USAF ACADEMY

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

ECE 281 GRADED REVIEW #3
SPRING 2016 – VER A

Name: ______________________________ **Section:** __________

Academic Security	This examination is not released from academic security until **1630** on **13 May 2016**. Until this time, you may not discuss the examination contents or the course material with anyone other than your instructor.
Integrity	Your honor is extremely important. This academic security policy is designed to help you succeed in meeting academic requirements while practicing the honorable behavior our country rightfully demands of its military. Do not compromise your integrity by violating academic security or by taking unfair advantage of your classmates.
Authorized Resources	▪ Calculator ▪ MIPS documentation
Instructions	▪ **Show all work for full credit** ▪ Box or circle your final answer. ▪ For all numerical answers, use engineering notation and include units. ▪ Completely label all your diagrams, drawings, graphs, etc. for full credit. ▪ You have **53 minutes** to complete this exam.

Problem	Value	Earned
1	38	
2	16	
3	16	
4	30	
Total	100	

This page intentionally left blank.

Problem 1 (38 points) *General Knowledge*

a. (3 points) Name one difference between a branch and a jump instruction in MIPS.

b. (2 points each) Indicate if the following statements are true or false.

T F Microarchitecture is the specific arrangement of registers, memories, ALUs and other building blocks to form a microprocessor.

T F A computer architecture defines the underlying hardware implementation.

T F An exception can be caused by either hardware or software.

T F MIPS is an example of a CISC architecture.

c. (2 points) Which type of memory does MIPS use? Circle the correct answer.

Byte-addressable ***Word-addressable***

d. (8 points) Match the following descriptions with the associated memory segment.

____ 1. Location of the stack A. Reserved

____ 2. Location of machine language program B. Dynamic data

____ 3. Cannot be used directly by the program C. Global data

____ 4. Contains variables that can be seen by all functions D. Text

e. (3 points) What is the purpose of the linker in creating an executable file?

f. Your friend Tom stored the number 0x12345678 in memory in a computer (shown below). Indicate the correct answer to the following question:

(2 points) Is this memory ***big-endian*** or ***little-endian***?

(2 points) Explain why your answer is correct.

Address	Data
0x00400000	0x12
0x00400001	0x34
0x00400002	0x56
0x00400003	0x78

g. (4 points) One of the design principles articulated by the class text says that "Simplicity favors regularity". Give **one** example of how this principle appears in MIPS. Be specific and explain how this example exemplifies the design principle.

h. (6 points) Suppose the single-cycle MIPS processor suffered a fault on RegDst control line, resulting in that line always having a value of 1. Circle ***all*** of the following MIPS instructions that would no longer work properly.

i. `add` ii. `lw` iii. `sw` iv. `bne` v. `ori` vi. `j`

Problem 2 (16 points) *MIPS machine code*

a. (8 points) Convert the following MIPS assembly instruction to machine code. Write your answer in hexadecimal.

```
   addi $t0, $0, -10
```

b. (8 points) Convert the following machine code to its corresponding MIPS assembly instruction:

```
0x0089502A
```

Problem 3 (16 points) *MIPS Assembly*

No pseudoinstructions are allowed to be used in answers to the next two questions.

a. (12 points) Write a MIPS assembly program to implement the below `if` loop. Be sure to use appropriate comments. For full points, implement the solution using the `slt` command and ensure your program does not "fall off the end" of memory.

```
if (A >= B)
    A = A + 1
else
    B = B - 1
#A is in $t0
#B is in $t1
```

b. (4 points) MIPS does not contain a `nori` command. Write some MIPS assembly code that will enable the following logical operation to occur (full points only for the fewest possible lines):
`$s0 = $s1 NOR 0xDFEC`

Problem 4 (30 points) *MIPS Microarchitecture*

Modify the MIPS single-cycle processor by adding the lui (load upper immediate) instruction. You will need to add hardware and a new control signal. **Hint: fully correct solutions will adhere to the following operation for lui: [rt] = [rs] + [imm, 16'b0].**

a. (10 points) **Hardware**: Modify the schematic on the next page. Clearly and neatly indicate where you are breaking the current schematic to insert your added functionality, and clearly indicate what goes on inside the component(s) you insert. Briefly describe below the specific changes you are making to the schematic.

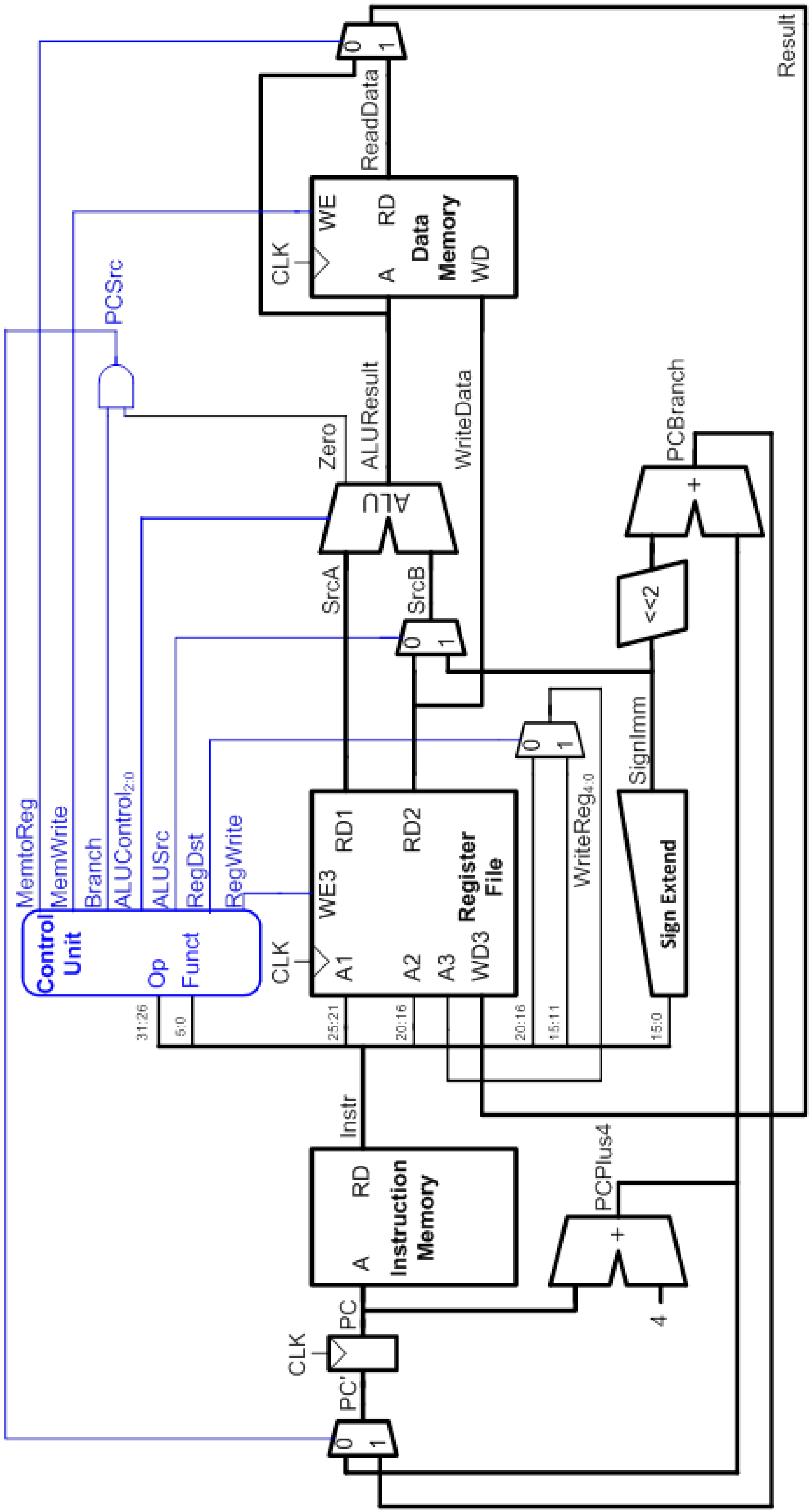
b. (20 points) **Control Table:** Fill in the missing elements of the main decoder table below. Don't forget to account for the new control signal too.

Instr	**$Op_{5:0}$**	**RegWrite**	**RegDst**	**AluSrc**	**Branch**	**MemWrite**	**MemtoReg**	**$ALUOp_{1:0}$**	**Jump**	
R-type	000000	1		0	0	0	0	10	0	
lw	100011	1	0		0		1	00	0	
sw	101011	0	X		0	1	X	00	0	
addi	001000			1	0	0	0	00	0	
j	000010	0	X	X	X	0	X	XX	1	
lui	001111									

ALU Decoder:

$ALUOp_{1:0}$	**Meaning**
00	Add
01	Subtract
10	Look at funct
11	Not Used

This page intentionally left blank.


CLK
PC'
PC
0
1
A
RD
Instruction Memory
4
+
PCPlus4
Instr
Control Unit
Op
Funct
31:26
5:0
MemtoReg
MemWrite
Branch
ALUControl$_{2:0}$
ALUSrc
RegDst
RegWrite
PCSrc
25:21
20:16
20:16
15:11
15:0
CLK
WE3
A1
A2
A3
WD3
RD1
RD2
Register File
WriteReg$_{4:0}$
Sign Extend
SignImm
<<2
+
PCBranch
SrcA
SrcB
ALU
Zero
ALUResult
WriteData
CLK
WE
A
RD
WD
Data Memory
ReadData
Result