Project Plan
for
Project 2: Pipelined Processor

Team: Ryan Pencak, Uttam Kumaran, Evan Harrington, Peyton Rumachik

Contents

1. Overview

This project involves the development of a pipelined MIPS processor. This processor will be implemented in Verilog and build on the previously single cycle processor. The foundation of this project is the project repository for the single cycle processor which contains modules for each part of the MIPS processor. This project will expand the functionality of the processor and improve performance by implementing a five-stage pipeline with EX-EX, MEM-EX, and MEM-MEM forwarding. The pipelining process involves segmenting the process into fetch, decode, execute, memory, and writeback stages. A hazard unit will handle the logic and behavior of stalling the decode stage. The final product will be a pipelined processor that will be able to execute multiple test codes to demonstrate improved performance.

2. Goals

Functional Goals:

- Pipelining segments execution into 5 stages: fetch, decode, execute, memory, and writeback.
- Hazard unit module contains correct logic for handling stall events and stalls for the appropriate number of cycles.
- Forwarding is implemented and the hazard unit logic is adapted to allow for EX-EX, MEM-EX, and MEM-MEM forwarding.
- Working processor that successfully runs the two required programs, Hello World and Fibonacci, as well as a third program to demonstrate the processor functionality outlined above.

Quality Goals:

- Final project code is modular, concise, and well commented
    - Should allow for other people to easily understand the code
- Project repository should be logically laid out and easy to read, including README files for documentation on compiling and running the code.

3. Organization

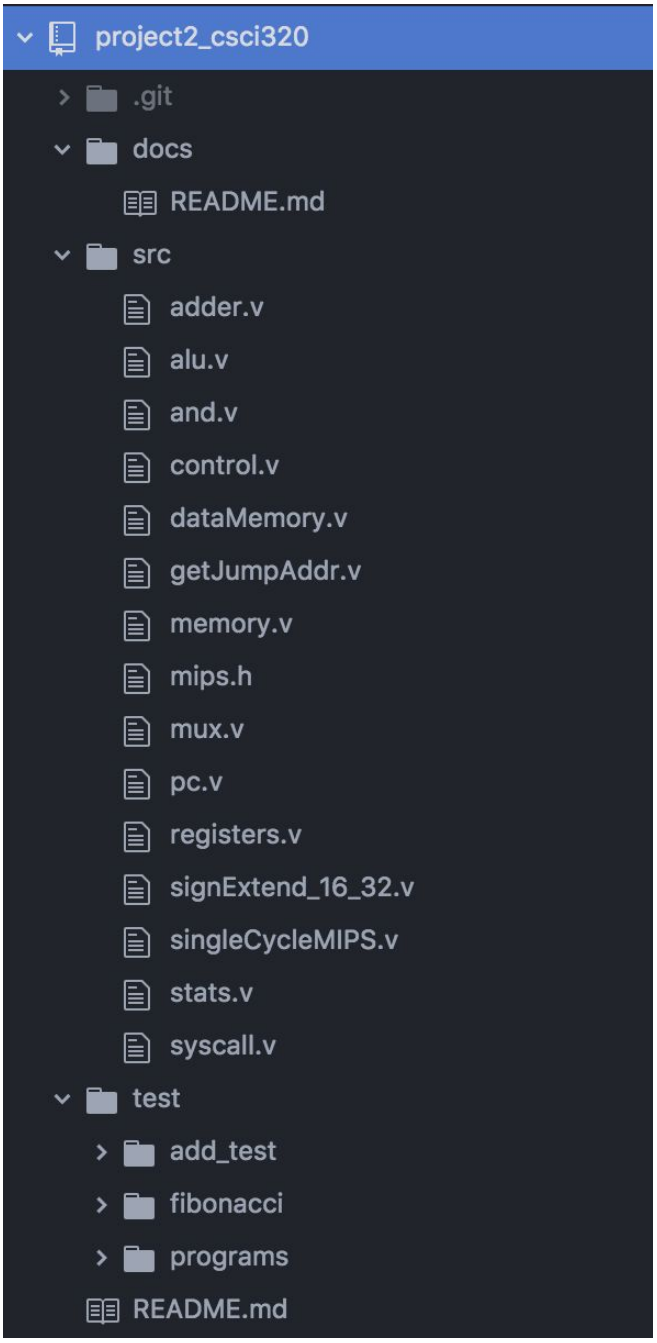
3.1 *Project Organization*

3.1.1 *Project Team*

Organization: Name	**Developement Focus**	**Comment**
Ryan Pencak	Hazard Unit/Forwarding	High-level focus on functionality
Evan Harrington	Hazard Unit	Low-level focus on hazard logic
Uttam Kumaran	Forwarding	Low-level focus on forwarding
Peyton Rumachik	Testing	Ensuring project is functioning

3.1.2 *Project File Structure*

The project will be split into directories for documents, source verilog files, and testing. Documents will contain all details about the project including this work plan and other information. The source directory contains all verilog code, which will soon include modules for hazard unit control and forwarding. Last, testing includes MIPS code for three different programs that will be used to test the functionality of our system (see functional goals Sec 2.1).



4. Schedule

4.1 *Schedule and Milestones*

Milestones	Description	Milestone Criteria	Planned Date
M0	Start Project	Team Planning	02-28
		Project expectations and team details defined	
M1	Segment code into 5 stages		03-08
		Set up processor for pipelining with 5-stages	
M2	Develop Hazard Unit and Forwarding Functionality		03-22
		Create the foundation of verilog modules and code required for these functions	
M3	Finalize Hazard Unit and Forwarding Functionality		03-26
		Test and ensure the stalls and forwarding work with three required programs	
M4	Deliver Project		03-29
		Create final presentation to demonstrate functionality of pipelined MIPS processor	

4.2 Testing Plan

We plan to test our pipelined processor with three different programs. This includes a basic Hello World program as well as fibonacci and a third custom program. This final program will be determined later but will include three more instructions than fibonacci. It will be able to show that our code handles all of the functional goals outlined in Section 2.1.

Our testing procedure will consist of tracking register changes through the MIPS code. We will use GTK Wave to observe changes in register contents and ensure that the processor is executing the instuctions individually and correctly. This will be more complex than testing the single cycle processor because it is pipelined in five stages and registers will change before an instruction is complete. An additional text output may be created to test our progress along the way, which would ultimately show the 5 different stages and what is happening in them as well as hazards that occur and how the hazard control unit deals with them (display stall/forwarding type).

5. Delivery Plan

5.1 *Deliverables*

Ident.	Deliverable	Planned Date
D0	Team planning and work plan	03-01
D1	MIPS processor segmented into 5 stages	03-08
D2	Functioning Pipelined Processor	03-22
D3	Final Presentation	03-29