

# Introduction to Computer Architecture

## Assignment 2

Due November 25, 2019

### 1. [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. $A \leftarrow \text{Regs}[\text{rs}]$        | B. $B \leftarrow \text{Regs}[\text{rt}]$                              |
| C. $\text{ALUOutput} \leftarrow A + \text{IMM}$ | D. $\text{IMM} \leftarrow \text{sign-extended immediate field of IR}$ |

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

- A.  $\text{Regs}[\text{R1}] \leftarrow \text{Regs}[\text{R1}] + \text{Regs}[\text{R3}]$
- B.  $\text{Regs}[\text{R1}] \leftarrow \text{Regs}[\text{R1}] + \text{Mem}[\text{Regs}[\text{R3}]]$
- C.  $\text{Regs}[\text{R3}] \leftarrow \text{Regs}[\text{R1}] + \text{Mem}[\text{Regs}[\text{R3}]]$
- D.  $\text{Regs}[\text{R1}] \leftarrow \text{Regs}[\text{R1}] + \text{Mem}[\text{Mem}[\text{Regs}[\text{R3}]]]$

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 MIPS processor with double-bump but no forwarding, 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. 0    B. 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

## 2. [6 = 2 x 3]

For each of R, I, J type instructions:

- draw its encoding format with each field marked;
- explain the meanings of each field.

## 3. [4]

Analyze why ideal pipelining with equal-length pipe stages yields the highest speedup.

## 4. [10 = 2 x 5]

Based on the following pipeline datapath, draw the links taken by each instruction:

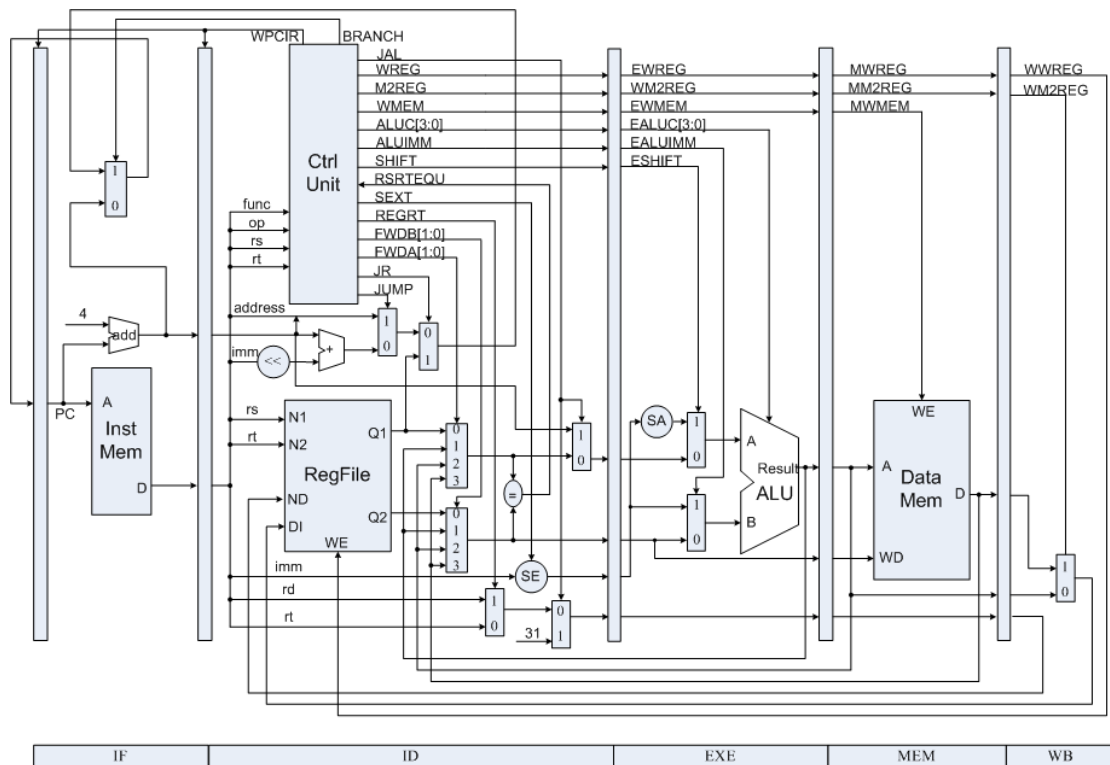
ADD

Load

Store

BNE

J



### 5. [10 = 5 x 2]

Reason about the following two penalties:

- PredictedUntaken-PenaltyUntaken0;
- 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

### 6. [10 = 5 x 2]

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.

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
```

- 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?

**7. [4 = 2 x 2]**

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.

- How much energy do you save if you execute at the current speed and turn off the system when the computation is complete?
- How much energy do you save if you set the voltage and frequency to be half as much?

**8. [16 = 4 x 4]**

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?

**9. [10 = 5 x 2]**

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 memory access time and
- average stall cycles per instruction?

Ignore the impact of writes.

**10. [10 = 5 x 2]**

Assume a fully associative write-back cache with many cache entries that starts empty. Below is a sequence of five memory operations (the address is in square brackets):

```
Write Mem[100];
Write Mem[100];
Read  Mem[200];
Write Mem[200];
Write Mem[100];
```

What are the number of hits and misses (and which are them) when using no-write allocate versus write allocate?

## 11. [10] Wherever You Go, There You Are

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 support. 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 weight 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 important to decide 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. In return, be part of the initiative that motivates yourself and people around you.

“See the world not as it is, but as it should be.”

??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?

“Where are you? Here.

What time is it? Now.

What are you? This moment.”