**HW#01: Register File, Arithmetic Logic Unit, and Multiplier**

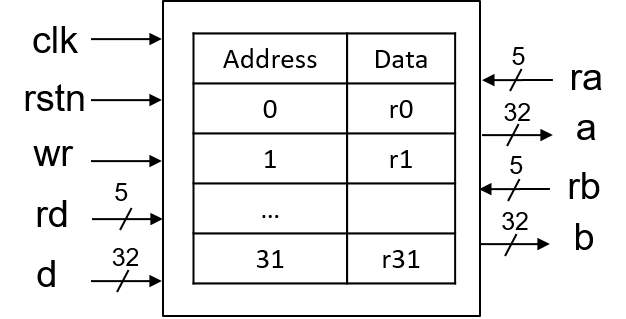
**Issued:** March 14 (Tue.), 2023 **Due:** March 20 (Mon.), 2023

**What to turn in**: Copy the text from your **MODIFIED** codes and paste it into a document. If a question asks you to plot or display something on the screen, also include the plot and screen output your code generates. Submit either a \*.doc or \*.pdf file.

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

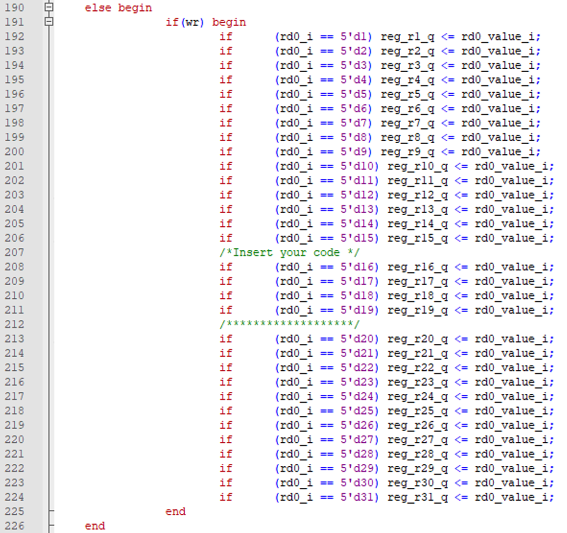
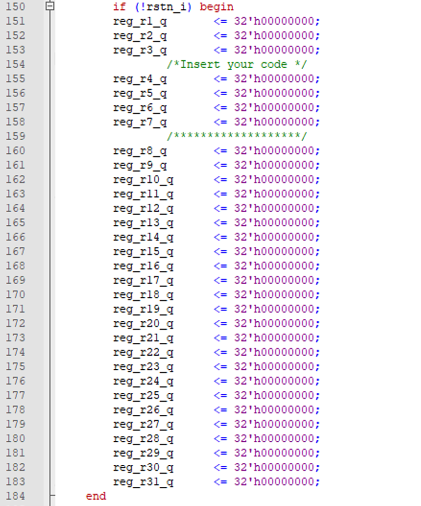
**Problem 1 (15p): Register File**

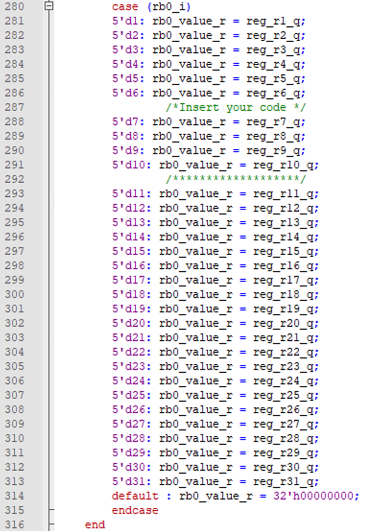
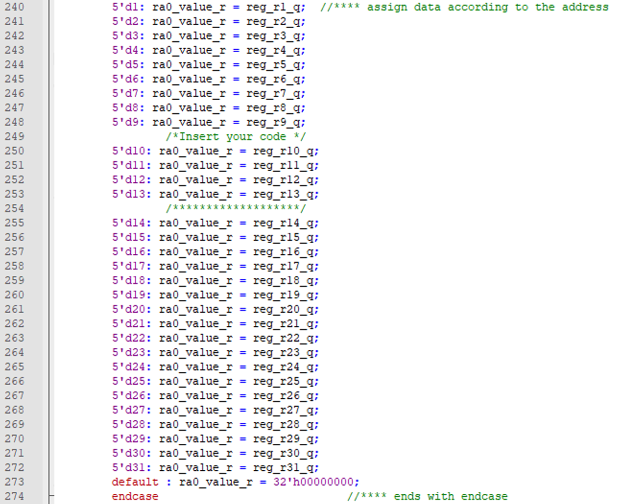
Design a RISC-V register file (RF) in Verilog. Please see the detailed description in the lecture note.

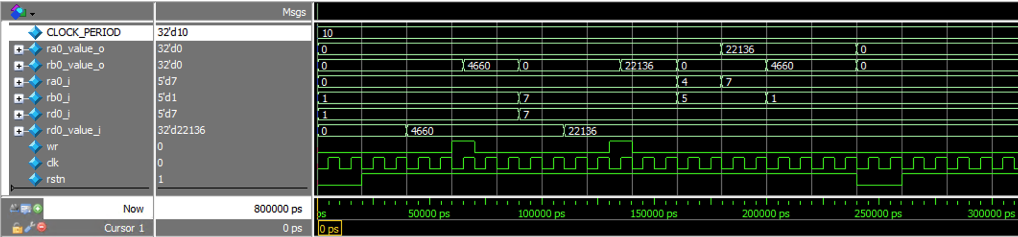


What you have to do:

* Design an RF and its test bench based on the baseline codes.
* Submit your RTL files (riscv\_regfile.v and riscv\_regfile\_tb.v)
* Capture the waveform.

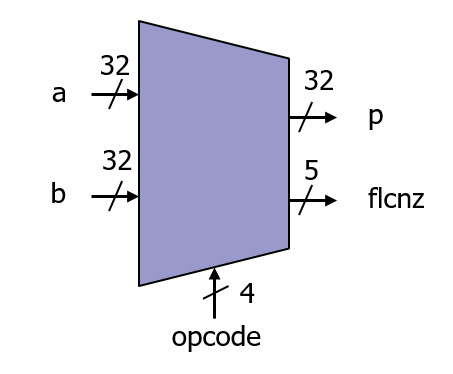
****

****

****

**Problem 2 (15p): Arithmetic Logic Unit (ALU)**

Design a RISC-V ALU in Verilog. Please see the detailed description in the lecture note.



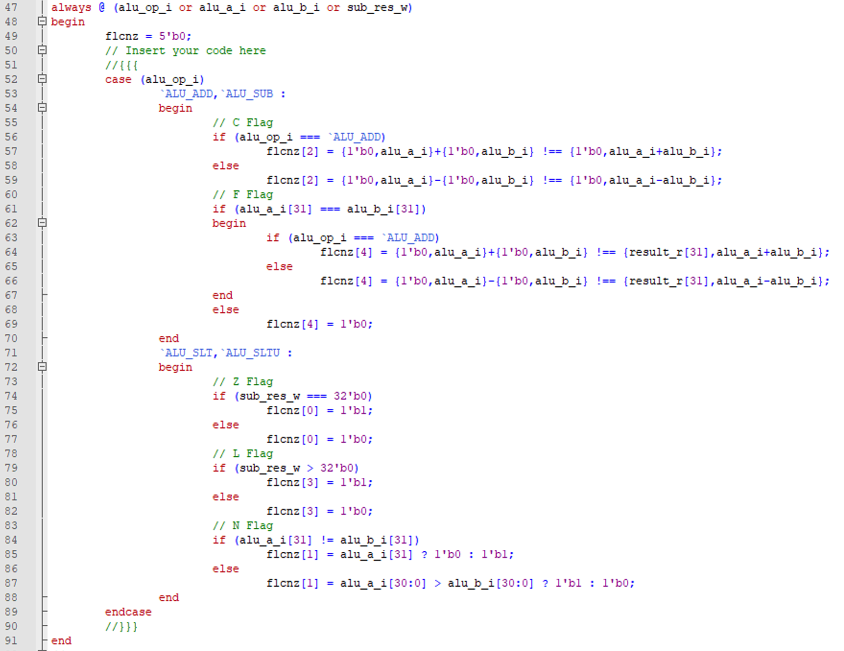
What you have to do:

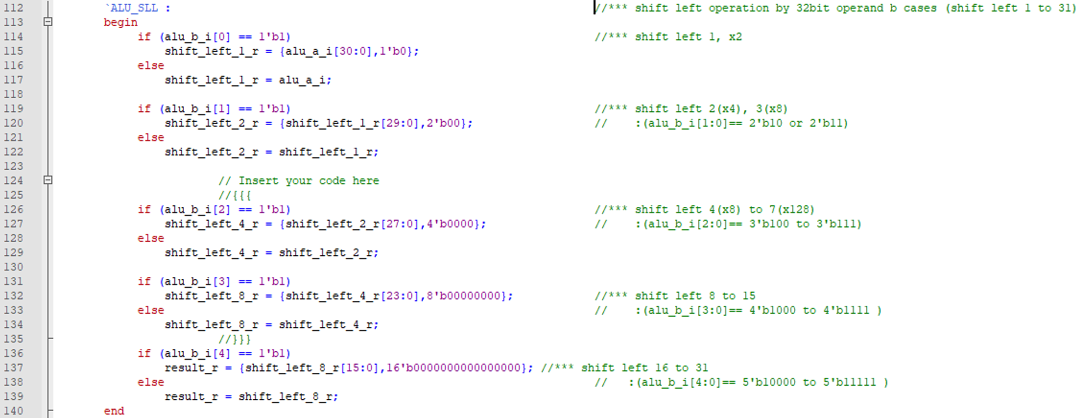
* Design an ALU with FLCNZ and its test bench based on the baseline codes.
* Submit your RTL files (riscv\_alu.v and riscv\_alu\_tb.v).
* Capture the waveform.

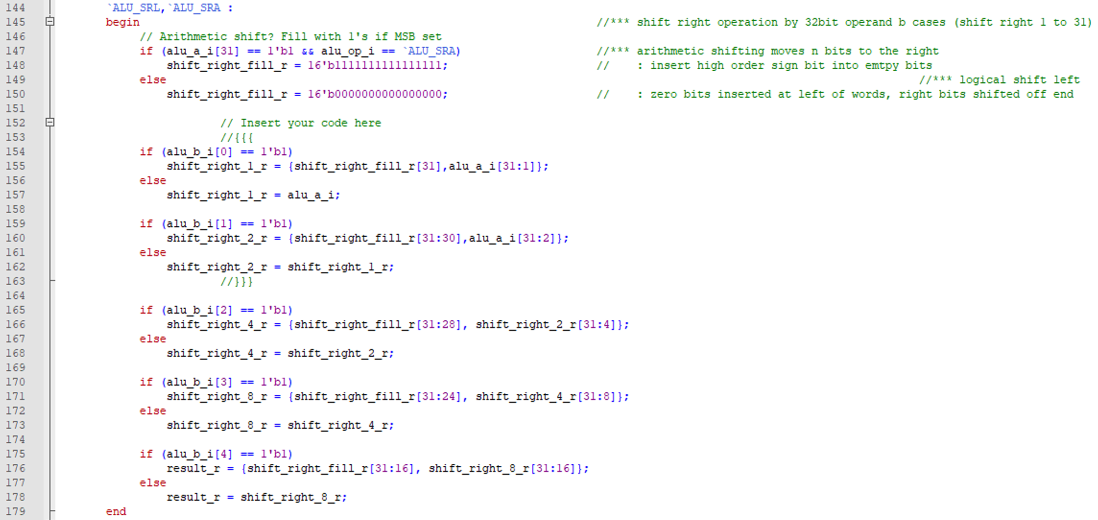
The FLCNZ flags are defined as follows:

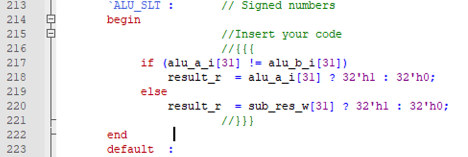
|  |  |  |  |
| --- | --- | --- | --- |
| Flag | Opcode | Description | Remarks |
| C | ADD, SUB | Set C to HIGH if a carry/borrow from the most significant bit (MSB) position occurs when the operands are treated as unsigned numbers. | Arithmetic |
| F | ADD, SUB | Set F to HIGH if an overflow occurs when the operands are treated as two’s complement numbers (signed numbers). | Arithmetic |
| Z | SLT, SLTU | Perform a subtraction between Rsrc1 (a) and Rsrc2 (b) without writing back to Rdest (p) and set Z to HIGH if the result is zero | Comparison |
| L | SLT, SLTU | Perform a subtraction between Rsrc1 (a) and Rsrc2 (b) without writing back to Rdest (p) and set L to HIGH if a > b when the operands are treated as unsigned numbers. | Comparison |
| N | SLT, SLTU | Perform a subtraction between Rsrc1 (a) and Rsrc2 (b) without writing back to Rdest (p) and set the N flag if a > b when the operands are treated as signed numbers. | Comparison |

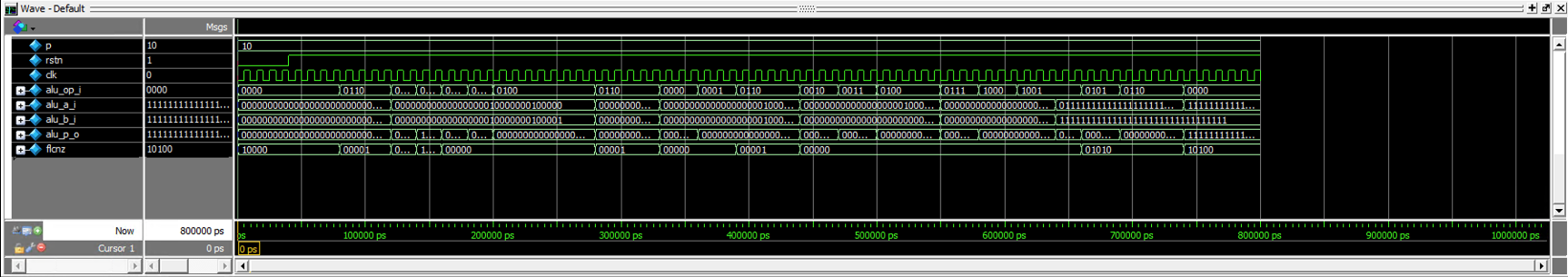
SLT = Smaller-or-Less-Then, SLTU = Smaller-or-Less-Than for Unsigned numbers

****

****

****

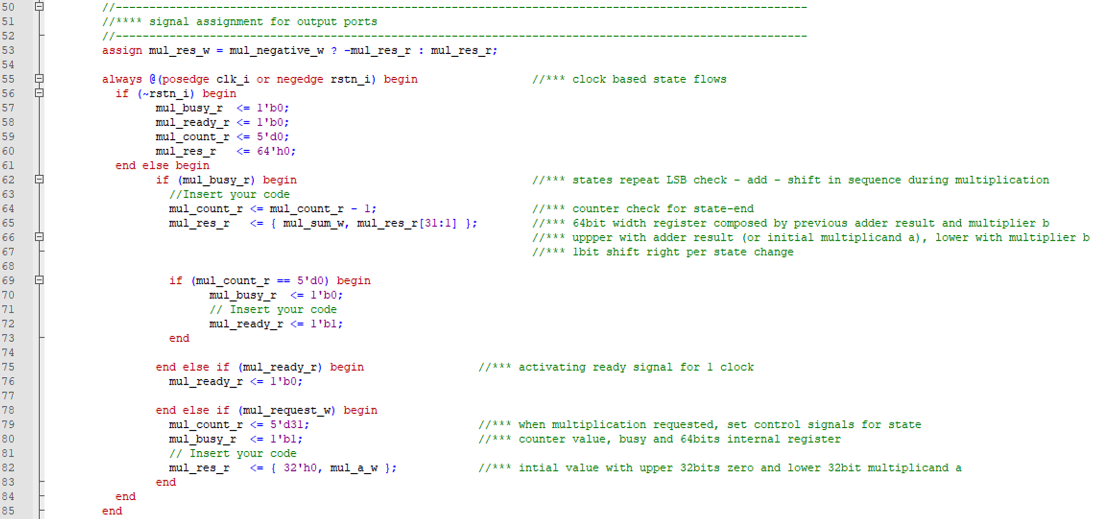
****

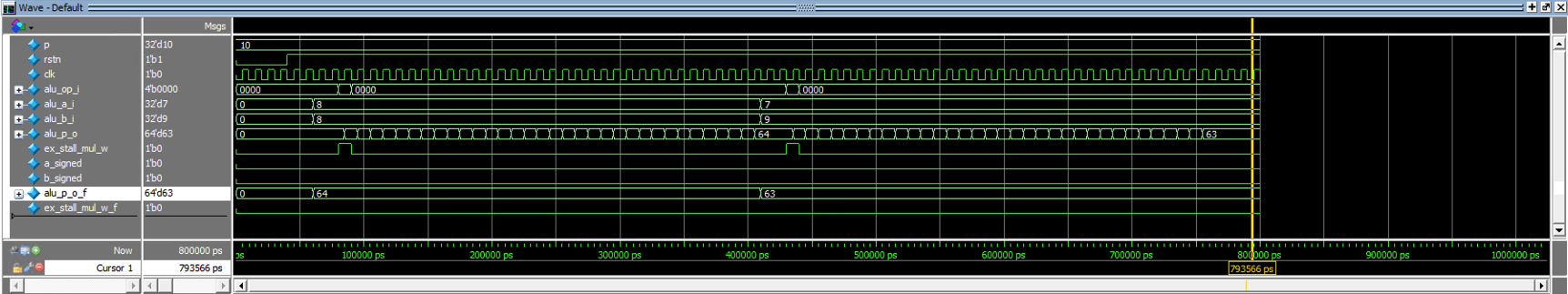
****

**Problem 3 (5p): Multiplier**

Design a sequential 32bx32b multiplier in Verilog. Please see the detailed description in the lecture note. What you have to do:

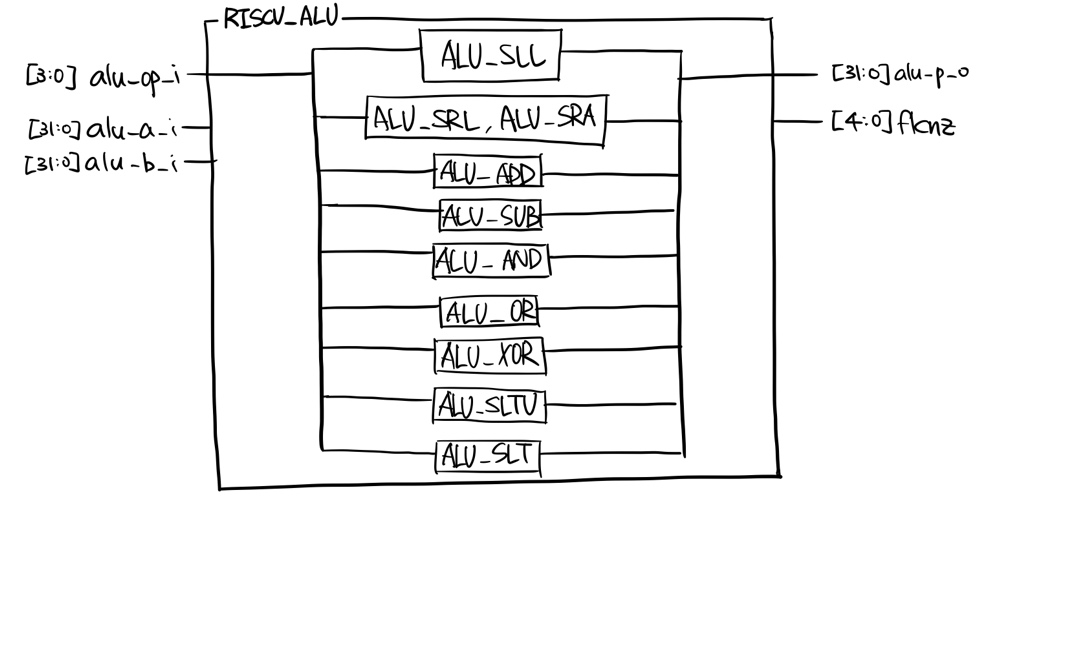
* Design a 32-bit multiplier and its test bench based on the baseline codes.
* Submit your RTL files (riscv\_multiplier.v and riscv\_multiplier\_tb.v).
* Capture the waveform.



****

**Problem 4 (3p): (Optional) Bonus**

1. (1p) Assume that each operation in ALU is described in a block. Draw a block diagram of ALU.



1. (2p) In problem 3, a slow sequential multiplier completes a 32-bit multiplication in 32 cycles. Is a computation result correct if the input is changed during computation? Modify the code to make the computation result correct even though input is changed during computation. Show the test cases and the waveform corresponding to both the baseline code and a modified code.

i) Is a computation result correct if the input is changed during computation?

: No. Since “mul\_b\_w” wire variable is assigned by condition operator with the input and it is used in multiplication, if the 2nd input(id\_rb\_value\_r) is changed, the computation result will not be correct.

ii) Modify the code.

Modified multiflier : riscv\_multiplier\_modified.v

Modified testbench : riscv\_multiplier\_modified\_tb.v

iii) Show the test cases and the waveform corresponding to both the baseline code and a modified code.

Test case #1 : alu\_a\_i, alu\_b\_i = 8

Test case #2 : alu\_a\_i = 7, alu\_b\_i = 9 -> alu\_a\_i = 65527, alu\_b\_i = 16777208 ( after 5\*period of clock )

