CprE 381 – Computer Organization and Assembly-Level Programming

Lab-04 Report

Student Name Brandon Cortez

Section / Lab Time Section 8 / Wed(9:55-11:45)

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

The Component that provides the waveforms seen in the prelab appears to function as a type of interface between registers and memory module. It uses both a standard cycle clock as well as a separate clock for writing data to the memory locations as it seems to take longer than one standard clock cycle to write data from the registers to the memory but not longer than one cycle to read from memory.

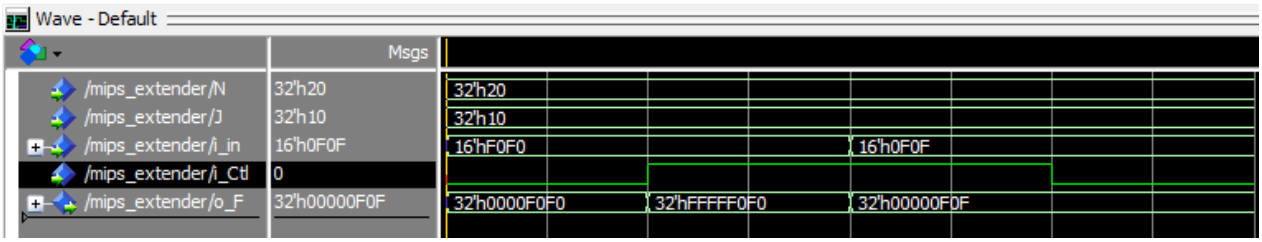
b. [Part 1 (a)] What are the MIPS instructions that require some value to be sign extended? What are the MIPS instructions that require some value to be zero extended?

Sign extension is required when using signed values that are less than 32 bits such as in instructions like lb with signed bytes and lh with signed halfwords.
Zero extension is required when using unsigned values that are less than 32 bits such as in instructions like lbu with unsigned bytes and lhu with unsigned halfwords.

c. [Part 1 (b)] what are the different 16-bit to 32-bit "extender" components that would be required by a MIPS processor implementation?

The first extender component would deal with unsigned values by filling out the remaining bits with zeros to make the value 32 bits while not changing the value.
The second component would deal with signed value by identifying the most significant bit of the value and replicating it in the remaining bits to make the value 32 bits while not changing the value or sign.

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).

i. DATA_WIDTH: This is set to 32 to have the width of each address in the memory be 32 bits. This will allow us to have a 32-bit memory.

ii. ADDR_WIDTH: This is set to 10 to allow us to have 10 addresses for memory. This allows our memory to be made of 10 addresses each 32 bits wide.

iii. CLK: This is the standard system clock to run our instructions.

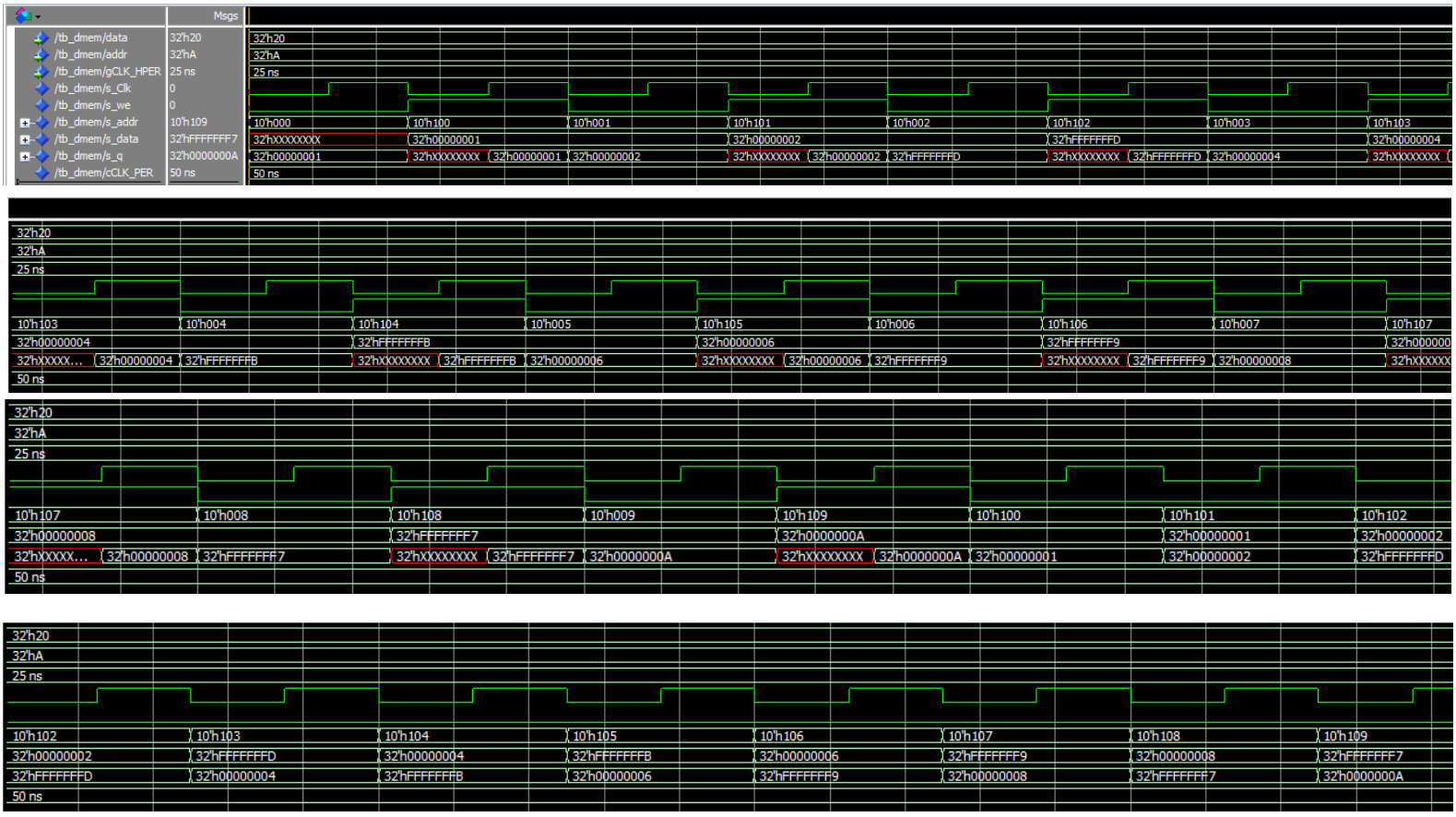
iv. Addr: This is a vector the size of the address width. This allows us to have one address to write to and read from.

v. Data: This is a vector the size of the data width. This acts as the input value to be stored in the memory addresses.

vi. We: this is the write enable signal. It is one bit and it determines if we write to a location of not.

vii. Q: This is the output vector, it is the size of the data width. The value of the output is updated each clock cycle with the value currently stored in the address that Addr is set to. It becomes valid after Addr is set to a location with readable data in it.

f. [Part 2 (c)] Waveform.



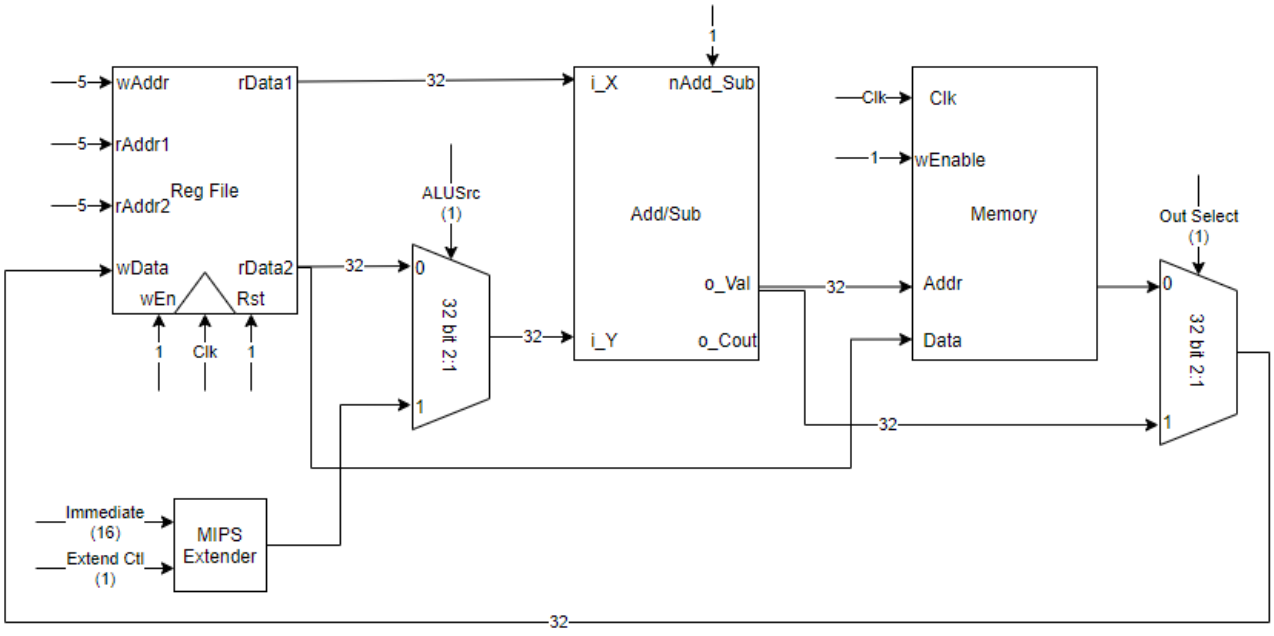
g. [Part 2 (d)] Briefly describe how the waveforms for this mem.vhd module differ from those that you analyzed as part of the pre-lab.

Our waveforms differ in that our write enable is toggled at the falling edge of the clock instead of the rising edge. We also do not have a separate data input signal, we use the output signal of the read process as the data input.

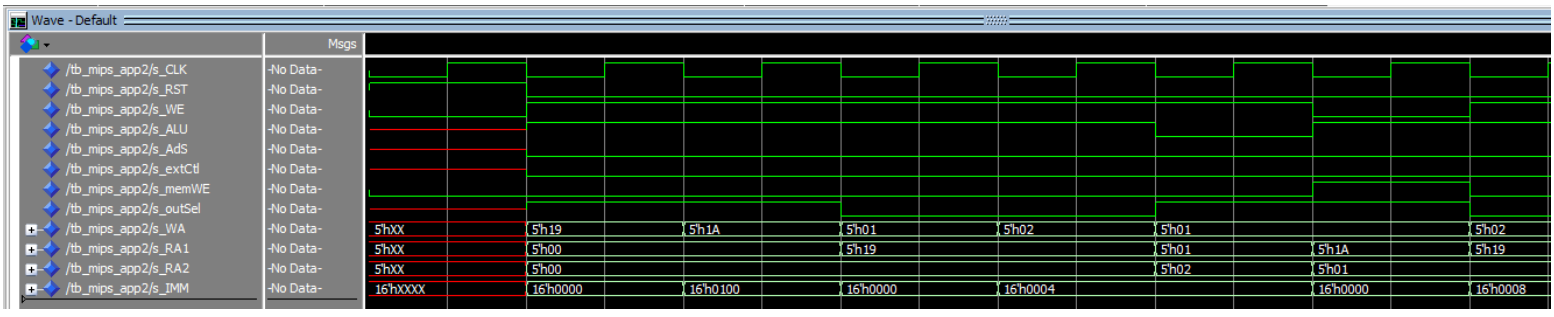
h. [Part 3 (a)] what control signals will need to be added to the simple processor from Lab #3? How do these control signals correspond to the ports on the mem.vhd component analyzed in problem 2)?

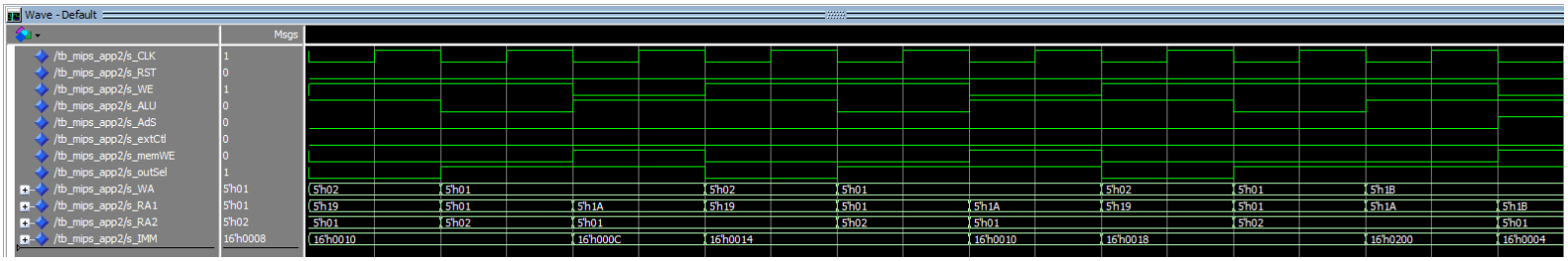
i. Memory write enable: will allow us to control when we want to change values in memory or not.
ii. Output select: will allow us to control if the value from the ALU or memory is what we want to use for output.
iii. Extend control: will allow us to tell if an immediate value being loaded is signed or unsigned.

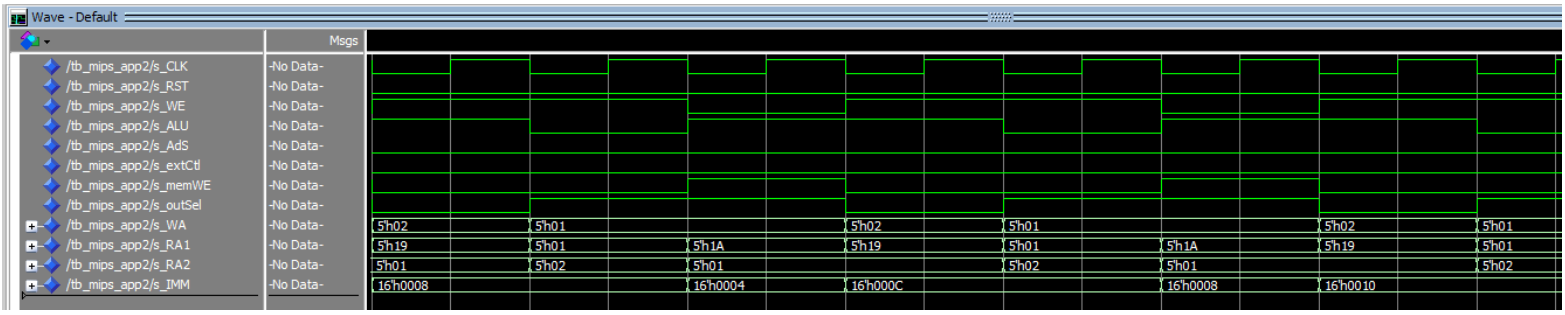
i. [Part 3 (b)] Draw a schematic of a simplified MIPS processor consisting only of the base components used in Lab #3, the extender component described in problem (1), and the data memory from problem (2).



j. [Part 3 (c)] Waveform.







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	.1	.5
Pencil/paper design	0	
VHDL design	1.5	4
Assembly coding	0	
Simulation	.15	1
Debugging	.25	1.5
Report writing	0	.75
Other:	0	0
Total	2	7.75

ii. If you could change one thing about the lab experience, what would it be? Why?

If I could change one thing about the lab I would have a few smaller components dealing with memory before putting it in a complex combination of components working together.

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

The most interesting part for me was also the most frustrating, it took me a good amount of time to get my part 3 processor to function. After I did however it was super interesting making the testbench and seeing just exactly it was capable of.