CMPUT 229 Computer Organization and Architecture I
2000-2001 (Winter)
Final Examination

(April 23, 2001)

Name: ______________ SID: ______________

Do all problems. Closed book and notes. No calculator. Instruction set sheet is provided.

Problem 1

Fill in the blanks. All answers must be in the specified base.

1. Using 8 bits to encode unsigned numbers, the largest we can have is ______ (in decimal).
2. Using 8 bits to encode signed numbers with 2's complement representation, the most negative and the most positive we can have are ________ and ________ (in decimal), respectively.
3. The decimal equivalent of `0xf0` is ______, if we interpret it as a number in 8-bit 2's complement representation.
4. Knowing that $(100)_9 - (100)_8 = (17)_{10}$, $(100)_n - (100)_{n-1} =$ ____________. (Your answer will be in terms of n.)
5. The address space of 32-bit MIPS architecture is 4 ____________ (kilobytes, megabytes, or gigabytes).

Problem 2

Fill in the blanks. All answers must be in the specified base.

1. Decimal -4.5 in fractional binary is equal to ____________.
2. $(111.010)_2$ in scientific notation is ____________.
3. In IEEE single precision (SP) representation, the exponent field of $(1.0011) \times 2^1$ is encoded as ________________ (in decimal), and its mantissa field ________________ (in binary).
4. In IEEE SP representation, (10.011) is expressed as ________________ (in hexadecimal).
5. $(10.011)_2 = (2.375)_{10}$. If the very next, slightly more positive number than 2.375, that can be encoded exactly is $2.375 + 2^x$, then $x =$ ______.

Problem 3

1. Implement the pseudo-instruction `li $t0, 0x12345678` in terms of two real instructions.

2. Suppose `arr[n]` is an array of integers, if `a[0]` is stored at `0x10010040` in the memory, what is the address of `a[15]`?

3. Encode the `beq` assembly instruction in the following loop:

```
loop:    nop
         beq  $8, $9, loop
```

4. `0x0043001a` and `2128fff9` represent two MIPS machine instructions. Decode what they are in symbolic form.

5. If `$t0` contains `0x00001234` initially, what will it hold after the execution of the instruction `sll $t0, $t0, 3`.

6. Instruction `addi` may or may not be a pseudo-instruction, depending upon what operands are used with it. Give an example that makes "`addi`" a pseudo-instruction.

7. The following three ways of "jump"ing all seem to serve the same purpose.

```
1. beqz   $zero, there    2. j there    3.  la    $t0, there
                                            jr    $t0
```

   (a) Which one of the three is the most flexible in terms of where label "there" can be?

   (b) Give one reason why one wouldn't always use this "most flexible" case.

Problem 4

```
        .kdata  0x90000000
__m1_:  .asciiz "  Exception "
__m2_:  .asciiz " occurred and ignored\n"
__e0_:  .asciiz "  [Interrupt] "
__e1_:  .asciiz ""
__excp: .word __e0_,__e1_
```

Given the above assembler directives that initialize the kernel data segment, answer the following questions

1. What is the address of the label __m1_?
2. How many bytes does string in __m1_ take to store?
3. What is the address of the label __m2_?
4. What are the two words in array __excp initialized to?

Problem 5

Write a subroutine, `toupper(char c)` in MIPS assembler that accepts as input in `$a0` a character, and converts it to upper-case, if possible. Specifically,

1. if `c` is lower case, then return the equivalent upper case in `$v0`.
2. Otherwise, return `c` in `$v0`.

Problem 6

MIPS registers are 32-bits wide. Therefore, it is possible to store an IEEE SP floating-point (FP) number in a MIPS register. Write a subroutine `ge1()` whose C equivalent would be

```
int ge1(float x) { if(x >= 1.0) return(1); else return(0);}
```

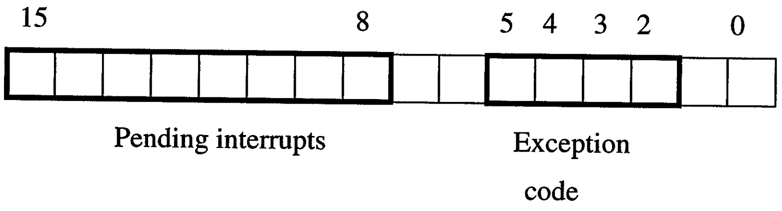
i.e., `ge1()` accepts as input a number in `$a0`, interprets it as a float-point number, and returns a 1 in `$v0`, if the magnitude of the input number is $\geq$ `1.0`; otherwise, it returns a 0 in `$v0`. Note that you are allowed only to use integer instructions to analyze the input FP number, and that instructions such as `bge` and `sub` operate on integers, and are not able to handle FP numbers directly. (Hint: For the magnitude of any FP number to be ≥ 1.0, its real exponent must be ≥ 0, when expressed in scientific notation.)

Problem 7

The format of the Cause register is given in the figure below. Write an exception handler that will

1. if it is internal exception, do nothing and return to the exception causing program or
2. if it is interrupt (external exception), print the message "Interrupt Level #" has occurred" where "#" is the interrupt level number.

Use SPIM's `syscall print_int (1)` and `print_string (4)` to output the string and level number. Recall that in both syscalls, the address of the string or the number to be printed is passed through `$a0`.



Problem 8

Shown below is an entry in a direct-mapped cache with a block size of 4 words. Note that the cache index has 3 bits, and the tag field is 5-bits long.

Index	V	Tag	Data Field
...	.	.....	....
011	1	11011	mem block[?]
...	.	.....	....

Fill in the blanks.

1. The data field of each cache entry contains a total of ______ bytes.

2. The cache cotains a total of ______ blocks. and is capable of holding ______ bytes of data.

3. The memory block address of the block shown above is ________________.

4. The memory contains a total of ______ blocks. and is capable of holding ______ bytes of data.

5. Among the addresses of the bytes in the memory block shown above, the smallest byte address is ________________ (in hexadecimal), and the largest ________________ (in hexadecimal).

6. Without changing the data capacity of the above cache and the block size, if we were to use a two-way set associative cache organization, we would have a total of ________________ sets.

7. Without changing the data capacity of the above cache, if we were to reduce the block size from four to one word and, we would have a total of ________________ blocks.

8. If we were to employ a fully-associative cache organization and maintain the same block size, the tag field of each entry would have ________________ bits.

9. If we were to employ a fully-associative cache organization and use the block size of one word, the tag field of each entry would have ________________ bits.