ICS 233, Term 072

Computer Architecture & Assembly Language

HW# 2

Q.1. Carry out resulting from addition of unsigned numbers can be used to check if the result of addition is incorrect. Write the shortest sequence of MIPS instructions to determine if there is a carry out from the addition of two registers $t1 and $t2. Place the carry out (0 or 1) in register $t0.

Q.2. Write a MIPS assembly program that asks the user to enter an integer, reads the integer and then displays the integer representation in both binary and hexadecimal, assuming 32-bit representation. A sample execution of the program is given below:

Enter an integer: -5
Number representation in binary is: 11111111111111111111111111111011
Number representation in hexadecimal is: FFFFFFFB

Q.3. Write a program to implement the procedure, **SelectionSort**, to sort an array of integers (i.e. 32-bit signed numbers) in an **ascending** order.

The pseudocode for the **SelectionSort** procedure is given below:

```
SelectionSort (Array, Size)
        for (position= 0 to Size-2)
                MinValue = Array[position]
                MinPosition = position
                for (j=position+1  to Size-1)
                        if (Array[j] < MinValue) then
                                MinValue = Array[j]
                                MinPosition = j
                        end if
                end for
                if (position ≠ MinPosition) then
                        Array[MinPosition] = Array[Position]
                        Array[Position] = MinValue
                end if
        end for
end SelectionSort
```

Store the array to be sorted in variable Array as defined below.

```
Array: .word 10, 2, 0, 15, 25, 30, 7, 22
```

Your program should display the following:
Array before sorting is: 10 2 0 15 25 30 7 22
Array after sorting is: 0 2 7 10 15 22 25 30

Clearly indicate in your assembly code where each pseudocode statement is translated. Also clearly indicate what registers are used to store the variables. Your program should be very well documented.

The solution should be well organized and your programs should be well documented. Submit a soft copy of your solution in a zip file. The soft copy should include a Readme file indicating the file names of the programs containing the solution and whether they work or not. The Readme file should also contain your name and ID.