Name (print): _____ UCSC Spring 2018
CRUZID: _____ May 14th, 2018

CMPE110 Midterm Exam

Exam Instructions: Answer each of the questions included in the exam. Write all of your answers directly on the examination paper, including any work that you wish to be considered for partial credit. The examination is closed book, but you can make use of one page of notes (both sides) and a calculator. You may not use a computer or cell phone of any kind.

On equations: Wherever possible, make sure to first write the equation with symbolic terms, then the equation rewritten with the numerical values, and then the final solution. Partial credit will be weighted appropriately for each component of the problem, and providing more information improves the likelihood that partial credit can be awarded.

On writing code: Unless otherwise stated, for any answers that require code examples or fragments, you should write C-like pseudocode. You do not need to optimize your code unless specifically instructed to do so. Comments for any code are not strictly required on the exam, but are highly recommended. They may help you receive partial credit on a problem, if they help us determine what you were trying to do.

On time: You will have **one hour (60 minutes)** to complete this exam. Budget your time and try to leave some at the end to go over your work. The point weightings correspond roughly the difficulty of each problem. If you find a problem too difficult at first, move on to the other problems and revisit it later.

UCSC CODE OF STUDENT CONDUCT

The Code of Student Conduct is an undertaking of the students, individually and collectively that they will not cheat. This includes, but is not limited to:
a.) Providing answers to or receiving answers from others for any academic assignment.
b.) Using notes, information, calculators, or other electronic devices or programs during exams or for assignments from which they have been expressly or implicitly prohibited;
c.) Improperly obtaining or using improperly obtained information about an exam or assignment in advance of its availability to other students, or assisting others in doing so;
d.) Putting one's name on another person's exam or assignment; or e. Altering previously graded work for purposes of seeking a grade appeal.
I acknowledge and accept UCSC's Code of Student Conduct:

Name (sign) _____

		Score	Grader
Problem 1	20	_____	_____
Problem 2	9	_____	_____
Problem 3	21	_____	_____
	Total (50)	_____	

Space for Notes & Comments

Problem 1:

Multiple Choice. Answer the questions below by selecting exactly one of the possible answers by circling one of the letters a), b), c), or d). If you need to revise your selection, draw an X across your selection (strike-through) and circle another option. Solutions with 2 marked answers or where markers are unclear receive zero points. **(2 Points for each question)**

1.) The RISC-V ISA we discussed uses 32 general purpose registers. Assume we want to increase the number of registers to 256. To enable this, the machine instruction format of three operand instructions would require:

a) The same number of bits
b) 9 more bits
c) 6 more bits
d) 64 bits

2.) **srl x1, x1, 4** performs a division of the value in x1 by:

a) 4
b) 8
c) 16
d) 24

3.) In RISC-V, instructions can have

a) Two input register operands
b) No register operands
c) One input and one output register operands
d) All three above

4.) Assume a multi-core chip with the MSI coherence protocol for the private caches, where each cache line can be in one of the following states: M = modified, S = shared, I=invalid. If a line is in M state in one core's cache, there may exist:

a) Unlimited copies of that line in S state in other caches
b) At most one copy of that line in S state in another cache
c) No other copies of that line in M or S state
d) At most one copy of that line in M state in another cache

5.) An in-order, scalar, pipelined processor in which all instructions require the same number of pipeline stages is subject to:

a) WAR Hazards
b) WAW Hazards
c) RAW Hazards
d) All three above

6.) Consider a 32-bit processor with a 8-way set associative cache, with 512 total entries and a block size of 16 Byte. Which address bits contain the tag?

a) 31:10
b) 31:11
c) 31:12
d) 31:13

7.) Which of the following is a valid control flow changing instruction:

a) jr x1
b) jalr x1, label
c) beq x1, x2, label
d) All three above

8.) Assume a processor where all instructions have a CPI of 1 except branches which introduce a 1 cycle bubble (control point is the instruction decode stage). Assume a branch strategy of **Predict Not Taken**, a branch frequency of 20% (% of instructions) and that 50% of all branches are taken. What is the branch stall CPI?

a) 0.1
b) 0.2
c) 0.25
d) 0.5

9.) Assume the following C pseudo code snippet. What is the branch misprediction percentage of the branch instruction implementing the for-loop, assuming a hardware branch predictor that uses a single history bit to determine whether a branch was taken/not taken the last time? Assume the foo() is called many times.

```
void foo() {
        for (int i = 0; i < 8; i++) { bar() }
}
```

a) 0%
b) 12.5%
c) 25%
d) 50%

10.) Assume the following instruction sequence executed on an out-of-order processor that supports register renaming (32 additional physical registers). What is the maximum number of instructions that can be executed in parallel during a single clock cycle?

```
Add x1, x1, x2
Sub x1, x3, x4
Mul x3, x1, x5
Div x5, x3, x4
```

a) 1
b) 2
c) 3
d) 4

Problem 2:

a) Average CPI: Compute the average CPI for three different applications A, B and C that utilize 2 different instructions 1 and 2. **(6 Points)**

Application	IC (ins 1)	CPI (ins 1)	IC (ins 2)	CPI (ins 2)	Avg CPI
A	5	3	4	7	
B	7	2	4	1	
C	3	2	6	2	

b) Execution Time:
Assume that you execute the three applications from a) on a 2.5 GHz processor. Compute the execution time for all three applications. **(3 Points)**

Application	Execution time
A	
B	
C	

Problem 3:

Consider the 6-pipeline stage, in-order processor architecture shown in Figure 1 which resembles the 5-pipeline stage processor that we covered in class with the exception that data accesses from memory are split into two pipeline stages. Note that instructions can still be fetched in a single clock cycle. With this, the processor's stages are as follows: Instruction fetch (IF), instruction decode (ID), execution (EX), memory 1 (MEM 1), memory 2 (MEM 2), and write back (WB). All instructions take 6 cycles to complete.

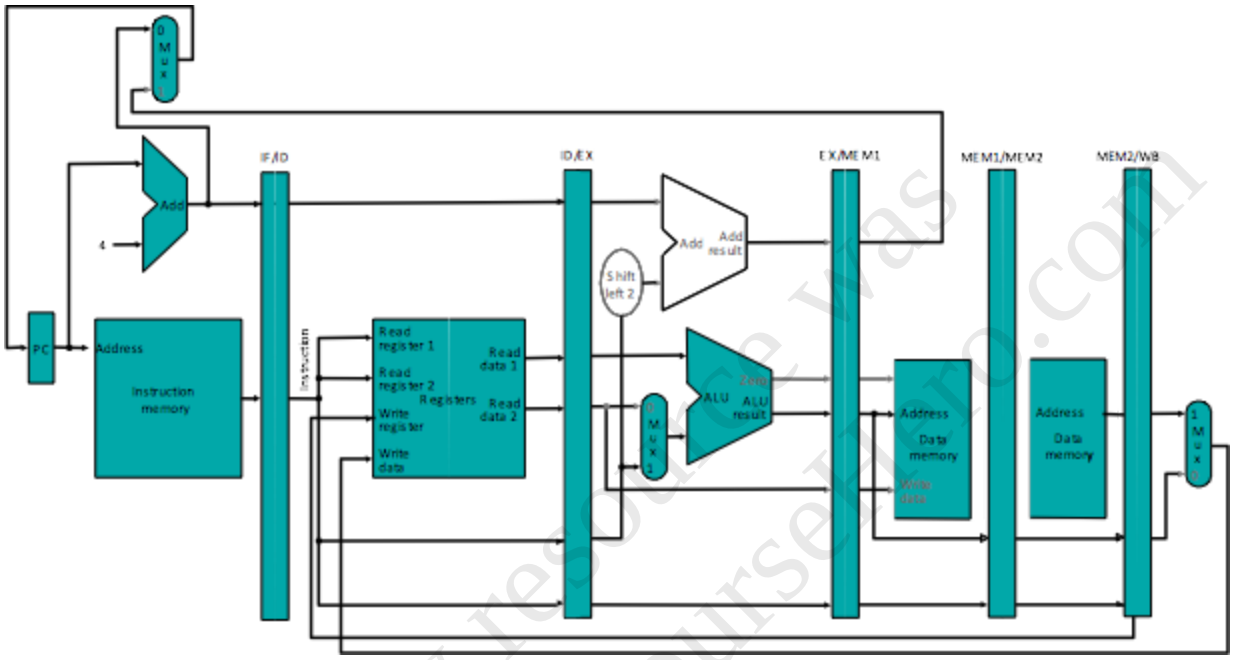

Figure 1: 6-pipeline stage processor

a) Data Dependencies. To ensure correct operation in the presence of RAW dependencies, bubbles/NOPs need to be inserted into the instruction sequence. The amount of NOPs can be reduced by introducing forwarding paths. To analyze the feasibility of adding forwarding paths, determine the stages that are producers, sources and consumers of forwarding data in the architecture above: **(6 Points)**

Producers of forwarding data:

Sources of forwarding data:

Consumers of forwarding data:

b) Assuming the addition of forwarding paths to the architecture in Figure 1), how many bubbles/NOPs need to be inserted to avoid RAW hazards between 2 dependent arithmetic instructions (EX to EX)? **(2 Points)**

c) Assuming the addition of forwarding paths to the architecture in Figure 1), how many bubbles/NOPs need to be inserted to avoid RAW hazards between a load and an arithmetic instruction? (MEM to EX)? **(2 Points)**

d) Determine all RAW, WAR and WAW **dependencies** (not necessarily hazards) in the instruction sequence below. There may be multiple dependencies per line. **(5 Points)**

Sequence	Instruction	Dependency (from->to)	Type (WAR, WAW, RAW)
1.	Ld x4, 0(x2)		
2.	Add x3, x2, 8		
3.	Ld x5, 8(x4)		
4.	Sub x7, x3, x4		
5.	Mul x8, x7, x5		
6.	Ld x8, 0(x5)		
7.	Div x7, x8, x1		

e) Consider the instruction sequence from 3 e) and assume a processor pipeline with added forwarding paths. Rewrite the sequence adding NOPs into the instruction stream to avoid all hazards that can occur in an in-order pipeline. **(6 Points)**

Instruction	Space for comments