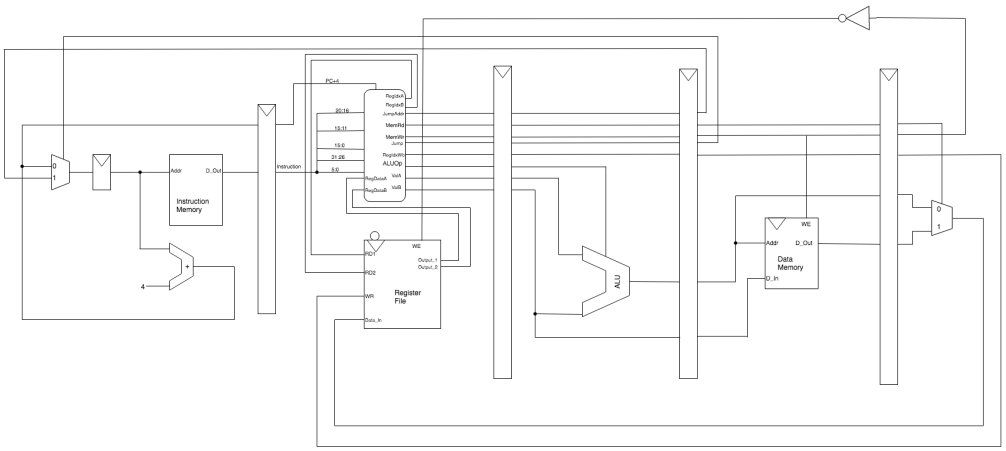
Project Part 1 Pipelined MIPS Processor

Introduction

Students will now to assemble the 5 stages of the microprocessor to create the overall microprocessor architecture as shown in Figure 1. In order to do this outputs of each stage need to be captured into a register. The output of this register needs to be fed as inputs to the next stage. All the registers will operate with the same clock signal to maintain a lock-step pipelined operation of the microprocessor.



Pre-Lab

- Write the pseudo-code for creating a Fibonacci series.

Methodology

Use parallel loading registers of appropriate size (number of bits) so that the output of each stage can be captured. The size of the register can be defined with a Generic. Create an upper level wrapper (a structural higher level) that instantiates all the 5 stages and the registers between them. This is your micro-processor. In addition, instantiate the register file from Lab 2.The overall input of this upper level are the inputs of the IF stage getting the instructions from the instruction memory. The outputs of the microprocessor will be the result of the WB and Memory stages in which case, the bits of these stages are output either to the register file or to the data memory.

Notes About Putting it Together

- The Data Memory should be synchronous write only. Reading should be asynchronous.

- A new Register File is supplied that is fully generic with a falling edge clock. This should be used, as it needs to be a 32x32 Register File. **Please do not route the clock through an inverter before it goes into the register file.**

Procedure

Part A

1. Create a block diagram (not hand drawn) for the overall microprocessor with the register file and the instruction and data memories showing all the connections.
2. Populate the instruction memory with instructions that test the operation of the all the various instructions as listed in Lab Exercise 4 (ID Stage).
3. Create a testbench such that the different kinds of instructions can be fetched from the instruction memory cycle by cycle and execute them. The testbench must be able to show the output of the ALU and the data written back or read/written to the data memory (output of the Mem stage).

Part B

Create instructions based on your pseudo-code to generate at least the first 10 Fibonacci Numbers using your microprocessor. The numbers should be written back in the data memory. Show the output of the ALU and the content of the data memory to demonstrate the creation of the Fibonacci series.

Exercise 1: Pipelined MIPS Processor

Student's Name:_____ Section:_____

PreLab		Point Value	Points Earned	Comments
PreLab	Fibonacci Pseudo-Code	10		

Demo		Point Value	Points Earned	Date
Demo	"Basic Testbench (Part A	25		
	Part B	25		

To receive any grading credit students must earn points for both the demonstration and the report.

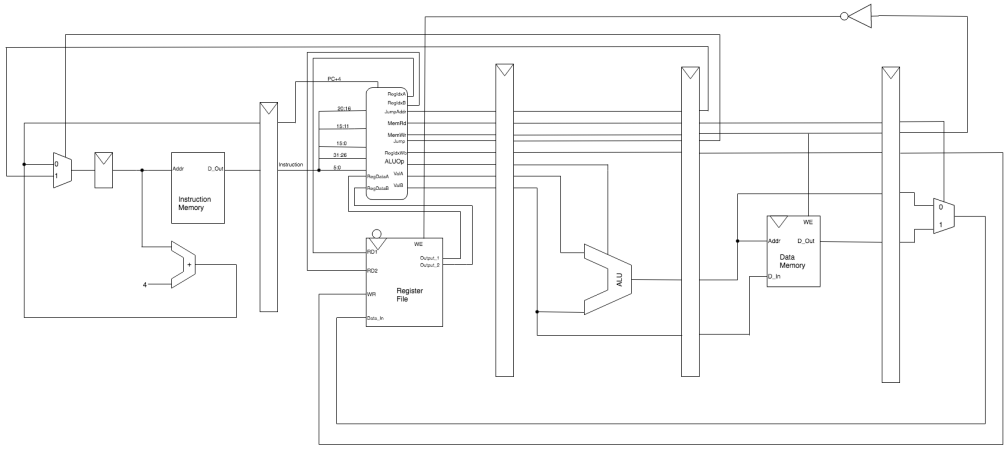
Exercise 1: Pipelined MIPS Processor

Report		Point Value	Points Earned	Comments
Abstract		5		
Design Methodology	Discussion of Full MIPS Pipeline Functionality	5		
	Discussion for Method for Calculating Fibonacci Numbers	4		
Results & Analysis	Part A: Behavioral Waveform	4		
	Part A: Result Discussion	4		
	Part B: Behavioral Waveform	4		
	Part B: Result Discussion	4		
Conclusion		5		
Source Code (Style)		5		
Total for prelab, demo, and report		100		

Project Part 2 Processor Timing

Introduction

In this part of the project, students will be able to observe and understand timing concepts in a pipelined microprocessor.The idea is to enable students to verify static timing performance of the designed 5-stage microprocessor.



Methodology

Use the assembled 5-stage MIPS based microprocessor to verify if the architecture can be operated at a given clock frequency. Use the clocking wizard IP Core to generate clock frequencies within a specified range. Vary the range of clock frequencies from 10MHz in steps of 10MHz until setup or hold times are violated. For each clock frequency instantiated synthesize and implement your design and check if setup and hold times are satisfied by analyzing the Timing Reports. Note and report the highest clock frequency at which the design can be operated without violating setup or hold times.

Deliverables

- A worksheet with the following:
  - A brief explanation of setup and hold times for a flip-flop
  - A table of which frequencies the clock was run at and where it failed.
  - Implementation waveforms for each of the frequencies, noting which ones failed

Notes