**Mini-project report**

Name: Nguyen Duc Dung

Subject: Assembly Language and Computer Architecture Lab

Student ID: 20160674

Class: ICT-02.K61

Teacher: Pham Ngoc Hung

**Problem 5:** ***Write a program to get decimal numbers, display those numbers in binary and hexadecimal***

**Overview:**

In this problem, I need to handle both integer and floating point number, so I consider integer is a floating point number. Therefore, I use InputDialogDouble to get the input from user. Since this dialog has limited range feature, I only deal with the situation that user enters wrong input type and empty input.

After getting correct input from user, I move it from $f0 to register $t8 and $t9.

In order to represent the floating point number precisely, I print it in IEEE 754 format for both binary and hexadecimal form. First, to print decimal input in binary form, I use a mask with only one bit 1 to get each bit of input and save it in a string. By shifting bit 1 from right to left of the mask, I can get all the bits of input. Next, to print input in hexadecimal form, I also use a mask that is 0x0000000f. In the same way, I can get each 4-bit from input. However, I don’t shift four bits 1 of the mask in this case. I shift each 4-bit of input to left most side instead. The result after using mask is going to be change to ASCII character. After that, I save it in a string and print the string when the processing input action finishes.

**Detail explanation:**

* Data:

Bin\_result: string in which the binary form of input will be saved

Hex\_result: string in which the hexadecimal form of input will be saved

* Coprocessor 1:

$f0: saving input from user

* Register:

$v0: use to call system service

$a0, $a1: use to save input for system service

$s1: save parameter for procedures

$s5: count the number of character in string result

$s6: get the address of string result

$t0: save value of $s1 in procedure

$t1: result after using mask for input

$t3: mask

$t4: number of loop

$t5: number of bit that $t0 will be shifted right to (print\_hex only)

$t6: the result of comparison the value of $t1 and 10

$t8, $t9: save 2 part of input when moving from coprocessor 1 to register

* Procedure:

Print\_bin: print 32 bits of a register to a string in binary form

Print\_hex: print 32 bits of a register to a string in hexadecimal form