CprE 381, Computer Organization and Assembly-Level Programming

Lab 2 Report

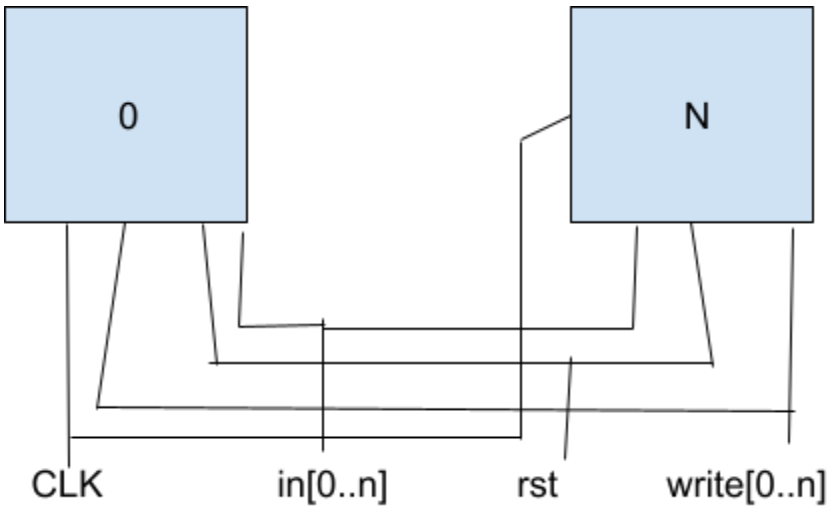
Student Name Sophie Waterman Hines

Submit a typeset pdf version of this on Canvas by the due date. Refer to the highlighted language in the lab document for the context of the following questions.

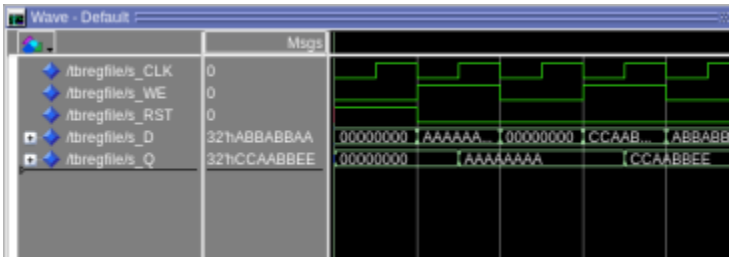
[Part 2 (a)] Draw the interface description for the MIPS register file. Which ports do you think are necessary, and how wide (in bits) do they need to be?

The register file will need two source address inputs (5 bits each), a destination address (another 5 bits), and a 32 bit data in and out. It will also need one reset bit.

[Part 2 (b)] Create an N-bit register using this flip-flop as your basis.



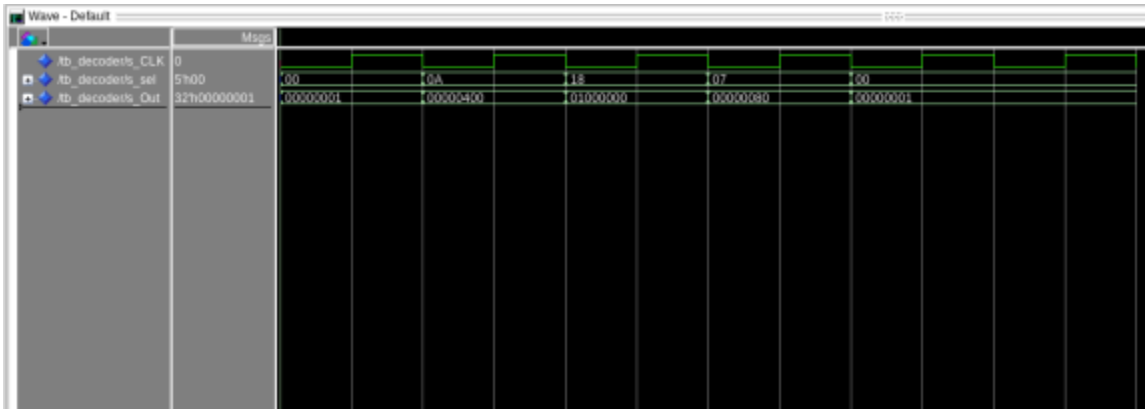
[Part 2 (c)] Waveform.



[Part 2 (d)] What type of decoder would be required by the MIPS register file and why?

A 5:32 bit decoder for selecting a 32 bi register's write enable.

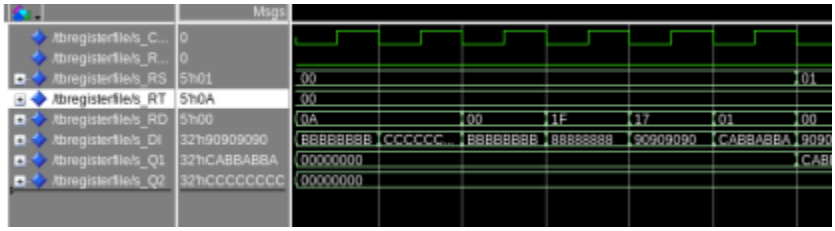
[Part 2 (e)] Waveform.



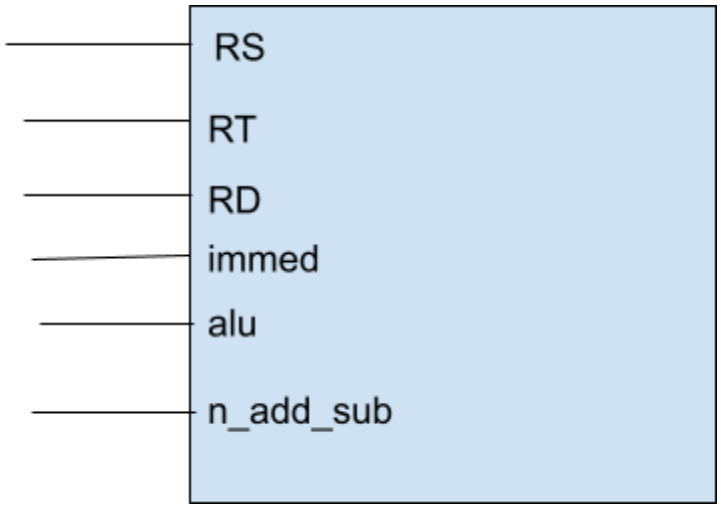
[Part 2 (f)] In your write-up, describe and defend the design you intend on implementing for the next part.

I created a 32x32 bus that input into a dataflow mux and then output one 32 wide logic vector. Made my life a lot easier.

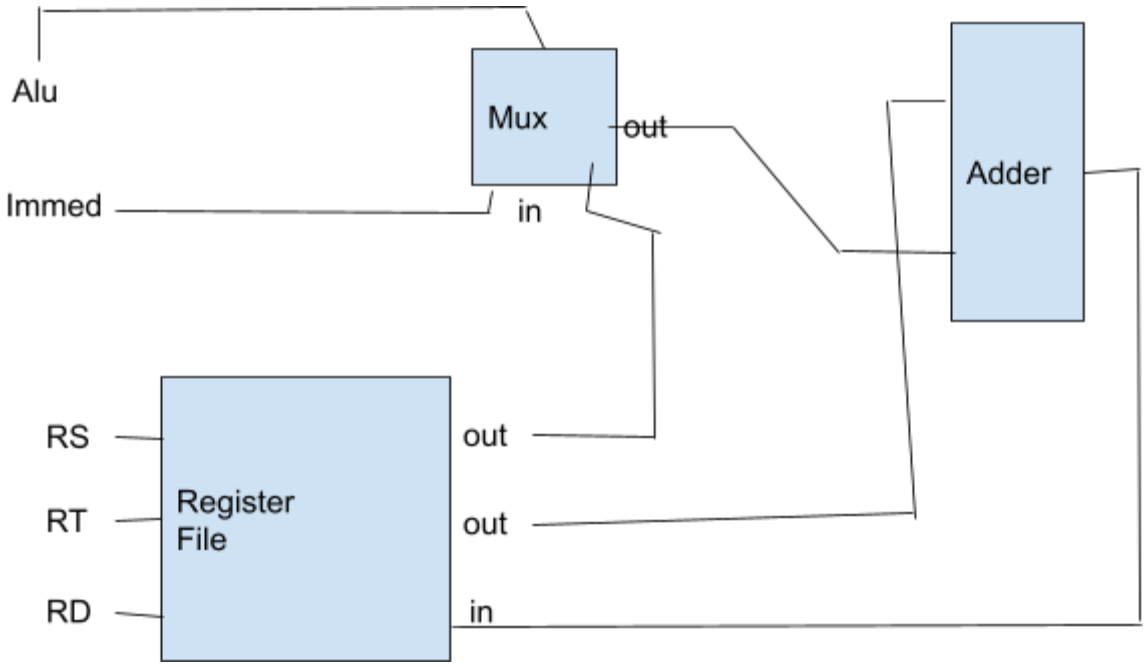
[Part 2 (i)] Waveform.



[Part 3 (b)] Draw a symbol for this MIPS-like datapath.



[Part 3 (c)] Draw a schematic of the simplified MIPS processor datapath consisting only of the component described in part (a) and the register file from problem (1).



[Part 4 (a)] Read through the mem.vhd file, and based on your understanding of the VHDL implementation, provide a 2-3 sentence description of each of the individual ports (both generic and regular).

Clk is the clock, and tells the memory when it's time to update. The addr port is the location where memory is tampered with (load or store). The data port is what needs to be written at the addr location. The we port is alway 1, and port q is the output of memory at the addr location. This port is used for loading.

[Part 4 (c)] Waveforms.

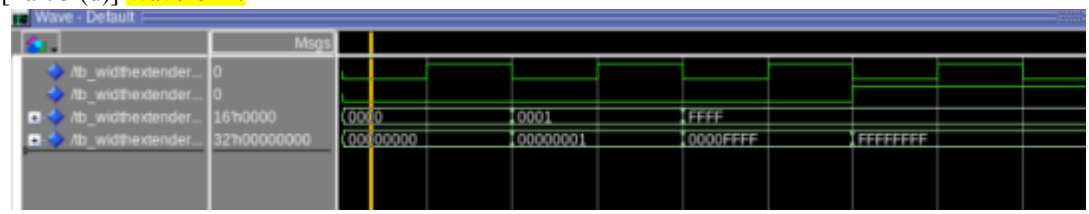
[Part 5 (a)] What are the MIPS instructions that require some value to be sign extended? What are the MIPS instructions that require some value to be zero extended?

An example of an instruction that needs to be sign extended is the addi operator, since it takes in a 16 bit value and needs it to be extended to preform operations on it. The same can also be said abou addiu.

[Part 5 (b)] what are the different 16-bit to 32-bit “extender” components that would be required by a MIPS processor implementation?

An “and” gate was used and a signal that constantly remained at 1 to copy the 16 bit to the 32 bit output.

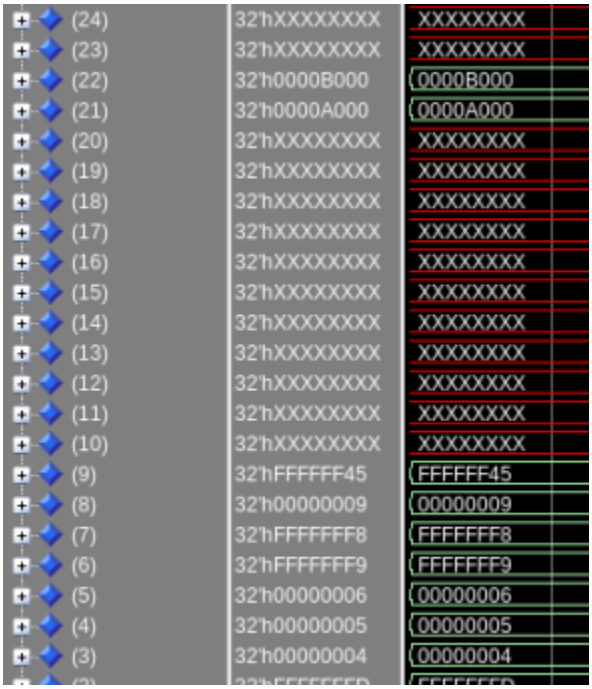
[Part 5 (d)] Waveform.

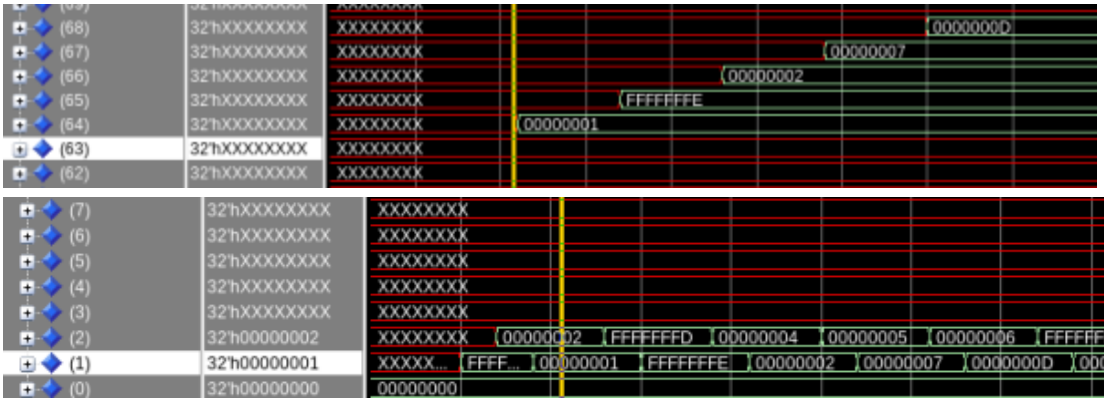


[Part 6 (a)] what control signals will need to be added to the simple processor from part 2? How do these control signals correspond to the ports on the mem.vhd component analyzed in part 3?

Memory writing control, add controller, load controller, and a sign control for the extender.

[Part 6 (c)] Waveform.




(68) 32'hXXXXXXXX XXXXXXXX 0000000D
(67) 32'hXXXXXXXX XXXXXXXX 00000007
(66) 32'hXXXXXXXX XXXXXXXX 00000002
(65) 32'hXXXXXXXX XXXXXXXX FFFFFFFE
(64) 32'hXXXXXXXX XXXXXXXX 00000001
(63) 32'hXXXXXXXX XXXXXXXX
(62) 32'hXXXXXXXX XXXXXXXX
(7) 32'hXXXXXXXX XXXXXXXX
(6) 32'hXXXXXXXX XXXXXXXX
(5) 32'hXXXXXXXX XXXXXXXX
(4) 32'hXXXXXXXX XXXXXXXX
(3) 32'hXXXXXXXX XXXXXXXX
(2) 32'h00000002 XXXXXXXX 00000002 FFFFFFFD 00000004 00000005 00000006 FFFFFFF9
(1) 32'h00000001 XXXXX... FFFF... 00000001 FFFFFFFE 00000002 00000007 0000000D
(0) 32'h00000000 00000000