CS/ECE 552: Introduction to Computer Architecture
Mock Midterm 2022

CLOSED BOOK

ONE 8.5”x11” SHEET OF NOTES (TWO-SIDED) ALLOWED

NAME: ____________________________

DO NOT OPEN EXAM UNTIL TOLD TO DO SO!

Read over the entire exam before beginning. You should verify that your exam includes all of the problems listed in the table below as well as two spare pages for scratch work. Budget your time according to the weight of each question and your ability to answer them. I have provided a large amount of space for you to write your answers, but if that space is not sufficient, please write on the BACK of the SAME sheet. WRITE YOUR NAME ON EACH SHEET. You will turn in your cheat sheet with your exam, so please make sure your name is written on it.

Upon announcement of the end of the exam, stop writing on the exam paper immediately. Pass the exam to the head of the tables to be picked up by the proctor (the TAs and me). The instructor will announce when to leave the room. Failure to follow instructions may result in forfeiture of your exam and will be handled according to the UW Academic Misconduct policy.

Problem	Possible Points	Points
Problem 1	10	
Problem 2	10	
Problem 3	15	
Problem 4	30	
Problem 5	10	
Total	75	

Name: ____________________________

Problem 1 [10 points]

Part A [5 point]

Given a program *P*, and a processor *Q*, you are given two choices for running P on Q (assume Q has support for both of these options):

- Option 1: Run *P* at 500 KHz, which provides a CPI of 2.25
- Option 2: Run *P* at 200 KHz, which provides a CPI of 1.5

Given these options, which is the better choice? To receive credit you must justify your answer (NOTE: you may assume K = 1000 or 1024 for your answer)

Part B [5 points]

If a MIPS design has a buggy sign extension unit, name five instructions that will produce incorrect outcomes? Justify your answer in one line for each instruction. [4 points]

Name: ___________________________

Problem 2 [10 points]

Part A [5 points]

Convert the following C code to its equivalent MIPS assembly code:

```
i= 0;
do {
        Z[i] = (A + X[i]) ^ Y[i];
        ++i;
} while (i < N);
```

Assume that \$s0 initially holds X[0] (base address of X), \$s1 initially holds Y[0], \$s2 initially holds Z[0], \$s3 initially holds i, \$s4 initially holds N, and \$s5 initially holds A. Moreover, assume that i has been initialized to 0, and that the arrays are all arrays of integers (4B per element) and have been properly allocated before this code snippet.

Name: ____________________________

Part B [5 points]

Given the following four 32-bit binary numbers, what is the corresponding MIPS assembly code?

```
0000 0010 0001 0001 1000 0000 0010 0000

0000 0000 0001 0000 1000 1000 0010 0010

0001 0110 0001 0001 0000 0000 0000 0100

0010 0010 0001 0000 1111 1111 1111 1100
```

Name: ___________________________

Problem 3 [15 points]

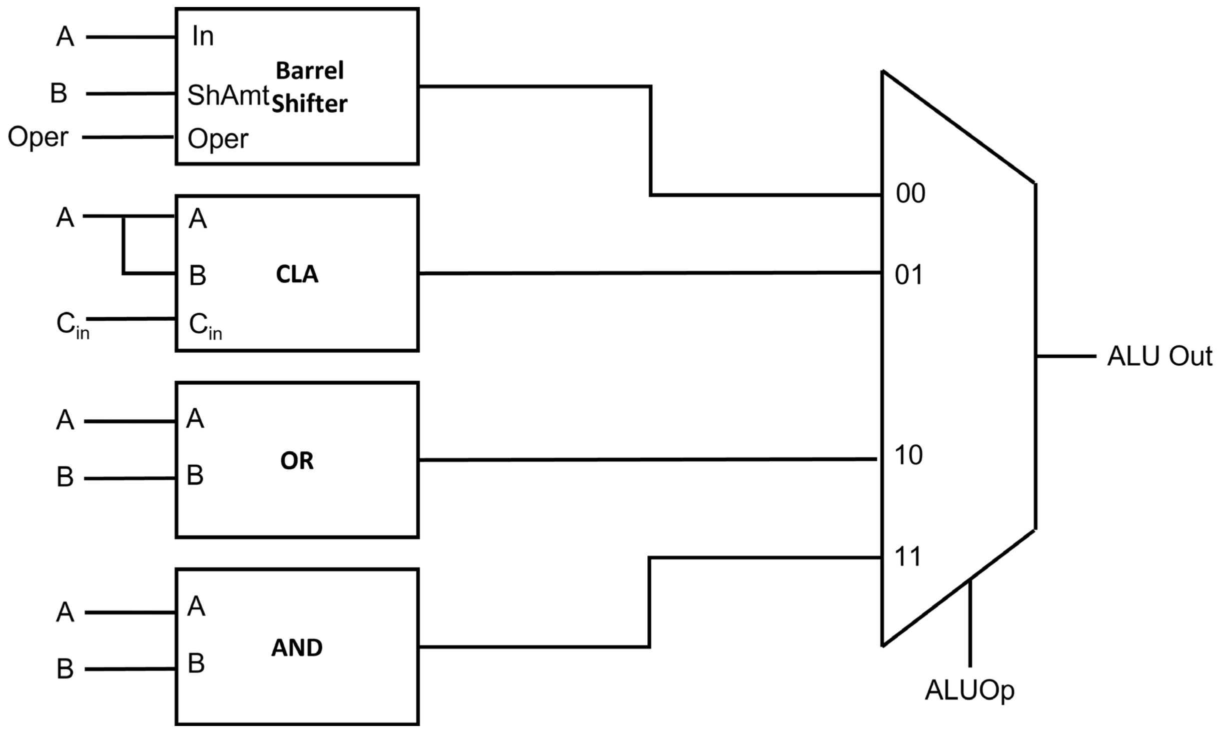
Part A [10 points]

In terms of gate delay, *quantitatively* calculate, compare, and contrast the performance for a 16-bit carry lookahead adder (CLA) **without ripple** and a 16-bit CLA **that accidentally ripples between every bit**. To receive credit you must explain your answer.

Name: ___________________________

Part B [5 points]

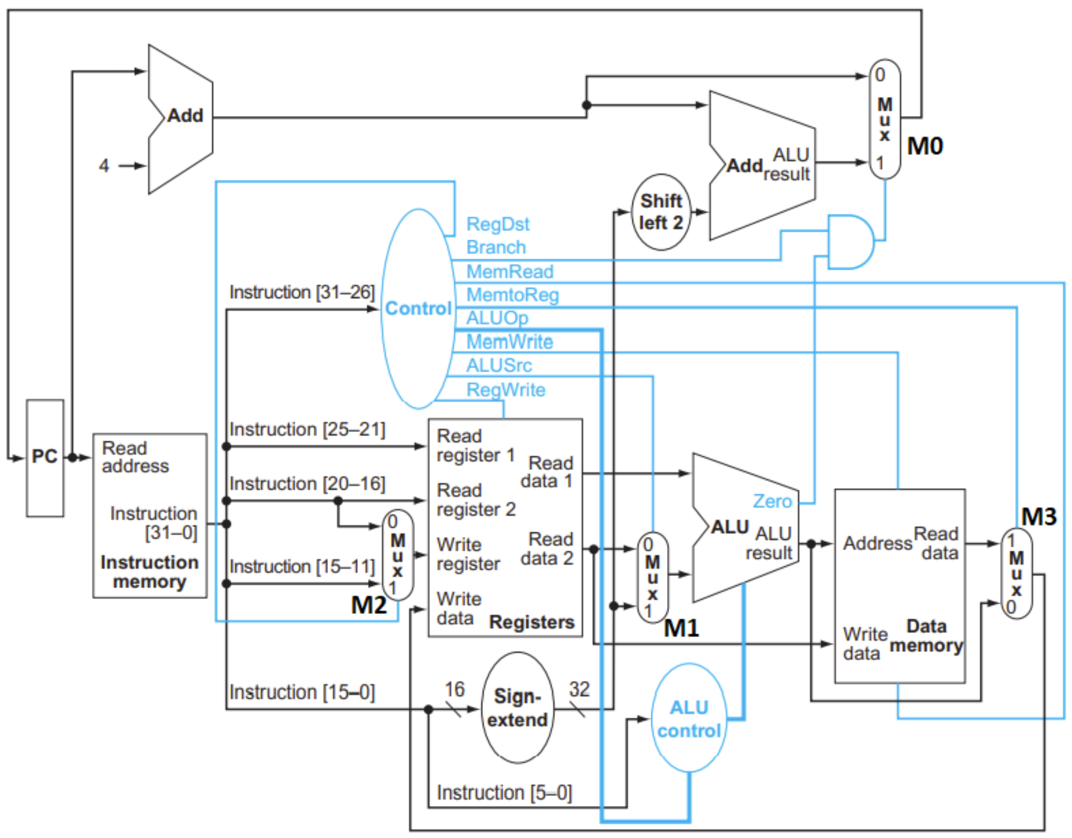
Imagine you are the designer of a single-cycle RISC processor similar to ones discussed in class. However there is a bug in your ALU as shown below:



Design a simple assembly test that will expose this bug (i.e., a sequence of MIPS assembly instructions that will get the wrong answer because of this bug). To receive credit, you must justify why this test exposes the problem with your ALU. You should not assume anything about the initial values of the registers (i.e., you should set the register values to be what you want in the assembly code).

Problem 4 [30 points]

Single Cycle Datapath and Control Path



For this question, we will refer to the above-given single-cycle datapath and control signals. For convenience, muxes are named M0 to M3 so they are easier to reference while answering the questions.

Additionally following are the delays for each unit:

I-Mem	Add	Mux	ALU	Registers	D-Mem	Sign extend	Shift left 2
225 ps	65 ps	20 ps	150 ps	100 ps	250 ps	20 ps	15 ps

Assume delays for all other elements are 0 ps.

Part A [10 points]

1. Suppose the processor only supports 1 instruction which is:

- beq rs, rt, offset: if(rs == rt) branch to offset relative to pc

What are the control signal values for “RegDst, Branch, MemRead, MemtoReg, MemWrite, ALUSrc & RegWrite”? (For each signal specific 1, 0, don’t care)

What will be the minimum cycle time for the processor?

Part B [10 points]

2. Now let's add support for 2 more instructions which are as follows: **[10 pts]**

- add rd, rs, rt: rd ← rs + rt
- lw rt, offset(rs): rt ← memory[rs + signExt(offset)]

What will be the minimum cycle time of the processor? Write the timings for both add and lw to explain your answer.

Part C [10 points]

3. Lets define a new instruction called:

- jmi rs, offset: pc = memory[rs + signExt(offset)]

For this new instruction:

i.) Which of the already existing blocks can be used?

ii.) What are the new blocks needed to support the instruction and where? Also, specify what are the control block changes? If any.

Problem 5 [10 points]

(5 points) We want to build low-power core for a smartwatch type product. Our design achieves low power but reduces the performance of 20% of the benchmark program running on the processor by a factor of 4. On the other hand, architects have come up with an optimization that improves the performance of another 12% of the benchmark by a factor of 6. Assume the two enhancements are independent and affect different parts of the program (there is no overlap between the 20% and 12% of the program). What is the overall effect on the performance of the program?

(5 points) Explain how jump instruction in MIPS works with an example – where PC address for the jump instruction is 0x FFFF 0000 and 26-bit label address is 0000 0000 0000 0000 1111 0000 11.

Name: ______________________________ (Write your name on this page if you use it)

SCRATCH PAGE #1

Name: ____________________________ (Write your name on this page if you use it)

SCRATCH PAGE #2