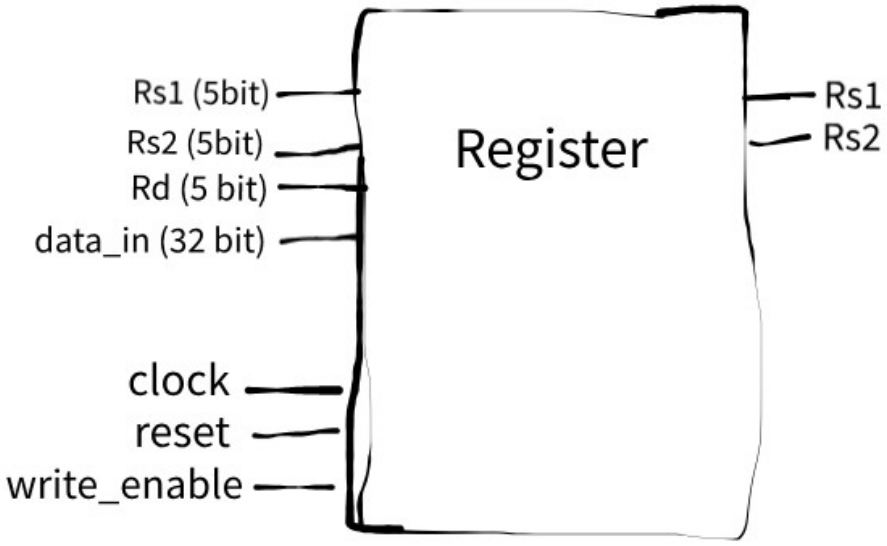
CprE 3810, Computer Organization and Assembly-Level Programming

Lab 2 Report

Student Name Cai Chen

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.

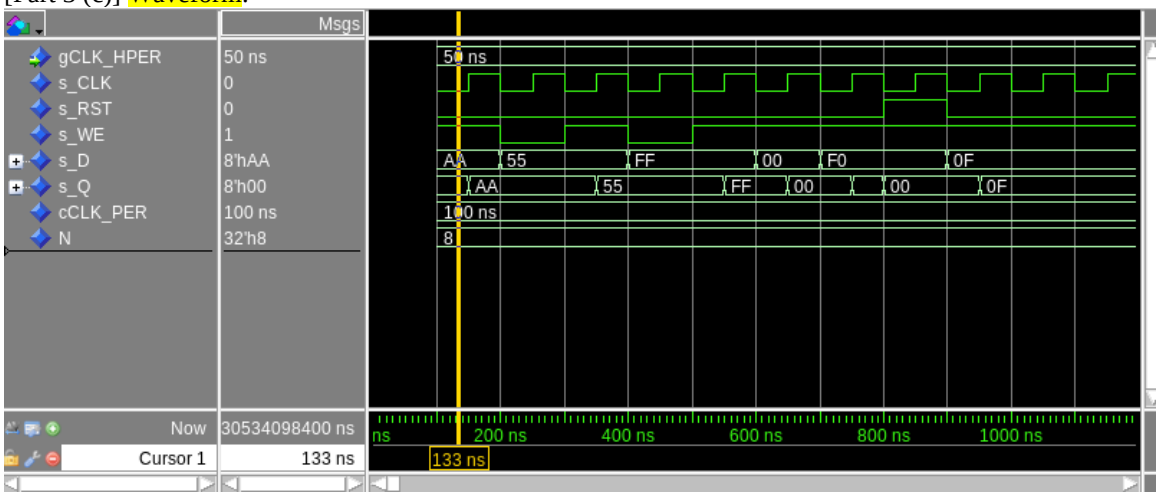
[Part 3 (a)] Draw the interface description (i.e., the "symbol" or high-level blackbox) for the RISC-V register file. Which ports do you think are necessary, and how wide (in bits) do they need to be?



[Part 3 (b)] Create an N-bit register using this flip-flop as your basis.

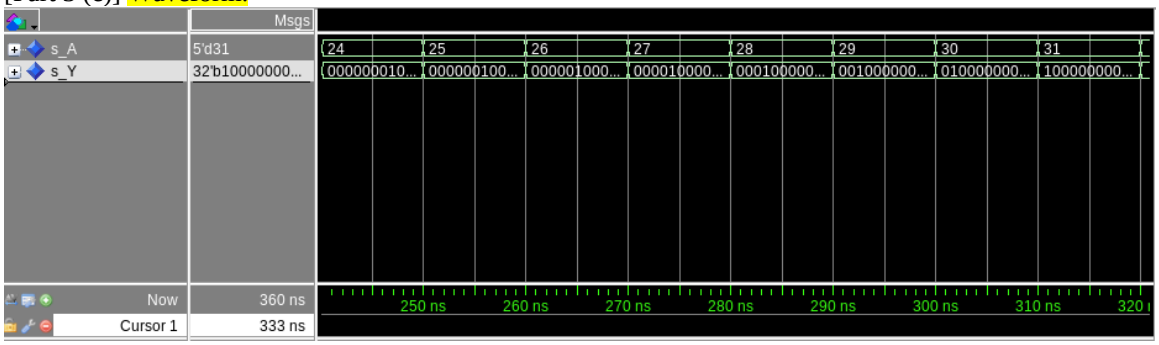
N/A

[Part 3 (c)] Waveform.



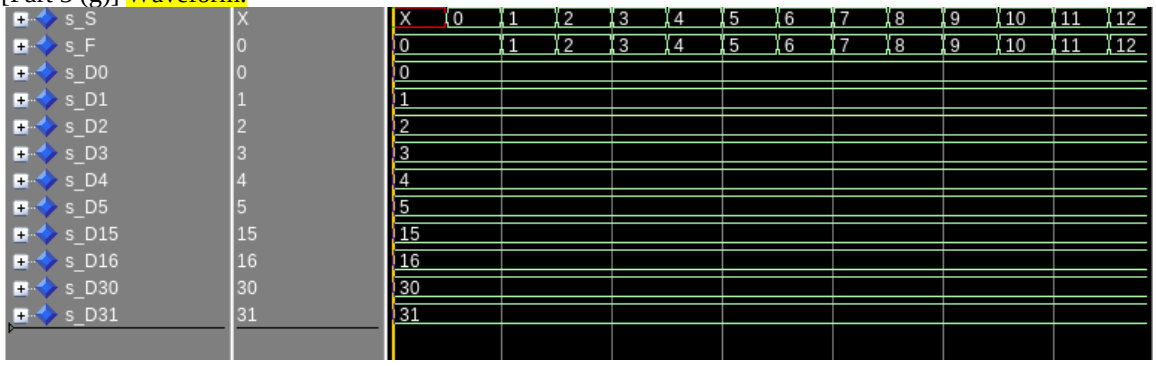
[Part 3 (d)] What type of decoder would be required by the RISC-V register file and why?
A 5:32 decoder because RISC-V has 32 registers that need to be individually addressable using 5-bit addresses

[Part 3 (e)] Waveform.

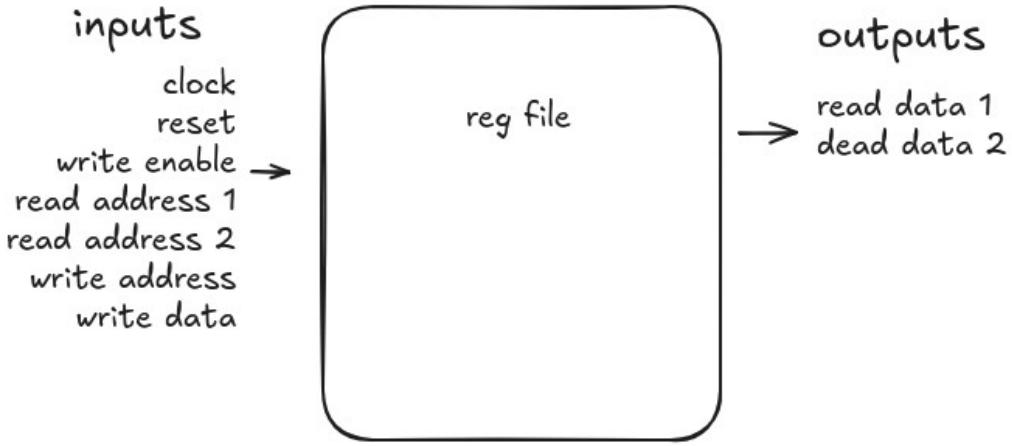


[Part 3 (f)] In your write-up, describe and defend the design you intend on implementing for the next part.
I used VHDL's with-select-when dataflow approach because it creates the fastest possible multiplexer with only one logic level delay. This is much better than building a tree of smaller multiplexers, which would be slower and use more resources. The dataflow code is also clean and easy to read.

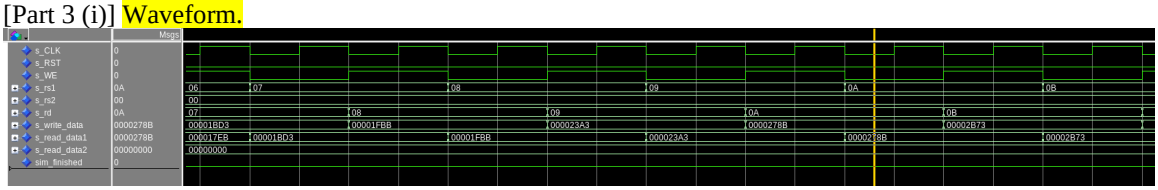
[Part 3 (g)] Waveform.



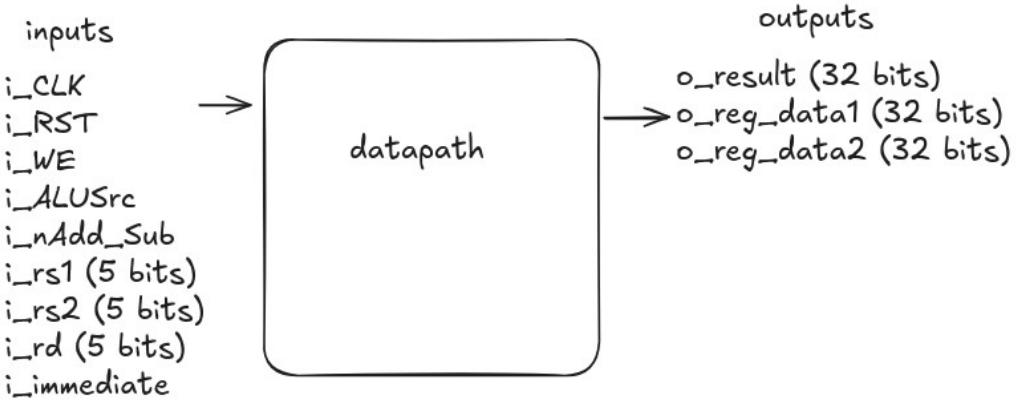
[Part 3 (h)] Draw a (simplified) schematic (i.e., components within the high-level blackbox) for the RISC-V register file, using the same top-level interface ports as in your solution describe above and using only the register, decoder, and mux VHDL components you have created.



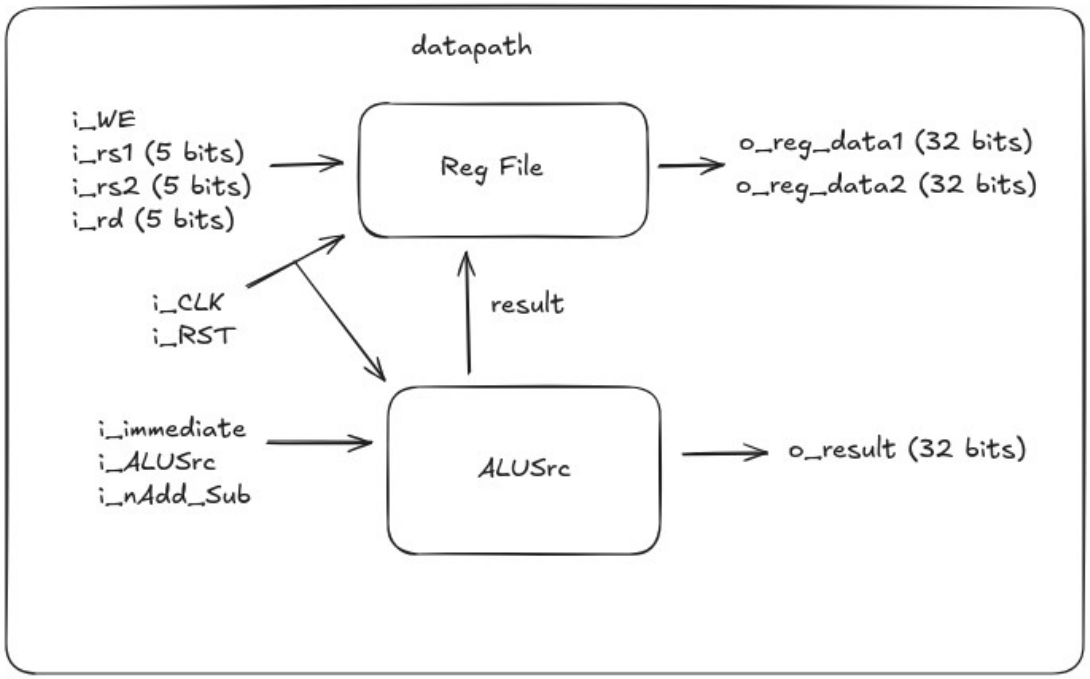
[Part 3 (i)] Waveform.



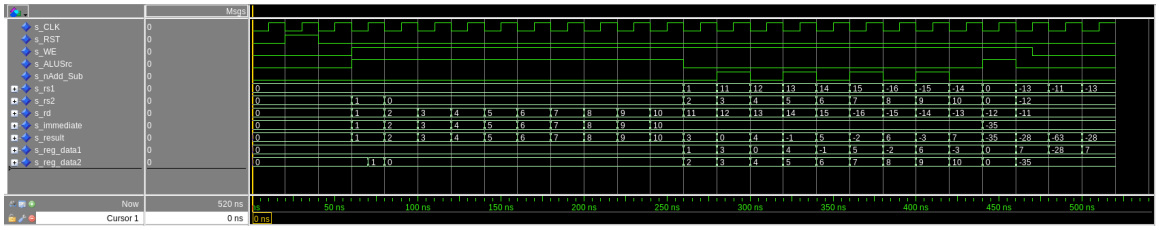
[Part 4 (b)] Draw a symbol for this RISC-V-like datapath.



[Part 4 (c)] Draw a schematic of the simplified RISC-V processor datapath consisting only of the component described in part (a) and the register file from problem (1).



[Part 4 (d)] Include in your report waveform screenshots that demonstrate your properly functioning design. Annotate what the final register file state should be.

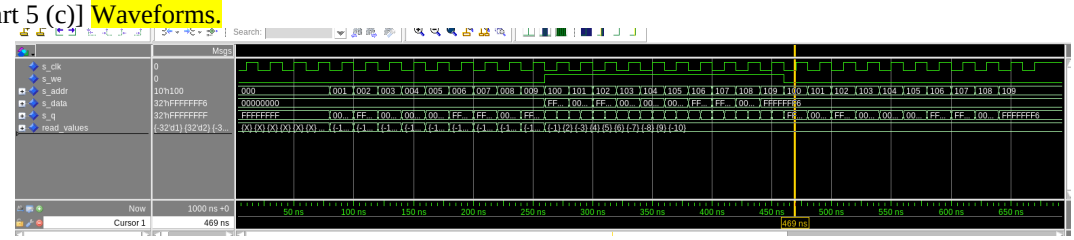


Final state: x21 = -28

[Part 5 (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).

DATA_WIDTH: Defines the width of each memory word as 32 bits
ADDR_WIDTH: Creates the addresses that memory can be stored in. 2^10 addresses total
clk: rising edge clock
addr: location for read & writing
data: data to be writen
we: enables write
q: data output port that gives data from the specific spot from addr. Output is asynchronous so it updates instantly whenever the address changes

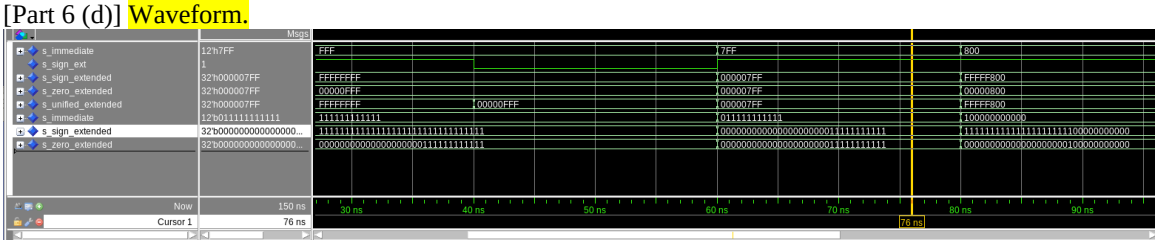
[Part 5 (c)] Waveforms.



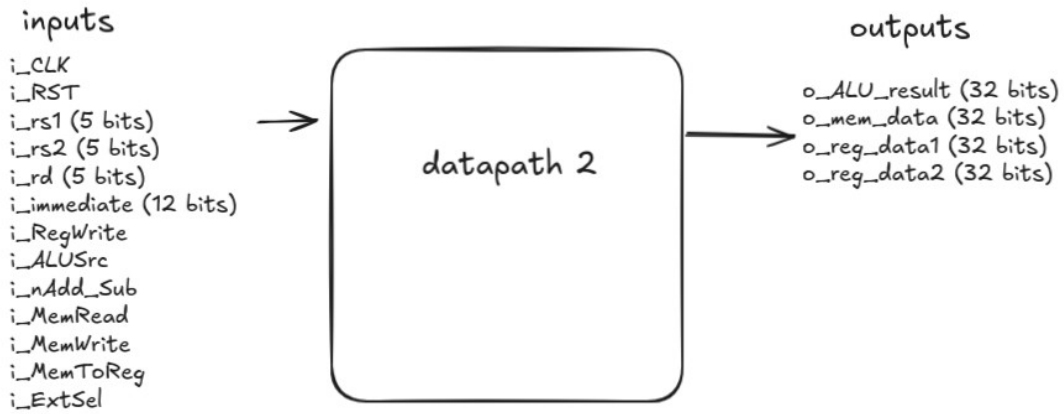
[Part 6 (a)] What are the RISC-V instructions that require some value to be sign extended? What are the RISC-V instructions that require some value to be zero extended?
Sign extended: ADDI, SLTI, LW, LH, LB
Zero extended: ANDI, ORI, XORI, SLTIU

[Part 6 (b)] what are the different 16-bit to 32-bit "extender" components that would be required by a RISC-V processor implementation?
sign extender and zero extender so all components that use those commands such as add, load, and, etc.

[Part 6 (d)] Waveform.



[Part 7 (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?



[Part 7 (b)] Draw a schematic of a simplified RISC-V processor consisting only of the base components used in part 2, the extender component described in part 4, and the data memory from part 3.

[Part 7 (c)] Waveform.