parts of the fetch/execute cycle, instruction implementation, integration, testing, etc.), and your comments on the amount and quality of work done by yourself and your teammate. I expect you to take these evaluations seriously, as this will make up a part of your grade. Submit this file with the command

```
try grd-453 135-team EVAL
```

6. Grading

The project will make up 25% of your course grade. This 25% will be broken up in the following ways:

Part	Grade %
Project Meeting	2%
ISA Document	4%
Hardware and Control Unit Document	4%
Initial Project Demo	3%
Project Presentation/Demo	5%
Final Project	7%