CprE 381 – Computer Organization and Assembly-Level Programming

Lab-04 Report

Student Name Colby McKinley

Section / Lab Time Section / Wednesday 6:10p – 8:00p

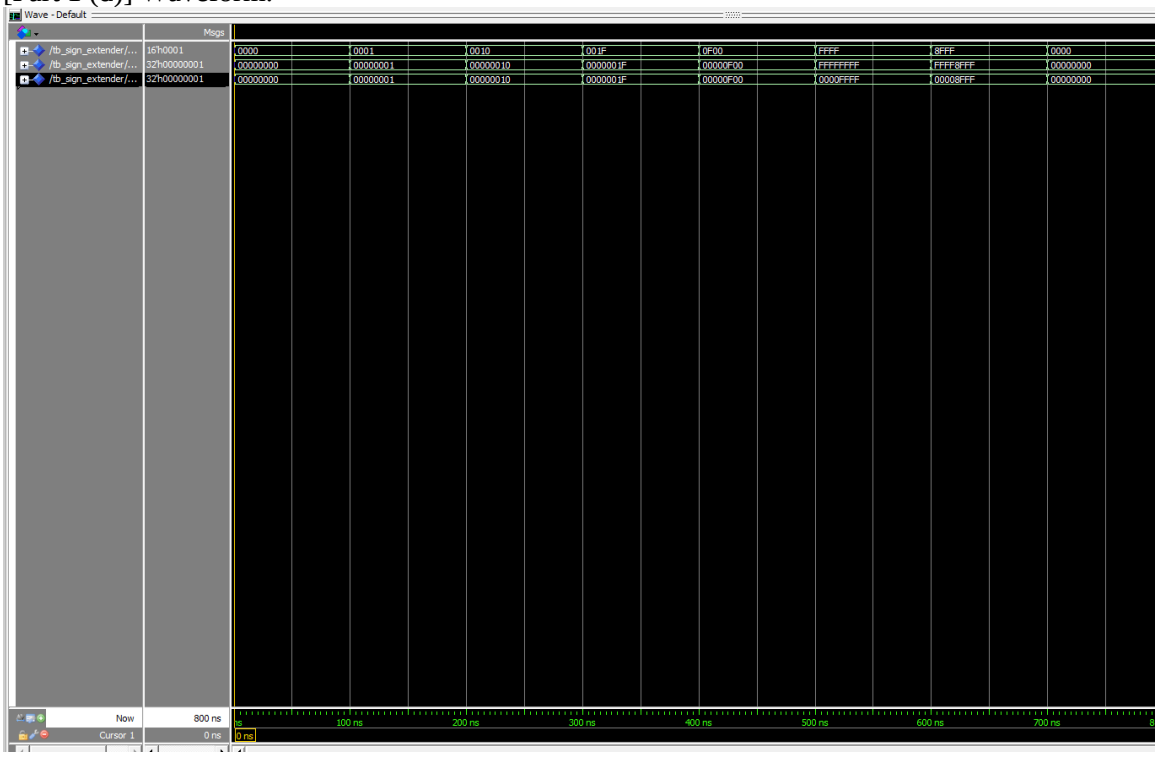
Submit a typeset pdf version of this on Canvas by the due date. Refer to the highlighted language in the Lab-04 instructions for the context of the following questions.

a. [Prelab] Based on the waveforms, provide a description in your own words of how this component operates. This can be in the form of a textual description, flow chart, or state machine
See /prelab

b. [Part 1 (a)] What are the MIPS instructions that require some value to be sign extended? What are the MIPS instructions that require some value to be zero extended?
Signed Extended: Subi, Subiu, lw, sw
Zero Extended: Addi, Addiu, lw, sw

c. [Part 1 (b)] what are the different 16-bit to 32-bit "extender" components that would be required by a MIPS processor implementation?
There are two extenders, a zero extender and sign extender are both used as a second option for the adder/subtractor unit.

d. [Part 1 (d)] Waveform.



e. [Part 2 (b)] Provide a 2-3 sentence description of each of the individual ports (both generic and regular).

The DATA_WIDTH generic specifics the width of each register inside the memory block. Given that the mem.vhd defaults to 10, we know that the max value which can be written to a register is $2^{32} - 1$.

The ADDR_WIDTH generic specifies the amount off registers inside the block. Given that the mem.vhd defaults to 10, we know that there are $2^{10} - 1$ registers available to write to.

The clock port is the clock which dictates if a memory address can be overridden. The implementation of mem.vhd dictates that we can write only on a rising clock edge.

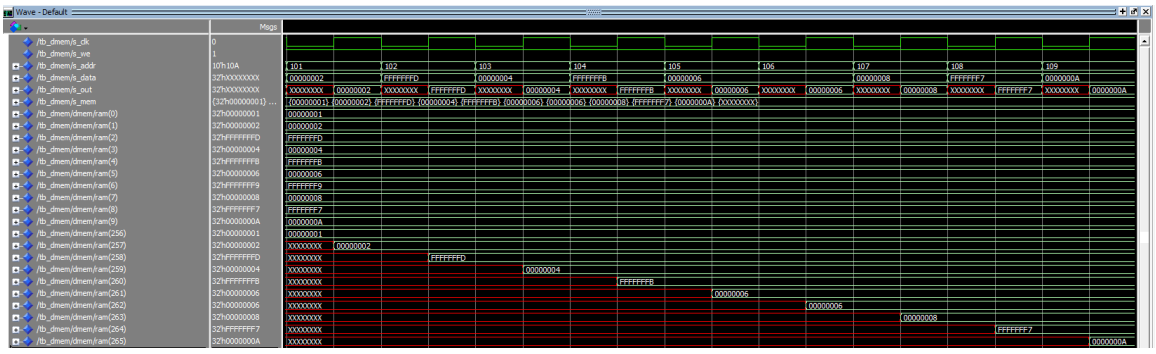
The addr port is the address to read or write in memory. This port width is the size of the generic ADDR_WITDH.

The data port is the data value that could be written to a memory addr. This port width is the size of the generic DATA_WITDH.

The we port is the write enable for the memory block. This is the control bit for if a value at the given addr will be overridden.

The q port is the memory value read from the given address inside the memory block. This signal is not dependent on the clock signal.

f. [Part 2 (c)] Waveform.



g. [Part 2 (d)] Briefly describe how the waveforms for this mem.vhd module differ from those that you analyzed as part of the pre-lab.

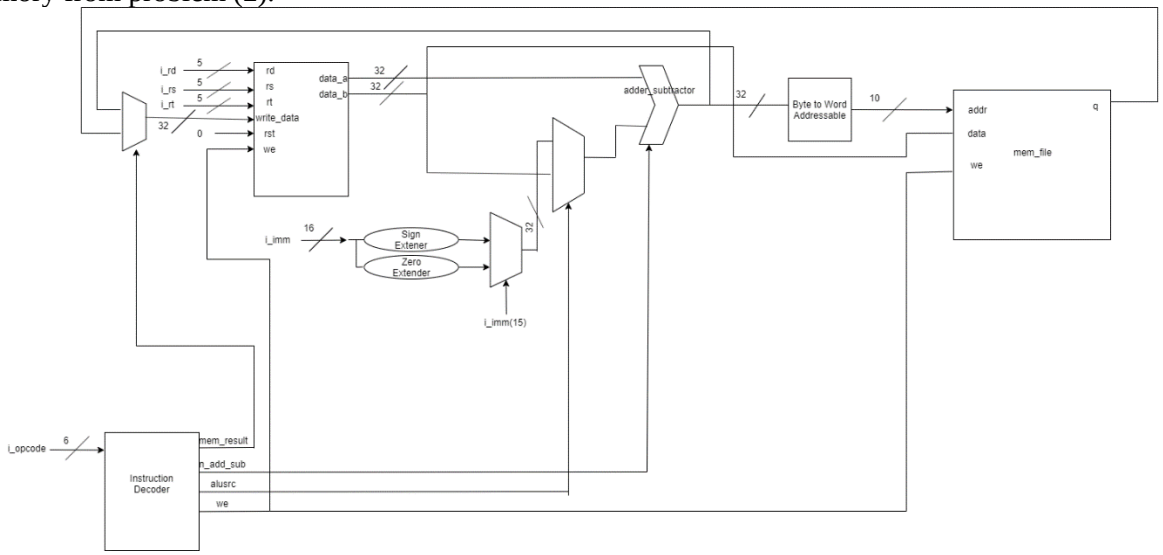
One difference is that q is undriven for half the clock cycles. In this we do not observe the byteena signal. It most resembles the

h. [Part 3 (a)] what control signals will need to be added to the simple processor from Lab #3? How do these control signals correspond to the ports on the mem.vhd component analyzed in problem 2)?

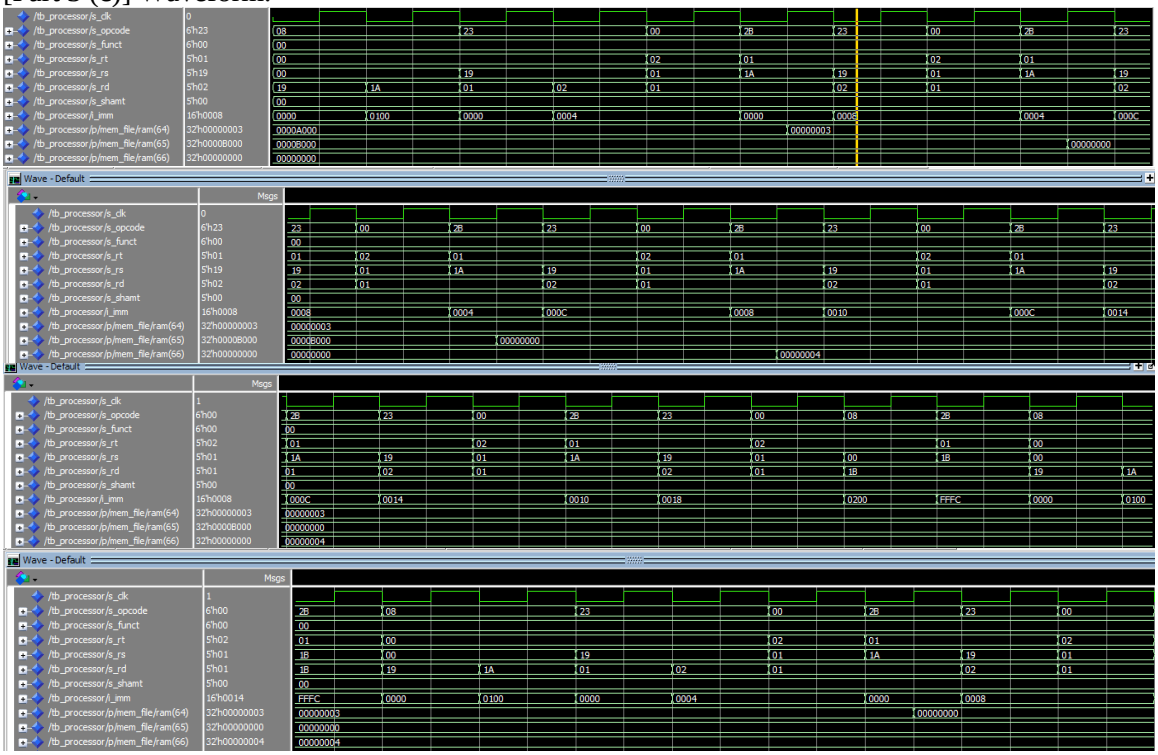
One design decision I made immediately was to have a single opcode drive all my values. I wanted my MIPS processor to be reusable later. I use the opcode to drive 4 control bits, mem_result, which dictates if the value fetched from memory should be used in the register file's write data port or the value from the alu; n_add_sub, which dictates if the adder

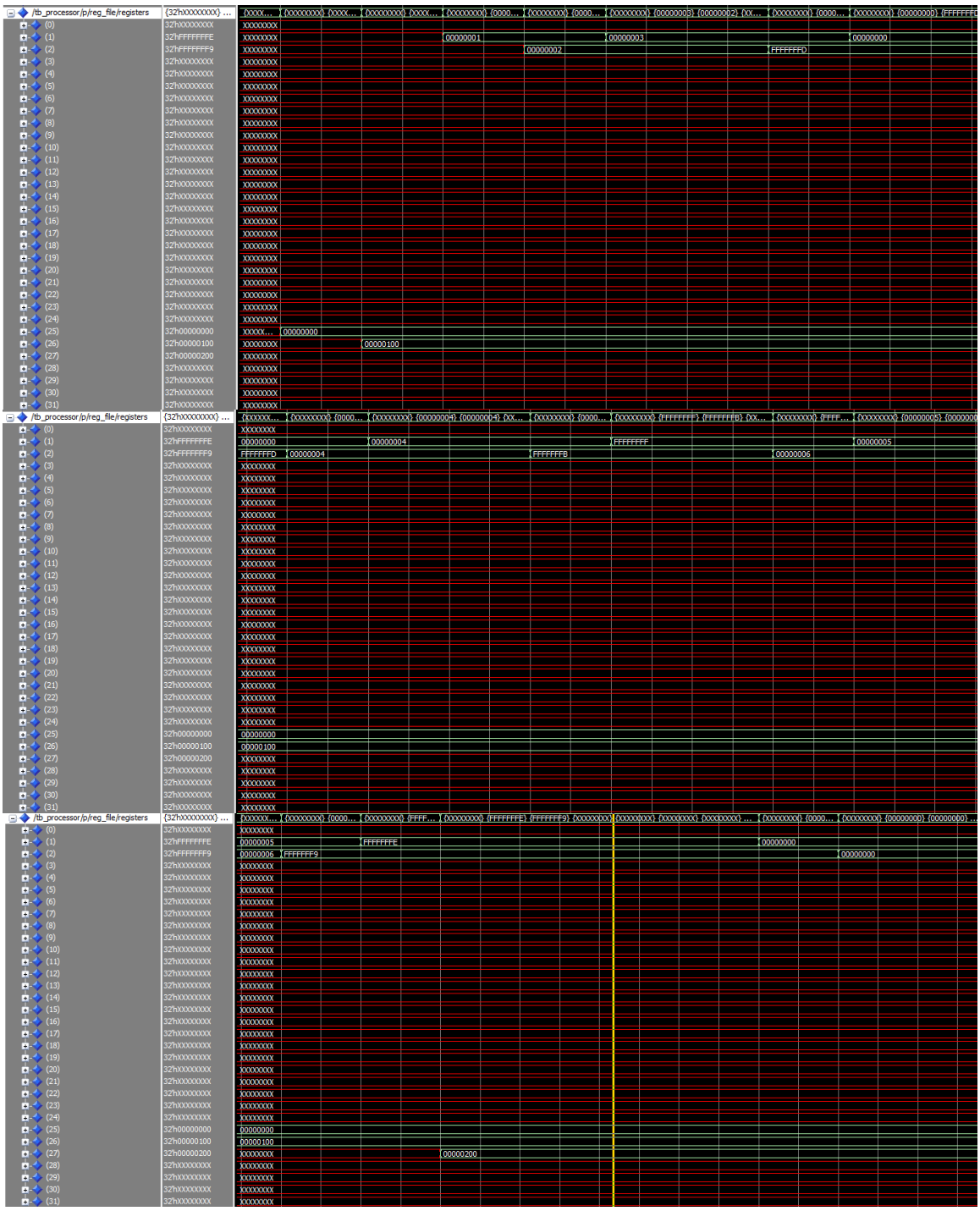
subtractor module should add or subtract; alusrc, which dictates if the immediate value should be used or the value read from the secondary read out from the register file; and we, which dictates if register values will be overridden. In my component ports, you will also find a function, shamt, however these were not required for the instruction we are supporting this week.

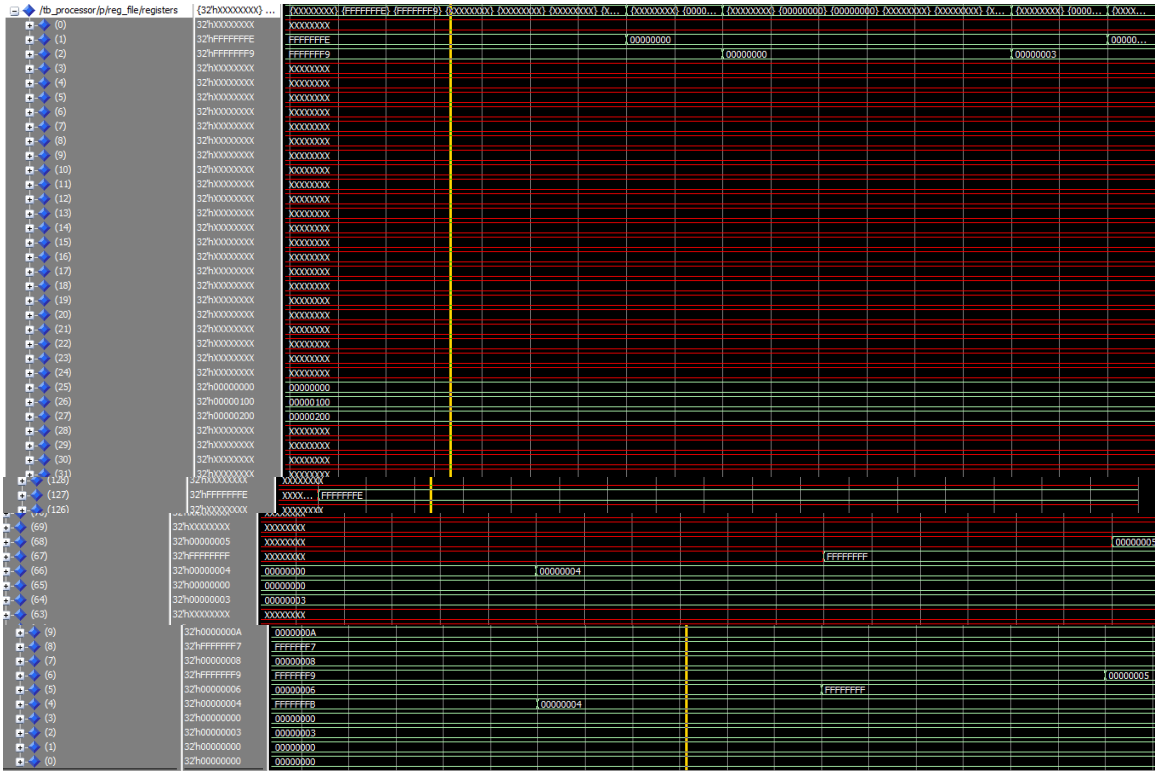
i. [Part 3 (b)] Draw a schematic of a simplified MIPS processor consisting only of the base components used in Lab #3, the extender component described in problem (1), and the data memory from problem (2).



j. [Part 3 (c)] Waveform.




/tb_processor/p/reg_file/registers
{32'hXXXXXXXX} ...
(0) 32'hXXXXXXXX
(1) 32'hFFFFFFFE
(2) 32'hFFFFFFF9
(3) 32'hXXXXXXXX
(4) 32'hXXXXXXXX
(5) 32'hXXXXXXXX
(6) 32'hXXXXXXXX
(7) 32'hXXXXXXXX
(8) 32'hXXXXXXXX
(9) 32'hXXXXXXXX
(10) 32'hXXXXXXXX
(11) 32'hXXXXXXXX
(12) 32'hXXXXXXXX
(13) 32'hXXXXXXXX
(14) 32'hXXXXXXXX
(15) 32'hXXXXXXXX
(16) 32'hXXXXXXXX
(17) 32'hXXXXXXXX
(18) 32'hXXXXXXXX
(19) 32'hXXXXXXXX
(20) 32'hXXXXXXXX
(21) 32'hXXXXXXXX
(22) 32'hXXXXXXXX
(23) 32'hXXXXXXXX
(24) 32'hXXXXXXXX
(25) 32'h00000000
(26) 32'h00000100
(27) 32'h00000200
(28) 32'hXXXXXXXX
(29) 32'hXXXXXXXX
(30) 32'hXXXXXXXX
(31) 32'hXXXXXXXX
00000001
00000003
00000000
00000002
FFFFFFFD
00000000
00000100
/tb_processor/p/reg_file/registers
{32'hXXXXXXXX} ...
(0) 32'hXXXXXXXX
(1) 32'hFFFFFFFE
(2) 32'hFFFFFFF9
(3) 32'hXXXXXXXX
(4) 32'hXXXXXXXX
(5) 32'hXXXXXXXX
(6) 32'hXXXXXXXX
(7) 32'hXXXXXXXX
(8) 32'hXXXXXXXX
(9) 32'hXXXXXXXX
(10) 32'hXXXXXXXX
(11) 32'hXXXXXXXX
(12) 32'hXXXXXXXX
(13) 32'hXXXXXXXX
(14) 32'hXXXXXXXX
(15) 32'hXXXXXXXX
(16) 32'hXXXXXXXX
(17) 32'hXXXXXXXX
(18) 32'hXXXXXXXX
(19) 32'hXXXXXXXX
(20) 32'hXXXXXXXX
(21) 32'hXXXXXXXX
(22) 32'hXXXXXXXX
(23) 32'hXXXXXXXX
(24) 32'hXXXXXXXX
(25) 32'h00000000
(26) 32'h00000100
(27) 32'h00000200
(28) 32'hXXXXXXXX
(29) 32'hXXXXXXXX
(30) 32'hXXXXXXXX
(31) 32'hXXXXXXXX
00000000
00000004
FFFFFFFF
00000005
FFFFFFFD
00000004
FFFFFFFB
00000006
00000000
00000100
/tb_processor/p/reg_file/registers
{32'hXXXXXXXX} ...
(0) 32'hXXXXXXXX
(1) 32'hFFFFFFFE
(2) 32'hFFFFFFF9
(3) 32'hXXXXXXXX
(4) 32'hXXXXXXXX
(5) 32'hXXXXXXXX
(6) 32'hXXXXXXXX
(7) 32'hXXXXXXXX
(8) 32'hXXXXXXXX
(9) 32'hXXXXXXXX
(10) 32'hXXXXXXXX
(11) 32'hXXXXXXXX
(12) 32'hXXXXXXXX
(13) 32'hXXXXXXXX
(14) 32'hXXXXXXXX
(15) 32'hXXXXXXXX
(16) 32'hXXXXXXXX
(17) 32'hXXXXXXXX
(18) 32'hXXXXXXXX
(19) 32'hXXXXXXXX
(20) 32'hXXXXXXXX
(21) 32'hXXXXXXXX
(22) 32'hXXXXXXXX
(23) 32'hXXXXXXXX
(24) 32'hXXXXXXXX
(25) 32'h00000000
(26) 32'h00000100
(27) 32'h00000200
(28) 32'hXXXXXXXX
(29) 32'hXXXXXXXX
(30) 32'hXXXXXXXX
(31) 32'hXXXXXXXX
00000005
FFFFFFFE
00000000
00000006
FFFFFFF9
00000000
00000000
00000100
00000200



k. [Feedback] You must complete this section for your lab to be graded. Write down the first response you think of; I expect it to take roughly 5 minutes (do not take more than 10 minutes).

i. How many hours did you spend on this lab?

Task	During lab time	Outside of lab time
Reading lab	10 min	10 min
Pencil/paper design	20 min	40 min
VHDL design	2hr 20 min	30 min
Assembly coding	0 min	
Simulation	1 hr	1 hr
Debugging	10 min	6 hr
Report writing	0 min	1 hr
Other:		
Total	4 hr	9 hr 20 min

ii. If you could change one thing about the lab experience, what would it be? Why?
The directions for part c are very ambiguous, the only starting point I had was piecing together many small components designed in lecture, and discussing architecture design with a ta in lab.

iii. What was the most interesting part of the lab?

The picture I drew of the MIPS processor is interesting.