The University of Alabama in Huntsville
Electrical and Computer Engineering Department
CPE 221 01
Spring 2016
Test 1
February 17, 2016

This test is closed book, closed notes. You may not use a calculator. You should have the reference packet that includes Figure 2.10 and Appendix B. You must show your work to receive full credit.

Name: _____

1. (1 point) The instruction _____ specifies the size and meaning of fields within the instruction.
2. (1 point) _____ instructions move data from a memory location or register to another memory location without changing its form.
3. (1 point) _____ instructions can alter the normal flow of control from executing the next instruction in sequence.
4. (1 point) A 0-address instruction uses a _____ to hold both operands and the result.
5. (1 point) _____ is an example of an addressing mode found in processors.

6. (10 points) Represent 208 and -132 as signed (2s complement) 16-bit numbers

7. (25 points) Consider the following SRC program. Trace the values of the registers shown as they change during program execution. Also, trace the writes to memory by the `st` instruction. There may be unused columns or rows in the tables. If you need to add columns or rows, you may do so. `.dc 1` reserves one word of storage and sets it equal to 1. `.dw 3` reserves 3 words but does not give those words a value.

```
        .org 200
size:   .dc 6
a:      .dc 5
x:      .dc 5, 3, -1, 2, 4, 37
y:      .dw 10
orig:   .org 1000
         la     r29, loop
         la     r10, x
         la     r11, y
         ld     r1, size
         ld     r2, a
loop:    ld     r3, 0(r10)
         add    r3, r3, r2
         st     r3, 0(r11)
         addi   r10, r10, 4
         addi   r11, r11, 4
         addi   r1, r1, -1
         brnz   r29, r1
         stop
```

r1													
r2													
r3													
r10													
r11													
r29													

Results of the `st` instruction.

Memory Address	Contents

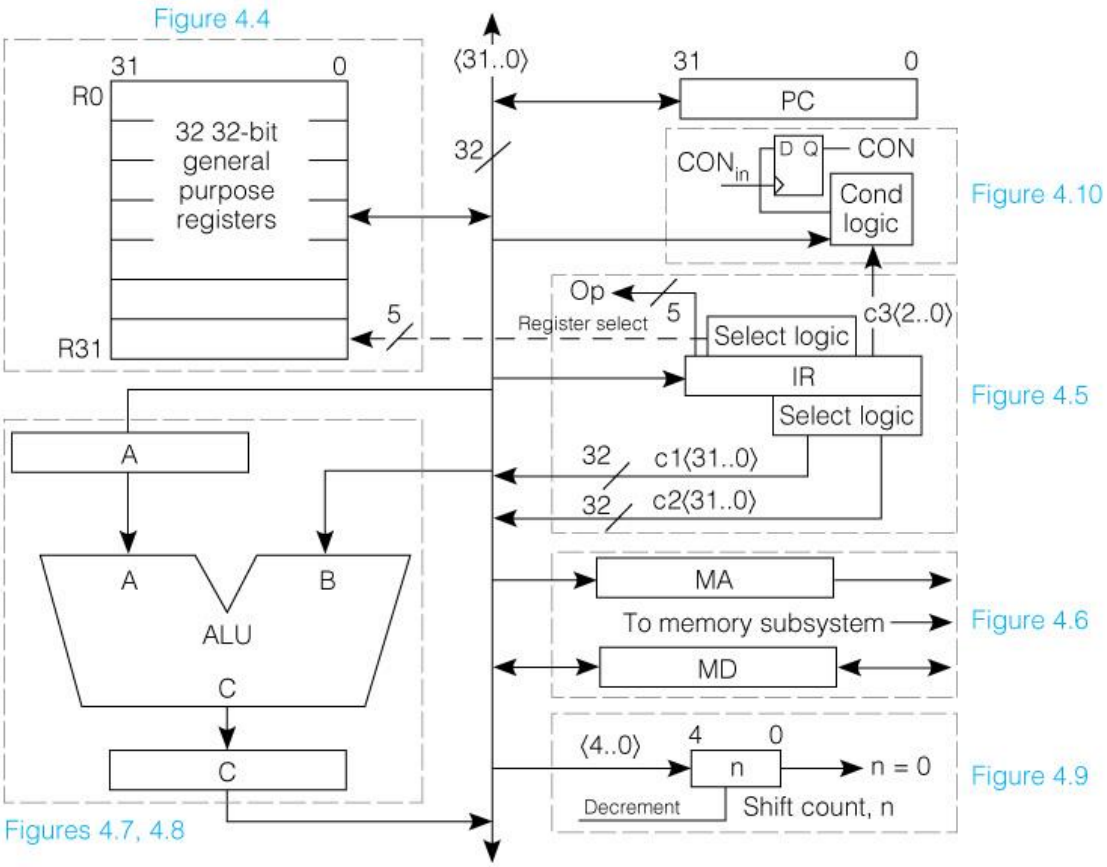
8. (15 points) Translate the selected statements from the SRC program shown below into machine code (a series of ones and zeros), then express the result in hexadecimal.

```
        .org 200
count:  .dw 1
bits:   .dw 1
num:    .dc 0
mask:   .dc 1
orig:   .org 1000
        lar   r30, done
        lar   r29, shift
        lar   r28, next
        ld    r1, num
        ld    r3, mask
        sub   r2, r2, r2
        sub   r5, r5, r5
        brzr  r30, r1
next:   and   r4, r1, r3
        addi  r5, r5, 1
        brzr  r29, r4
        addi  r2, r2, 1
shift:  shr   r1, r1, 1
        brnz  r28, r1
done:   st    r2, count
        st    r5, bits
        stop
```

Address	Instruction	op	ra	rb	rc	c1	c2	c3	Hexadecimal
	and r4, r1, r3								
	shr r1, r1, 1								
	brzr r29, r4								

9. (25 points) Write the code to implement the expression A = ((B/F) + (C $\times$ D)) $\times$ E on 3-, 2-, 1-, and 0-address machines. Do not rearrange the expression. In accordance with programming language practice, computing the expression should not change the values of its operands. When using a 0-address machine, the order used is SOS op TOS, where SOS is second on stack and TOS is top of stack.

10. (20 points) Extend the SRC instruction set by adding the instruction `ldrr ra, rb, rc`, that is described by the abstract RTN R[ra] ← M[R[rb] + R[rc]]. (a) Write concrete RTN steps for this new instruction using the 1-bus SRC microarchitecture shown. (b) Which format would be a good one to use for `ldrr`?



T0	MA ← PC : C ← PC + 4
T1	MD ← M[MA] : PC ← C
T2	
T3	
T4	
T5	
T6	
T7	