# Introduction to Computer Architecture

## Assignment 2

### Due November 30, 2020

**01. [10 = 1 x 10]**

Which is the code sequence for C = A + B for load-store ISA?

| A. | Push A | B. | Load A | C. | Load R1, A | D. | Load R1, A |
|----|--------|----|--------|----|-----------|----|-----------|
|    | Push B |    | Add B |    | Add R3, R1, B |    | Load R2, B |
|    | Add    |    | Store C |    | Store R3, C |    | Add R3, R1, R2 |
|    | Pop C  |    |        |    |            |    | Store R3, C |

Which of the following is NOT an operation of ID stage?

A. A <- Regs[rs]
B. B <- Regs[rt]
C. ALUOutput <- A + IMM
D. IMM <- sign-extended immediate field of IR

Which operation does Add R1, @(R3) enforce?

A. Regs[R1] <- Regs[R1] + Regs[R3]
B. Regs[R1] <- Regs[R1] + Mem[Regs[R3]]
C. Regs[R3] <- Regs[R1] + Mem[Regs[R3]]
D. Regs[R1] <- Regs[R1] + Mem[Mem[Regs[R3]]]

In the following codes, there exist _____ data dependences.

```
ADD    R3, R2, R1
SUB    R2, R4, R3
SW     R2, #8(R4)
```

A. RAW, WAR    B. RAW, WAW    C. RAW, WAR, WAW    D. WAR, WAW

Which sequence of instructions cannot use forwarding path to address all potential data hazards?

| A. | Lw R1, 0(R2) | B. | Lw R1, 0(R2) |
|----|--------------|----|--------------|
|    | And R6, R1, R7 |  | Sub R6, R2, R7 |
|    | Or R8, R1, R9 |   | Or R8, R6, R9 |

| C. | Add R1, R2, R3 | D. | Add R1, R2, R3 |
|----|----------------|----|----------------|
|    | Lw R4, 0(R1)   |    | SD R4, 12(R1)  |

If the branch condition and target address are determined until the end of the EXE stage, what are the penalties when 1) the branch is taken and the predicted taken policy is used and 2) the branch is untaken, and the predicted untaken policy is used?

A. 2, 0          B. 2, 2          C. 3, 0          D. 3, 3

Suppose that in 1000 memory references there are 40 misses in the first-level cache

and 20 misses in the second-level cache. Assume the miss penalty from the L2 cache to memory is 200 clock cycles, the hit time of the L2 cache is 10 clock cycles, the hit time of L1 is 1 clock cycle, and there are 1.5 memory references per instruction. What is the average stall cycles per instruction?
A. 6.6          B. 5.4          C. 11          D. 15

In a 5-stage MIPSprocessor with double-bump but noforwarding, how many pipeline stalls must be inserted at running following program?
ADD R1, R2, R3
ADD R4, R0, R0
ADD R5, R0, R0
ADD R3, R1, R2
A. 0B. 1          C. 2          D. 3

Which type of data hazard is called "true dependence"?
A. Read after write.
B. Write after write.
C. Write after read.
D. Read after read.

On handling exceptions, if the pipeline can be stopped so that the instructions issued before the faulting instruction complete and those after it can be restarted, then the pipeline is said to implement _____.
A. unordered exceptions
B. asynchronous exceptions
C. precise exceptions
D. imprecise exceptions

***CHAPTER 1 - FUNDAMENTAL***
02. [2 = 1 x 2]
a. Please describe the two kinds of parallelism in applications.
b. Please describe the four major ways to exploit the preceding two kinds of application parallelism.

03. [2 = 1 x 2]
Some microprocessors today are designed to have adjustable voltage, so a 15% reduction in voltage may result in a 15% reduction in frequency.
What would be the impact on dynamic energy and on dynamic power?

04. [2 = 1 x 2]
Please explain how the following techniques help modern microprocessors improve energy efficiency.
a. dynamic voltage-frequency scaling (DVFS);

b. overclocking.

## 05. [2 = 1 x 2]
Assume a disk subsystem with the following components and MTTF:
- o 10 disks, each rated at 1,000,000-hour MTTF
- o 1 ATA controller, 500,000-hour MTTF
- o 1 power supply, 200,000-hour MTTF
- o 1 fan, 200,000-hour MTTF
- o 1 ATA cable, 1,000,000-hour MTTF

a. Using the simplifying assumptions that the lifetimes are exponentially distributed and that failures are independent, compute the MTTF of the system as a whole.
b. Using the components and MTTFs from above, calculate the reliability after adding one redundant power supply.

## 06. [3 = 1 + 2]
a. Please describe the equation of the Amdahl's law;
b. Use the Amdahl's law to reason about whether processor performance can be indefinitely improved. More specifically, given the faction of professor operations that can be enhanced, reason about why it is this enhancement fraction that decides the ultimate performance improvement.

## 07. [2 = 1 x 2]
Suppose we have made the following measurements:
- o Frequency of FP operations = 25%
- o Average CPI of FP operations = 4.0
- o Average CPI of other instructions = 1.33
- o Frequency of FPSQR = 2%
- o CPI of FPSQR = 20

Assume that the two design alternatives are to decrease the CPI of FPSQR to 2 or to decrease the average CPI of all FP operations to 2.5.
Compare these two design alternatives using the processor performance equation.

## 08. [4 = 2 x 6]
You are designing a system for a real-time application in which specific deadlines must be met. Finishing the computation faster gains nothing. You find that your system can execute the necessary code, in the worst case, twice as fast as necessary.
a. How much energy do you save if you execute at the current speed and turn off the system when the computation is complete?
b. How much energy do you save if you set the voltage and frequency to be half as much?

## 09. [4 = 1 x 4]
Your company has just bought a new processor, and you have been tasked with optimizing your software for this processor. You will run two applications on this dual

core, but the resource requirements are not equal. The first application requires 80% of the resources, and the other only 20% of the resources. Assume that when you parallelize a portion of the program, the speedup for that portion is 2.

a. Given that 40% of the first application is parallelizable, how much speedup would you achieve with that application if run in isolation?

b. Given that 99% of the second application is parallelizable, how much speedup would this application observe if run in isolation?

c. Given that 40% of the first application is parallelizable, how much overall system speedup would you observe if you parallelized it?

d. Given that 99% of the second application is parallelizable, how much overall system speedup would you observe if you parallelized it?

## 10. [5 = 1 x 5]

When parallelizing an application, the ideal speedup is speeding up by the number of processors. This is limited by two things: percentage of the application that can be parallelized and the cost of communication. Amdahl's law takes into account the former but not the latter.

a. What is the speedup with $N$ processors if 80% of the application is parallelizable, ignoring the cost of communication?

b. What is the speedup with 8 processors if, for every processor added, the communication overhead is 0.5% of the original execution time.

c. What is the speedup with 8 processors if, for every time the number of processors is doubled, the communication overhead is increased by 0.5% of the original execution time?

d. What is the speedup with $N$ processors if, for every time the number of processors is doubled, the communication overhead is increased by 0.5% of the original execution time?

e. Write the general equation that solves this question: What is the number of processors with the highest speedup in an application in which $P$% of the original execution time is parallelizable, and, for every time the number of processors is doubled, the communication is increased by 0.5% of the original execution time?

## ***APPENDIX A - INSTRUCTION***
## 11. [9 = 3 x 3]

For each of R, I, J type instructions:

a. draw its encoding format with each field marked;

b. explain the meanings of each field;

c. provide an example instruction and explain its operations.

## 12. [7 = 1 x 7]

Please describe the meaning of the following addressing modes and provide an example instruction for each.

a. Immediate

b. Displacement

c. Register indirect
d. Direct or absolute
e. Memory indirect
f. Scaled
g. PC-relative or position independence

## 13. [6 = 2 x 3]

For the following assume that values A, B, and C reside in memory. Also assume that instruction operation codes are represented in 8 bits, memory addresses are 64 bits, and register addresses are 6 bits.

For each instruction set architecture shown in Figure A.2 (*stack*, *accumulator*, *register-memory*, *register-register*), how many addresses, or names, appear in each instruction for the code to compute $C = A + B$, and what is the total code size?

## 14. [6 = 2 x 3]

Consider the following fragment of C code:

    for (i = 0; i <= 100; i++)
    { A[i] = B[i] + C; }

Assume that A and B are arrays of 64-bit integers, and C and i are 64-bit integers. Assume that all data values and their addresses are kept in memory (at addresses 1000, 3000, 5000, and 7000 for A, B, C, and i, respectively) except when they are operated on. Assume that values in registers re lost between iterations of the loop.

a. Write the code for MIPS.
b. How many memory-data references will be executed?
c. What is the code size in bytes?

## 15. [2 = 2 x 1]

Our favorite program runs in 10 seconds on compute A, which has a 4 GHz clock. We are trying to help a computer designer build a computer, B, that will run this program in 6 seconds. The designer has determined that a substantial increase in the clock rate is possible, but this increase will affect the rest of the CPU design, causing computer B to require 1.2 times as many clock cycles as computer A for this program.

What clock rate should we tell the designer to target?

## ***APPENDIX C - PIPELINE***

## 16. [2 = 1 x 2]

What are two major benefits of the multi-cycle datapath over the single-cycle datapath?

## 17. [16 = 2 x 1]

Please list the types of exceptions that might arise in various stages of a five-stage pipeline?

## 18. [2 = 1 x 2]
a. Please describe the two-memory technique that eases pipeline design.
b. Use an example to justify the benefit of it for improving pipeline performance.

## 19. [2]
Analyze why ideal pipelining with equal-length pipe stages yields the highest speedup.

## 20. [5 = 1 x 5]
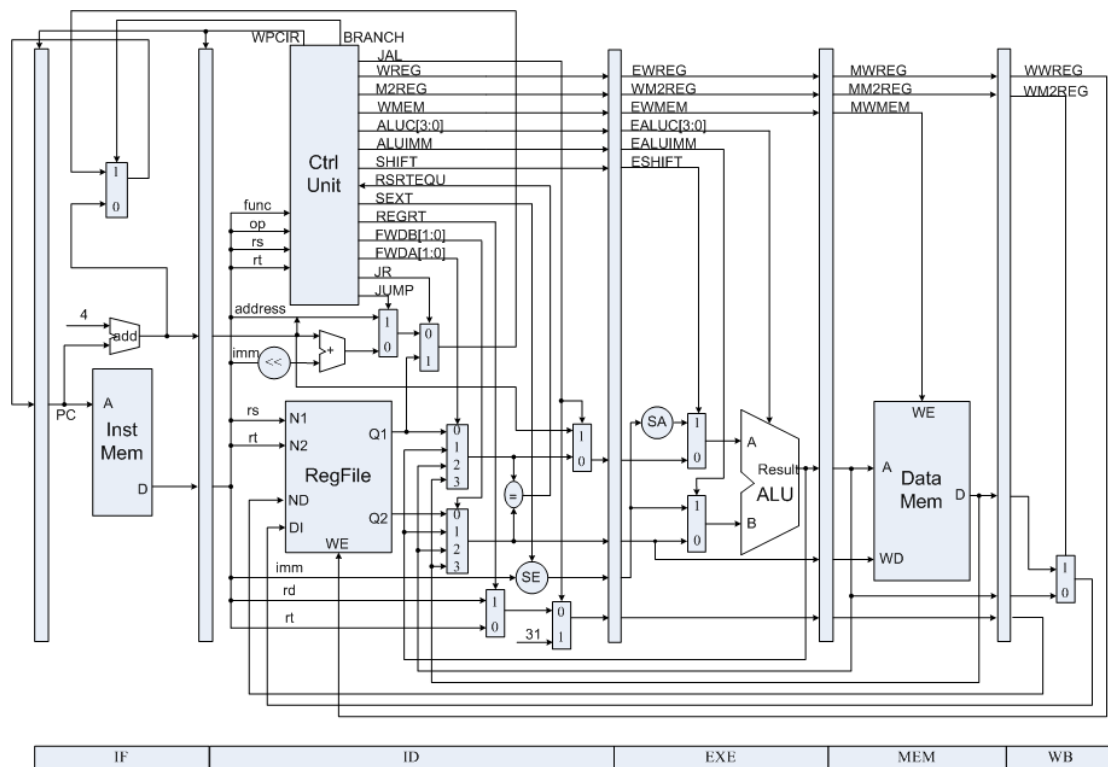Based on the following pipelined datapath, draw the links taken by each instruction:
ADD
Load
Store
BNE
J



## 21. [2 = 1 x 2]
Reason about the following two penalties:
a. PredictedUntaken-PenaltyUntaken0;
b. PredictedTaken-PenaltyTaken2.

| Branch scheme | Penalty unconditional | Penalty untaken | Penalty taken |
|---|---|---|---|
| Flush pipeline | 2 | 3 | 3 |
| Predicted taken | 2 | 3 | ② |
| Predicted untaken | 2 | ⓪ | 3 |

## 22. [6 = 2 x 3]

Use the following code fragment:

```
Loop:     LD          R1, 0(R2)        ; load R1 from address 0+R2
          DADDI       R1, R1, #1       ; R1 = R1 + 1
          SD          R1, 0(R2)        ; store R1 at address 0+R2
          DADDI       R2, R2, #4       ; R2 = R2 + 4
          DSUB        R4, R3, R2       ; R4 = R3 - R2
          BNEZ        R4, Loop         ; branch to Loop if R4 != 0
```

Assume that the initial value of R3 is R2 + 396.

a. Data hazards are caused by data dependences in the code. List all of the data dependences in the code above. Record the register, source instruction, and destination instruction; for example, there is a data dependency for register R1 from the LD to DADDI, that is,

R1      LD        DADDI

b. Show the timing of this instruction sequence for the 5-stage RISC pipeline without any forwarding or bypassing hardware but assuming that a register read and a write in the same clock cycle "forwards" through the register file, as shown in Figure C.6. Use a pipeline timing chart like that shown in Figure C.5. Assume that the branch is handled by flushing the pipeline. If all memory references take 1 cycle, how many clock cycles does this loop take to execute?

c. Show the timing of this instruction sequence for the 5-stage RISC pipeline with full forwarding and bypassing hardware. Use a pipeline timing chart like that shown in Figure C.5. Assume that the branch is handled by predicting it as not taken. If all memory references take 1 cycle, how many clock cycles does this loop take to execute?

## 23. [4 = 1 x 4]

We begin with a computer implemented in single-cycle implementation. When the stages are split by functionality, the stages do not require exactly the same amount of time. The original machine had a clock cycle time of 7 ns. After the stages were split, the measured times were IF, 1 ns; ID, 1.5 ns; EX, 1 ns; MEM, 2 ns; and WB, 1.5 ns. The pipeline register delay is 0.1 ns.
a. What is the clock cycle time of the 5-stage pipelined machine?
b. If there is a stall every 4 instructions, what is the CPI of the new machine?
c. What is the speedup of the pipelined machine over the single-cycle machine?
d. If the pipelined machine had ana infinite number of stages, what would its speedup be over the single-cycle machine?

## 24. [4 = 2 x 2]

In this question, we will explore how deepening the pipeline affects performance in two ways: faster clock cycle and increased stalls due to data and control hazards.

Assume that the original machine is a 5-stage pipeline with a 1 ns clock cycle. The second machine is a 12-stage pipeline with a 0.6 ns clock cycle. The 5-stage pipeline experiences a stall due to a data hazard every 5 instructions, whereas the 12-stage pipeline experiences 3 stalls every 8 instructions. In addition, branches constitute 20% of the instructions, and the misprediction rate for both machines is 5%.

a. What is the speedup of the 12-stage pipeline over the 5-stage pipeline, taking into account only data hazards?

b. If the branch mispredict penalty for the first machine is 2 cycles but the second machine is 5 cycles, what are the CPIs of each, taking into account the stalls due to branch mispredictions?

## 25. [5] Devote to Enjoy

[requirements: a. append an extra blank page after your feedback for possible comments; b. submit an extra copy of Q25 via qq/dingtalk for ease of (anonymous) compilation; photocopy of a handwritten version is also welcome.]

I hope you really enjoyed the class (at least for some moments) so far. You decided to study Computer Architecture with me among several instructors. I truly appreciate your trust and your continuing support, understanding, tolerance, and cooperation. Ever since we met, I keep working on how to provide you with a satisfactory learning experience in return.

For almost all previous editions of this course, an assignment question sought to solicit thoughts and suggestions from the participants. For example, do you gradually understand strategies or things from different perspectives and weigh their tradeoffs? What do you think is the real challenge for you to learn this course? Do you consider interactions in class helpful? What held you back when you were trying to ask or answer questions in class? What suggestions (for better learning this course) would you like to provide to other students? What help or assistance would like from me or other students? All their thoughtful and constructive feedback has encouraged us toward a more rewarding computer architecture class.

This time, I would like you to introspect beyond this course. Taking a course should be rewarding in many possible ways. As we discussed in the first lecture session, even if you may barely practice computer architecture principles after the class ends, if whatever you learn through this class---whether it be computer architecture per se, a philosophy it implies, a study tip or inspirational quote we share, or a new friend you know---keeps driving you toward your greater self, this course fulfills its mission.

In particular, you are highly expected to develop a clear vision and be determined to strive for it. Most students in the computer architecture class are junior. The third year cannot be more decisive for your future. If you aim to work in an IT company after graduation, start practicing skills that help you ace the interview. If you plan to pursue a higher degree, join a research lab to cultivate research experience and application

materials. If you have not decided, do both if your time and energy permit, see which one you really prefer. Try to make your decision based on your own goal and experience. Work with people who keep your interest and passion alive. It is really, really important to identify a right role model to emulate and learn from. It is also highly decisive to focus on what really is essential, especially given that some of constantly emerging paradigms might turn into hypes and fade away. Stick to golden rules that should be objective during the selection. It should not be simply about what one claims to be or what one claims others less so. "See the world not as it is, but as it should be." Think of the proverb "shallow brooks babble loudest, still waters run deep" once in a while. (Or as sg put it: "shallow water hualahuala, deep water kckc.") In return, be part of the initiative that motivates yourself and people around you.

Meanwhile, if you ever consider assistance from me as possibly helpful, whether course related or beyond, never hesitate to reach out. As I proudly claim on my webpage, I am always proud to be part of the journey for someone to excel. And I am pretty sure that I would also love to be part of yours as well.

???Therefore, with this question, I would like you to share your thoughts on your goal and plan. For example, what is your goal after graduation? What is your plan to achieve that goal and what challenges might be involved? How do you manage to be motivated and determined? What helpful advices or suggestions did you get from senior students and professors? What suggestions would you like to offer peers with similar goals? Or, it is possible that you are still finding your goals. Don't rush and take your time. Being at such a young age, you have infinite possibilities to live your dreams.

"Where are you? Here.
  What time is it? Now.
  What are you? This moment."

"Old urges continue to arise, but urges do not matter; only actions do."
"A warrior is as a warrior does."

"While they are deciding, make even more art."

So pleeeeease, live a life you deeply enjoy and will remember.

If not now, when?