CSE 490/590 – Quiz 1 – February 12, 2025 – V1

Instructions:

1. This Quiz is **closed book and closed notes.**
2. You may have pens, pencils, erasers, a calculator, and a water bottle/drink.
3. Electronic devices are **NOT ALLOWED** during this Quiz, with the exception of a calculator. **You shall NOT share a calculator with others during the Quiz.**
4. Place your student ID card on your desk for us to review.
5. When you are finished, remain in your seat and raise your hand and we will come and collect your **Quiz**. **You must not talk to anyone in the room** until your Quiz has been collected, and you have left the room.
6. **Any form of cheating/academic integrity violation, including (but not limited to) violation of the rules above will result in an automatic 0 for the Quiz.**

Please fill your name in the blank and sign the statement below:

I, ____,

have read, and acknowledge that I will adhere to the instructions above and if not followed, I will accept the penalty given by the instructor.

Signature: ______________________________________

STOP!! PLEASE <u>DO NOT</u> START THE QUIZ

UNTIL YOU ARE TOLD TO DO SO

FOR GRADERS ONLY:

Q1: ___________6 points

Q2: ___________6 points

Q3: ___________6 points

Q4: ___________10 points

Q5: ___________12 points

Q6: ___________10 points

Total: ____________50 points

[Question 1] (6 Points)

Suppose that we want to enhance the processor used for web serving. The new processor is 4 times faster on computation in a given web serving application than the original processor. Assuming that the original processor was busy with computation 40% of the time and was waiting for I/O 60% of the time, what is the overall speedup gained by incorporating the enhancement? **Show your work.**

$$\text{Speedup}_{\text{overall}} = \frac{\text{Execution time}_{\text{old}}}{\text{Execution time}_{\text{new}}} = \frac{1}{(1 - \text{Fraction}_{\text{enhanced}}) + \dfrac{\text{Fraction}_{\text{enhanced}}}{\text{Speedup}_{\text{enhanced}}}}$$

$$= \frac{1}{(1-0.4) + \dfrac{0.4}{4}} = \frac{1}{0.6+0.1} = \frac{1}{0.7} = 1.43$$

[Question 2] (6 Points)

A program's run time for a given processor is determined using the number of instructions per program, the number of cycles per instruction, and clock frequency of the processer. Consider a program with the following instruction mix. Find the CPI for the same. **Show your work.**

Instructions	%	cycle
branch	15%	4
integer ALU	40%	1
load	25%	3
sw	20%	2

$CPI_{BRANCH} = 0.15 \times 4 = 0.6$ $\quad CPI_{INT} = 0.4 \times 1 = 0.4$

$CPI_{LOAD} = 0.25 \times 3 = 0.75$ $\quad CPI_{STORE} = 0.2 \times 2 = 0.4$

$\therefore CPI_{PROG} = 0.6 + 0.4 + 0.75 + 0.4 = 2.25$

[Question 3] (6 Points) Provide the binary representation for the following instruction: [Use Green Sheet attached] **(Show your work)**

lw $s0,4($t0)

$$\underbrace{100011}_{OPCODE}\ \underbrace{01000}_{RS}\ \underbrace{10000}_{RT}\ \underbrace{0000000000000100}_{IMM}$$

[Question 4] [10 Points]

a. Consider a byte-addressable memory system with the following contents: **(3 points)**

Memory Location	**Value**
0x1246	0x56
0x1247	0x12
0x1248	0x93
0x1249	0x53
0x124a	0x21
0x124b	0x19
0x124c	0x67
0x124d	0x83

0x93532119

If the following instruction is executed:

```
lw $t0, 4($t1)
```

$t1 contains the address 0x1244. What will $t0 contain? Use Big-Endian.

Updated b. Assume that $s1 contains the value 0x34343434 and $s0 contains the address 0x12345A20. Assume that the memory data, starting from address 0x34343434 is: 0x77886604. What will be the value of $s0 after the following code is executed: **Show your work. (7 points)**

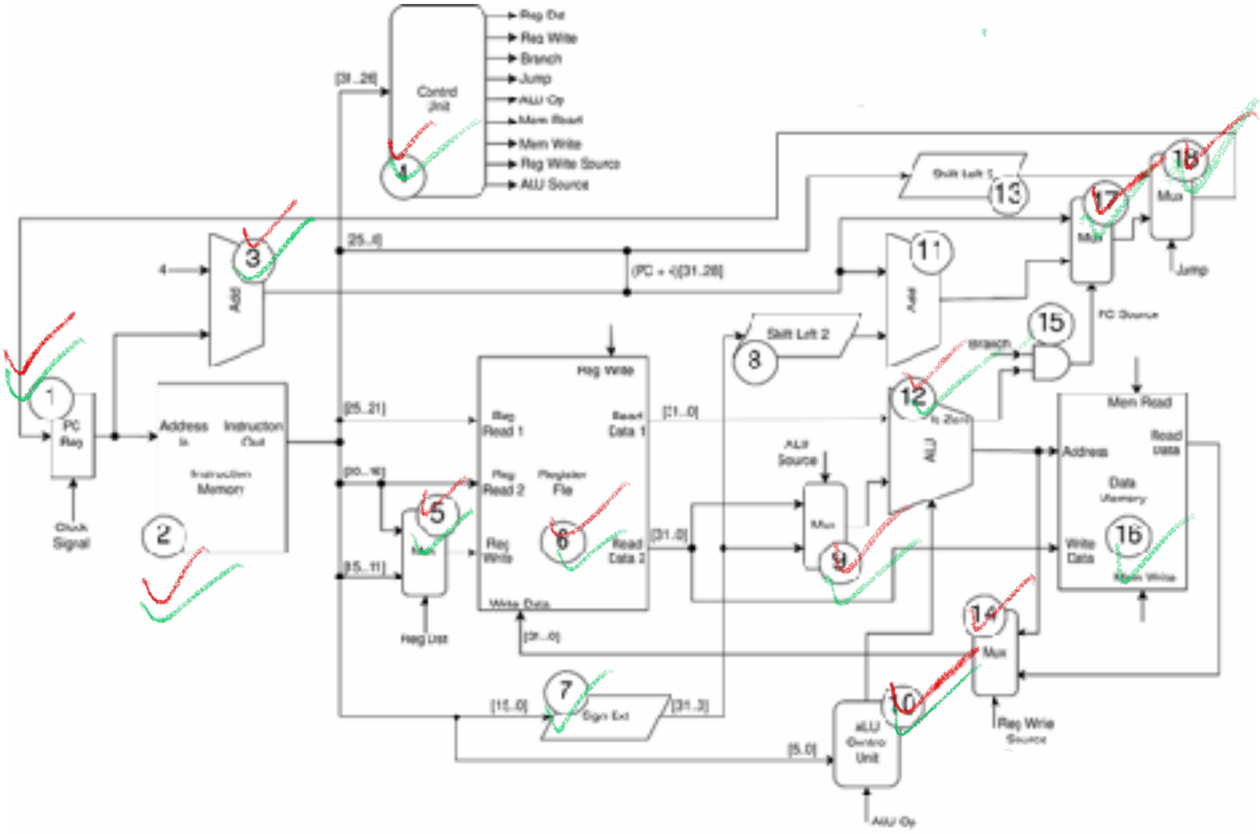
```
(i)     lb $s0, 1($s1)
```

0xFFFFFF88 since 0x88 gets sign-extended.

```
(ii)     lh $s0, 0($s1)
```

0x00007788

Updated **[Question 5] (12 Points)** Given the following single cycle MIPS datapath:



a. For each of the following instructions, list the stages that are necessary for the execution of the given instruction: (IF, ID, EX, MEM, WB)

i. `add $t6, $s3, $s2`

IF, ID, EX, WB

ii. `sw $s3, 24($s1)`

IF, ID, EX, MEM

b. For each of the following instructions, list the component numbers (as shown in the diagram above) that are required for the given instruction:

i. `add $t6, $s3, $s2`

in red

REQUIRED: 1, 2, 5, 6, 9, 12, 14

OPTIONAL: 3, 4, 10, 17, 18

WRONG: 7, 8, 11, 13, 15, 16

REQUIRED: 1, 2, 6, 7, 9, 12, 16
OPTIONAL: 3, 4, 10, 17, 18
WRONG: 5, 8, 11, 13, 14, 15

ii. `sw $s3, 24($s1)`

in green

Updated **[Question 6] (10 Points)**

Consider the following MIPS instruction sequence:

```
add $s3, $s2, $s1
sub $s4, $s2, $s3
lw $t0, 8($s1)
addi $t1, $t0, 2
sw $t0, 4($s1)
```

a. Find the data (RAW) hazards for the above instruction sequence and explain why it creates such hazard(s). **(3 Points)**

Line 2 reads from $s3 while line 1 hasn't updated the value stored in the register yet, causing a RAW hazard.

Line 4 and 5 read from $t0 while line 3 hasn't fetched the data from memory yet, causing a RAW hazard.

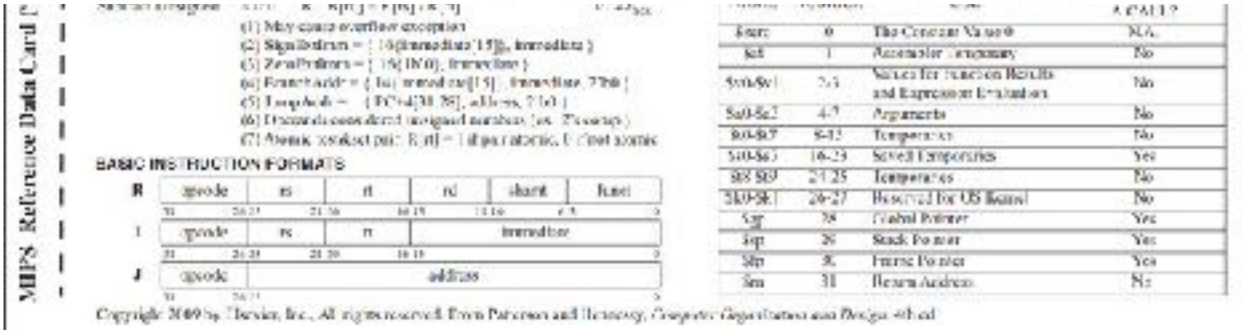
b. Show the pipeline diagram after inserting Data Forwarding Unit to overcome data dependencies: **(5 Points)**

Instr.	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Add $s3, $s2, $s1	IF	ID	EX	MEM	WB									
Sub $s4, $s2, $s3		IF	ID	EX	MEM	WB								
Lw $t0, 8($s1)			IF	ID	EX	MEM	WB							
-	-	-	-	-	-	-	-	-	-	-				
Addi $t1, $t0, 2					IF	ID	EX	MEM	WB					
Sw $t0, 4($s1)						IF	ID	EX	MEM	WB				

(c) If you did not use forwarding to handle data hazards as in (b), how many more clock cycles it will take to complete the instruction sequence above. **(2 points)**

3 more clock cycles

Instr.	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Add $s3, $s2, $s1	IF	ID	EX	MEM	WB									
-	-	-	-	-	-	-	-	-	-	-				
-	-	-	-	-	-	-	-	-	-	-				
Sub $s4, $s2, $s3				IF	ID	EX	MEM	WB						
Lw $t0, 8($s1)					IF	ID	EX	MEM	WB					
-	-	-	-	-	-	-	-	-	-	-				
-	-	-	-	-	-	-	-	-	-	-				
Addi $t1, $t0, 2								IF	ID	EX	MEM	WB		
Sw $t0, 4($s1)									IF	ID	EX	MEM	WB	



MIPS Reference Data Card

Subtract Unsigned subu R R[rd] = R[rs] - R[rt] 0 / 23hex

(1) May cause overflow exception
(2) SignExtImm = { 16{immediate[15]}, immediate }
(3) ZeroExtImm = { 16{1b'0}, immediate }
(4) BranchAddr = { 14{immediate[15]}, immediate, 2'b0 }
(5) JumpAddr = { PC+4[31:28], address, 2'b0 }
(6) Operands considered unsigned numbers (vs. 2's comp.)
(7) Atomic test&set pair; R[rt] = 1 if pair atomic, 0 if not atomic

BASIC INSTRUCTION FORMATS

R	opcode	rs	rt	rd	shamt	funct
	31 26	25 21	20 16	15 11	10 6	5 0
I	opcode	rs	rt	immediate		
	31 26	25 21	20 16	15		0
J	opcode	address				
	31 26	25				0

NAME	NUMBER	USE	PRESERVED ACROSS A CALL?
$zero	0	The Constant Value 0	N.A.
$at	1	Assembler Temporary	No
$v0-$v1	2-3	Values for Function Results and Expression Evaluation	No
$a0-$a3	4-7	Arguments	No
$t0-$t7	8-15	Temporaries	No
$s0-$s7	16-23	Saved Temporaries	Yes
$t8-$t9	24-25	Temporaries	No
$k0-$k1	26-27	Reserved for OS Kernel	No
$gp	28	Global Pointer	Yes
$sp	29	Stack Pointer	Yes
$fp	30	Frame Pointer	Yes
$ra	31	Return Address	No