Class	CS47, Sec 01
Homework	I
Due Date	Oct 19, 2016 11:59 PM PST

Instructions

1. There are 6 questions with total 10 points.
2. Please create electronic document with your answer.
3. There is no need to include the question itself. However, you **MUST** include question number and sub-part index if any. Example: 5(b)
4. Please create a PDF document **hw1.pdf** and ***upload that in Canvas*** assignment page by the due date.
5. **Please re-check you submission for any logistic errors (empty file, corrupted PDF, and many more) and re-submit if needed. Once grading is started, any file with logistics errors will be given 0 point.**
6. **NO** handwritten document is accepted.
7. **NO LATE SUBMISSION.**
8. **Please explain your answer clearly – just writing the final answer in a word or two is not sufficient in most of the cases.**

1. A 20-bit system (i.e. address and data are both 20-bit) uses 16 registers (0-15). It supports 3 types of instructions. Type I has machine code format [opcode (4bit) | rs (4bit) | rt (4bit) | rd (4bit) | funct (4bit)] and assembly code format '<mnemonic> <rd>, <rs>, <rt>'. Type II has machine code format [opcode (4bit) | rs (4bit) | rt (4bit) | immediate (8bit)] and assembly code format '<mnemonic> <rt>, <rs>, <immediate>'. Type III has machine code format [opcode (4bit) | address (16-bit)] and assembly code format of '<mnemonic> <address>'. Using this information compute the machine code in octal format for the following instructions assuming anything after '#' is comment and 0xnnnn is number in hexadecimal format. Note that the number beside 'r' in registers definition – this is the numeric representation of register to be used to encode the instruction. [**3 pts**]
   a) add r12, r05, r10 # opcode: 0x2 / funct: 0x2
   b) sub r04, r05, r15 # opcode: 0x2 / funct: 0x3
   c) addi r10, r12, 0x3A #opcode: 0x4
   d) ori r13, r03, 0x1B #opcode: 0x3
   e) jmp 0x23C # opcode: 0x5
   f) jal 0x100F # opcode: 0x6

2. A number system muNote uses symbol Do, Re, Mi, Fa, So, La, Ti with equivalent decimal weight 0, 1, 2, 3, 4, 5, 6 respectively. In that case, answer the following. [**1pts**]
   a) What is the decimal equivalent of ReDoLaTiLaSo?
   b) What is muNote equivalent of decimal number 987654321?

3. A program has a procedure 'my_proc' which takes 3 arguments passed in registers $a0-$a2. This procedure uses registers $t0, $t1, $s0-$s5. Assume the target system uses 32-bit data and address. [**1 pts**]

   a) What is the size of the stack frame size of 'my_proc'?
   b) Write the caller RTE storing code.

4. Write down the program that assembler generates intermediately after the macro expansion preprocessing step is done. **[pts 1]**

```
.macro pop_and_add($argS, $arg1, $arg2)
addi  $sp, $sp, -4
lw    $arg1, 0($sp)
addi  $sp, $sp, -4
lw    $arg2, 0($sp)
add   $arg3, $arg1, $arg2
.end_macro

.text
main:
      pop_and_add($s2, $s1,$s0)
      add   $s0, $s1, $s2
      pop_and_add($s3, $s4,$s5)
      add   $s5, $s4, $s3
      sub   $s6, $s4, $s0
```

5. In a byte addressable system byte sequences are following from address 0x100A000F towards higher memory address - 0x2F, 0x13, 0x2A, 0xFA, 0x2A, 0x61, 0x93, 0x65, 0x6C, 0x67, 0xA5, 0x6A, 0xCD, 0xDE, 0xEF, 0xF1. If the system uses 64-bit register and supports a load load command 'ld64bit <rt>, <address>' to load 64-bit information from memory. What would be the content of register t0 after 'ld64bit $t0, 0x100A0015' in following scenarios? [**1pts**]
   a) System uses little endian convention.
   b) System uses big endian convention.

6. Review the following code and answer the following questions, assuming line number is the address of the symbol (label).
   a) Determine the symbol table content with effective address assuming program startup address is 0x00010000 and data startup address is 0x00100000 **[pts. 2]**
   b) List instruction line number with forward references. **[pts. 1]**

```
1   .text
2   .globl main
3   main:    lw       $t0, V1
4            lw       $t1, V2
5            bne      $t0, $t1, main_L1
6            add      $s0, $t0, $t1
7            sw       $s0, V1
8            j        main_L2
9   main_L1: sub      $s1, $t0, $t1
10           sw       $s1, V2
11  main_L2: bne      $s0, $s1, main
12
13  .data
14  .align 2
15  V1: .asciiz "Hello World!"
16  V2: .word 115
```

Symbol	Calculation	Address