計算機結構 期中考

Undergraduate Computer Architecture, Midterm Exam

本次考試注重對於課程內容的理解，請在回答題目的時候列出推理和計算的過程有助於獲得部分分數，必要時可自行加入某些假設。

This examination emphasizes on your understanding of the course materials, and you may want to show how you reason and calculate in answering the questions to get partial credits. You may also add assumptions if it is necessary.

如果您同意以下的「榮譽考試」聲明的話，請簽名:

我在這場期中考試的過程中，沒有作弊也沒有接受其他同學的任何幫助。

If you agree with the following sentence, please sign your name below it.

I have not cheated nor have I received any help from other students in the exam.

我的學號和姓名 (My Student ID and Name): B07902048 李宥霆

1. (30 pts) [Instruction Set Architecture and CPU Performance]：Please read the following article (from anantech.com) and answer the questions after the article.

Apple Announces The Apple Silicon M1: Ditching x86 - What to Expect, Based on A14

by Andrei Frumusanu on November 10, 2020 3:00 PM EST

Today, Apple has unveiled their brand-new MacBook line-up. This isn't an ordinary release – if anything, the move that Apple is making today is something that hasn't happened in 15 years: The start of a CPU architecture transition across their whole consumer Mac line-up.

Thanks to the company's vertical integration across hardware and software, this is a monumental change that nobody but Apple can so swiftly usher in. **The last time Apple ventured into such an undertaking in 2006, the company had ditched IBM's PowerPC ISA and processors in favor of Intel x86 designs.** Today, Intel is being ditched in favor of the company's own in-house processors and CPU microarchitectures, built upon the Arm ISA.

The new processor is called the Apple M1, the company's first SoC designed with Macs in mind. **With four large performance cores, four efficiency cores, and an 8-GPU core GPU, it features 16 billion transistors on a 5nm process node.** Apple's is starting a new SoC naming scheme for this

(g) (5 pts) Among these five formats: FP64, FP32, TF32, FP16, and BFLOAT16, which format has the highest precision? Which has the lowest precision? Which has the lowest range?

(h) (5 pts) Why does the author believe that TF32 is a good format for AI applications?

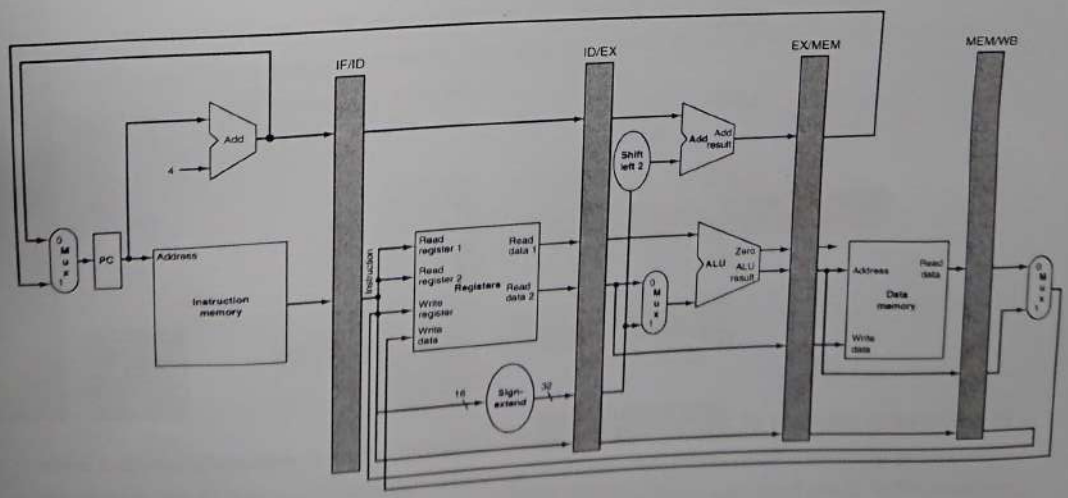
(i) (5 pts) The blog article mentions that applications using NVIDIA libraries enable users to harness the benefits of TF32 with no code change required, because "TF32 Tensor Cores operate on FP32 inputs and produce results in FP32". For the benefits, what do the applications have to sacrifice?

(j) (5 pts) The blog article compares A100 to V100 and shows a 6X speedup training BERT. Do you think that this is a fair comparison between the two processors? Does the two formats produce exactly the same results? How would you defend NVIDIA's claim?

(k) (5 pts) For even better performance, one can go with FP16. What are the advantages and disadvantages of using FP16?

(l) (5 pts) Design a hardware to convert FP32 numbers to TF32 and another hardware to convert TF32 numbers to FP16.

3. (40 pts) [Assembly Code and Pipeline] Let us analyze the performance of an assembly code running on the 5-stage pipelined RISC-V datapath shown below. Note that in this version, branch instructions are handled in the EX stage without any prediction, and there is no data forwarding unit and hazard detection unit.



Consider the following loop (Version A):

```
LOOP:   ld x10, 0(x13)
        ld x11, 8(x13)
        add x12, x10, x11
        subi x13, x13, 16
        bnez x13, LOOP
```

(m)(5 pts) Can the code run correctly without hazard detection? What could go wrong when it is executed by the pipelined datapath?

(n) (5 pts) We could make the code correct by inserting **nop** instructions. Please show a correct code and call it Version B.

(o) (5 pts) Which **nop** instructions in Version B are inserted to resolve data hazards? Which are inserted to resolve control hazards?

(p) (5 pts) Please calculate the number of cycles per iteration approximately, assuming the **x13** register is initialized to a very large number.

(q) (5 pts) It may be possible to optimize the code by re-ordering the instructions. Can you produce an instruction schedule to reduce the number of cycles per iteration?

(r) (5 pts) We can improve the performance by adding a forwarding unit and a hazard detection unit to the pipelined datapath. Suppose we have done that, can the new processor execute Version A correctly? What is the approximate number of cycles per iteration, assuming the **x13** register is initialized to a very large number?

(s) (5 pts) Can the new processor (with forwarding and hazard detection unit) benefits from code optimization? Calculate the approximate number of cycles per iteration and the speedup over Version A.

(t) (5 pts) Branch prediction can reduce the penalty caused by control hazards. In this case, how would you implement branch prediction on top of the new processor to improve the performance?