ECE251: Computer Architecture 2 April 2024

Final Project

Professor: Rob Marano *Due: Friday, 10 May 2024 11:59:59 ET*

Introduction

This course requires each student to design and implement a von Neumann computer with its five basic components, i.e., CPU, memory, datapath, input, and output. You are to implement the design using Verilog and to include a test bench to demonstrate the full functionality of your processor and memory, especially loading and running programs as well as taking input and showing output to the user. This project will have a total value of 200 raw points. There is allocation for a possible 70 points of extra-credit, should your team decide to submit. Final grades will be curved.

Requirements

The following requirements will hold for full assessment:

- **Team**
  Not less than and not more than two students per project team.

- **Instruction Set Architecture (ISA)**
  Design your ISA, document the design, and summarized on two-sides set on US Letter sized document. You are effectively creating your own ISA's ”Green Card.”

- **Mechanism for Submission of Project**
  Each team will submit their project in their own GitHub repository. The repository will include all Verilog code files, documentation in markdown format, supporting images, etc. Follow this URL when logged into GitHub to join the Spring 2024 classroom Git project: https://classroom.github.com/a/pelSJLGu

Rubric

The following topics will be assessed for each project.

- **ISA**
  ALU Operand Size, Address Bus Size, Addressability, Register File Size, Opcode Size, Function Size, shamt Size, Instruction Size, PC Increment, Immediate Size, R-type Instruction support, I-type Instruction support, Memory Reference Support, J-type Instruction support, R-type Instructions, I-type Instructions, J-type Instructions

- **Memory Design & Implementation**
  Instruction Memory, Data Memory, Memory Layout, Program Load into Processor

- **Processor Design & Implementation**
  Clock Design, Overall Control Signals (regwrite,regdst,alusrc,branch,memwrite,memtoreg,jump,aluop[]), Multiplexors (regdst,alusrc,pcsrc,mem2reg), Main Decoder Details, ALU Decoder, R-Type Instruction Impl, I-Type Instruction Impl, J-Type Instruction Impl, PC – Increment for R- and I-types, PC – Increment for J-type, PC – Increment for Cond Branch, Datapath Design (imem, dmem, alu, regfile, signext, sll), PC increment adders (pc+1 adder, pc+jump adder, shift logical left), Register File, Sign Entender(s), ALU (and,or,nor,add,sub,slt), Controller – Datapath Integration, Program Load Integration, Provided Assembly Program in code, minimum of a Hand-compiled Program into machine code for your ISA.

- **Project Documentation**
  Overall Design Explanation, Overall Design Diagrams, R-type timing diagram, I-type timing diagram, J-type timing diagram, and Instructions to Successfully Demo, representative example of a program needs to be included with one at least implementing a nested procedure or recursion. Fibonacci or Factorial algorithm implementation would be acceptable.

Extra Credit

The following topics are available for extra credit.

- Demo Video Recording
  The link to your team's YouTube video to describe your design and demonstrate how to write a program for your ISA, load it, run it, and show output. Ensure it is accessible to the public on youtube.com, but you may categorize is as unlisted.
- Programmatic Assembler
- Pipeline Design Support
- Cache Support

Extended Office Hours

I will provide extended offices on at least two Sunday afternoons either in person at the NAB 41 Cooper Sq or remotely via Microsoft Teams, depending upon my availability on weekends. These office hours are NOT mandatory nor required. The scheduled dates are as follows:

- April 28, 2023, 3:00-6:00pm ET
- May 5, 2023, 3:00-6:00pm ET