ECE 6913, Computing Systems Architecture

Fall 2020 NYU ECE

Please fill in your name: ___________________________________

Final, December 18th 2020

Maximum time: 150 minutes : 9:45 **AM** - **12:15 PM ET** [+ 15 minutes to assemble PDF and upload]

Open Book, Open Notes,

Calculators allowed.

Must show your work in steps – to get any credit

This is NOT a group project You may NOT discuss, share your Quiz solutions with anyone else.

You must stay logged in to Zoom throughout the Quiz, with Camera on

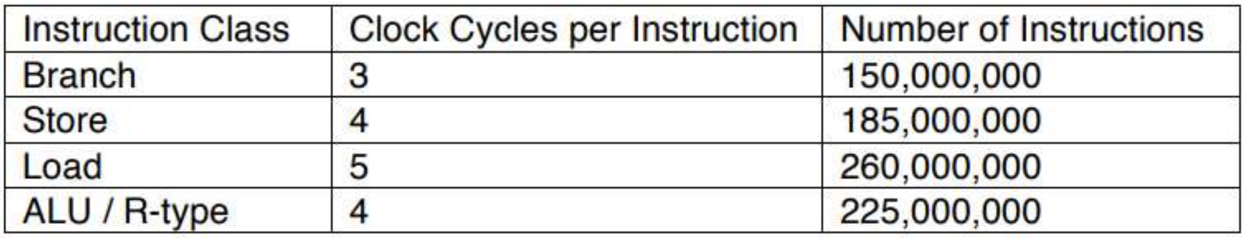
Instructor available online if you have questions on the Quiz, during the Quiz – enter question in Zoom chat box at any time during Quiz

This Test has 4 problems. Please attempt all of them. Please show **all work**. Please write **legibly**

1. Please be sure to have 5-10 sheets of white or ruled paper & a Pencil, Eraser
2. Write down your solutions on 8.5 x 11 sheets of white paper, single sided with your name printed in top right corner of each sheet and with **Page Number and Problem number identified clearly on each sheet**
3. **Stop working on your Quiz at 12:15 PM** – you have 10 minutes to scan/take pictures of each sheet and upload them as completed PDF assignment to NYU Classes – you may use any of several smartphone apps to integrate your scans/pictures of sheets into a PDF file
4. Take pictures of each sheet and **upload** the PDF of all sheets after checking you have all sheets in the right order **by 12:30 PM latest.**
5. You may use iPAD to write down your solutions directly rather than on paper
6. Portal **will close at 12:30 PM** not allowing upload of your quiz after this time

Problem 1.

Assume that to spell check a large file, 820,000,000 instructions are needed. The instructions in the program are broken down into 4 different classes, and each class requires N clock cycles to execute. Specific information is given in the table below. (Here, N is the same as in the RISC-V multi-cycle datapath discussed in class.)



Instruction Class	Clock Cycles per Instruction	Number of Instructions
Branch	3	150,000,000
Store	4	185,000,000
Load	5	260,000,000
ALU / R-type	4	225,000,000

(i) If the total execution time for this program is found to be 1.57 seconds, what is the clock cycle time of the computer on which it was run?

(ii) Assume that as part of the 820,000,000-instruction spell check, 25% of all load instructions are immediately followed by an ALU / R-type instruction that uses the data that was just loaded. To speed up this program, we are contemplating adding a new type of instruction – an ALU instruction where one of the source operands is a value from memory.

- This new instruction will replace the previous, 2 instruction sequence.

- It will take 7 clock cycles.

Will this change offer any speedup over the original design? If so, how much?

You may assume that the clock rate does not change and your answer to this question does not depend on your answer to (i)

(iii) Qualitatively, if you see a speedup, where does it come from? If you do not, why not?

Problem 2.

A. This question considers the basic, RISC-V, 5-stage pipeline. (In this problem, you may assume that there is full forwarding.)

(i) Explain how pipelining can improve the performance of a given instruction mix

(ii) Show how these instructions will flow through the pipeline:

	1	2	3	4	5	6	7	8	9	10	11
lw x10, 0(x11)											
add x9, x11, x11											
sub x8, x10, x9											
lw x7, 0(x8)											
sw x7, 4(x8)											

(iii) Where might the sw instruction get its data from? Be very specific. (i.e. "from the lw instruction" is not a good answer!)

B. This question considers the basic, RISC-V, 5-stage pipeline. (In this problem, you may assume that there is full forwarding.) Show how these instructions will flow through the pipeline below. For the instruction mix above, on what instruction results does the last add instruction depend on? predict that the `beq` instruction is not taken

	1	2	3	4	5	6	7	8	9	10	11
beq x1, x2, X											
lw x10, 0($11)											
sub x14, $10, $10											
X: add x4, x1, x2											
lw x1, 0(x4)											
sub x1, x1, x1											
add x1, x1, x1											

Problem 3.

(i) A cache may be organized such that:

- In one case, there are more data elements per block and fewer blocks
- In another case, there are fewer elements per block but more blocks However, in both cases – i.e. larger blocks but fewer of them OR shorter blocks, but more of them – the cache's total capacity (amount of data storage) remains the same

What are the pros and cons of each organization? Support your answer with a short example assuming that the cache is direct mapped

(ii) Assume:

- A processor has a direct mapped cache
- Data words are 8 bits long (i.e., 1 byte)
- Data addresses are to the word
- A physical address is 20 bits long
- The tag is 11 bits
- Each block holds 16 bytes of data

How many blocks are in this cache?

(iii) Consider a 16-way set-associative cache:

- Data words are 64 bits long

- Words are addressed to the half-word

- The cache holds 2 Mbytes of data

- Each block holds 16 data words

- Physical addresses are 64 bits long

How many bits of tag, index, and offset are needed to support references to this cache?

Problem 4.

Implement in RISC V these line of code in C:

(i) `f = g - A[B[C[64]]]`

(ii) `f = g - A[C[16] + B[32]]`

(iii) `A[i] = 4B[8i-81] + 4C[32i+32]`

Assume a 64-bit machine. Assume that the variables f, g, h, i, and j are assigned to registers x5, x6, x7, x28, x29 respectively. Assume base address in memory of Array data structures ‘A, B, C’ (or address in memory of ‘A[0]’, ‘B[0]’and ‘C[0]’) are stored in Registers x27, x30, x31. Write RISCV code that implements