Computer Architecture II

Name: _____

SSN: _____

Signature: _____

Email: _____@_____

Your test should have six (6) questions on seven (7) pages (including this title page). The point value for each question is given below. If you have a question, please raise your hand and someone will be there as soon as possible. You have two hours to complete this test. Use your time wisely. Good Luck!!!

SCORE:

Question	Possible	Points
1	15 pts	_____
2	20 pts	_____
3	15 pts	_____
4	20 pts	_____
5	15 pts	_____
6	15 pts	_____
TOTAL:	100 pts	_____

Use the rest of this page to inform the instructor of any special circumstances that you feel should be considered when determining your letter grade. **Note: The instructor may disagree with you.**

1. In Mythsim, the Negation operation (also called Change Sign) on an integer value can be done in two ways. We could (1) subtract the original value from zero or we could (2) invert all the bits of the original value and add one. The second way is possible since Mythsim uses the twos complement format to encode integer values. In our program, the Negation operation had the original value specified by Rj. It may turn out that by specifying the original value in Rk would result in a more efficient implementation of the Negation operation. For each of the two implementations of the Negation operation (subtract-from-zero and invert-and-increment), specify whether it is more efficient to specify the original value in Rj or Rk. Explain your answers. Note: in both cases the most efficient implementation requires exactly two statements in Mythsim.

2. Assume that the following data section is written in a MIPS/SPIM program that has an array "arr" which can hold up to 200 integers, an integer "size" which holds the number of values stored in the array and an integer "largest":

```
arr:        .word 0,200
size:       .word 0
terget:     .word 0
count:      .word 0
```

Write the MIPS/SPIM code that will find the number of times that the value in "target" appears in the array and place that value into "count". You are to assume that the values are already stored in the array and that the size contains the current number of values stored in the array.

3. Write a function that will take two integer parameters and return one integer value. The function is to return the value of "1" if the two parameters are the same or the value of "0" if the two parameters are not the same. You are also to write the code that will call the function that will use the following data section. The integers "val1" and "val2" contain the values to be sent as parameters. The return value from the function is to be stored in the integer "val3". Your program is to follow the MIPS register usage convention. The parameters may be passed to the function on the stack or in the appropriate registers.

```
val1:   .word 0
val2:   .word 0
val3:   .word 0
```

4. There is a function the will follow the MIPS register usage convention. It will have three parameters stored on the stack. It will call two other functions. It will use the registers: $s0, $s1, $t0, $t1 and $t2. It will have an array as a local variable. The size of the array is given in the second parameter stored on the stack. Drawing a picture of the stack may be helpful, but is not required.

4a. Write code for the function that will allocate space and save any needed information on the stack. This code would be at the start of the function.

4b. Write code for the function that will initialize the local variable array to have values of zero.

4c. Write code for the function that will restore any needed information from the stack and deallocate ALL space from the stack used by this function (including the space for the parameters). The code would be at the end of the function followed by the jr instruction.

5a. Describe the IEEE Floating Point Standard (FPS) for single precision floating point numbers.

5b. Show the binary encoded value in the IEEE FPS for single precision floating point number for the decimal value of -23.65625 (which is the binary value of -10111.10101). Be sure to clearly specify the value in each bit of the encoded value.

6. For each of the following addressing modes specify what information is stored in an instruction using that addressing mode and how the actual operand is accessed.

6a. Immediate Mode

6b. Register Mode

6c. Direct Mode

6d. Register Direct

6e. Base Displacement