CS220 Exam 1 Spring 2010

The only outside materials you may use on this exam is the MIPS green card and a calculator. You must show work or explain your answer to receive credit. Some correct answers are better than other correct answers and will receive full credit.

1. [5 points] What is the decimal value of the unsigned hexadecimal number **A7**?

   167

2. [5 points] Write the decimal number -108 as an 8-bit two's complement binary number.

   That is -128 + 20 or 10010100

3. [5 points] Why does the MIPS multiply instruction use two registers, **HI** and **LO**, for the result?

   When you multiply two 32-bit numbers the result requires more than 32 bits, in fact it yields a 64 bit result hence using one register for the upper 32 bits (HI) and another for the lower 32 bits (LO).

4. [5 points] What decimal number does the 16-bit two's complement number 1111111111111001 represent?

   -7 remember the trick is to ignore the leading 1s except for the last one. That is this number is that same as 1001 4-bit two's complement number.

5. [5 points] Assume signed integers are 8 bits and two's complement. What is the range of decimal values that can be represented?

   -128 to 127

6. [5 points] How many bits are required to represent a register number is a MIPS instruction?

   There are 32 registers so we need 5 bits. The $log_2(32) = 5$

7. [20 points] Write a MIPS assembly language program that reads an integer from the user, counts the number of times that integer can be divided by 2 before

reaching zero and prints the count to the console. Your program should use prompts for input and output.

```
# this counts the number of time an integer can be divided by 2
# before it hits zero
    .data
prompt: .asciiz "enter an integer: "
    .align 2
output: .asciiz "the answer is "

    .text
main:
      la $a0, prompt      # print input prompt
      li $v0, 4
      syscall

      li $v0, 5           # read the integer
      syscall

      move $t0, $v0       # move number to $t0

      li $t1, 0           # counter in $t1

loopbegin:
      slti $t2, $t0, 2    # if number < 2 then we're done
      bne $t2, $zero, loopend
      addi $t1, $t1, 1    # update counter
      srl $t0, $t0, 1     # divide by 2
      j loopbegin         # go back an do it again

loopend:

      la $a0, output      # print the output prompt
      li $v0, 4
      syscall

      li $v0, 1           # print the answer
      move $a0, $t1
      syscall

      jr $ra              # bye bye
```

8. [20 points] Consider the definition of **list** below that contains eight numbers. Write a MIPS program that writes the list in reverse order to the console (10, 0, -15, etc.) Your program should be general and work if I modify the list of numbers.

```
# this counts the number of time an integer can be divided by 2
# before it hits zero
    .data
list:     .word 43, 22, 77, 96, 20, -15, 0, 10

    .text
main:
      la $t0, list              # get address of list
      li $t1, 8                 # 8 items in list
      addi $t0, $t0, 28         # jump to end of list
      li $v0, 1                 # code for printing an integer

loop:
      beq $t1, $zero, endloop # if counter is zero we're done
      lw $a0, 0($t0)            # put the number into a0 for printing
      syscall
      addi $t0, $t0, -4         # address to next item in list
      addi $t1, $t1, -1         # update counter
      j loop
endloop:

      jr $ra                    # bye bye
```

9. [10 points] Suppose processor $\mathbf{P_1}$ has a cycle time of 1 nanosecond and processor $\mathbf{P_2}$ has a cycle time of 0.5 nanoseconds. Furthermore $\mathbf{P_1}$ has a CPI of 2.5 for a program and $\mathbf{P_2}$ has a CPI of 5 for the same program. Which machine is faster for the program and by what factor?

Time = IC(CPI)(CT)

Time1 = IC(2.5)(1) = 2.5IC
Time2 = IC(5)(.5) = 2.5IC

The two processors have the same performance for program P.

10. [10 points] Assume that the execution time of a program **p** on a processor is 200 seconds. Also assume that **p** spends 20% of the execution time doing floating-point multiplications. What would the execution time be if we quadrupled (four times) the performance of floating-point multiplication? What is the overall performance improvement of **p** running on the modified processor?

The program spends 200(.20) or 40 seconds executing floating-point multiplications.

So Time = 200 = 160 + 40. If we speed up the multiplies by a factor of 4 then we would only spend 10 seconds multiplying. So the time would be 170 seconds.

Speedup = 200/170 = 1.18 times facster.

11. [5 points] If the clock cycle time for a processor is 2×10^{-9} seconds what is the clock rate?

Rate = 1 / cycle time = $1/(2 \times 10^{-9})$ = 500MHz