

**Programming Assignment #3:**  
Adding floating-point numbers in IA-32 assembly language

Due: May 30, 11:59PM

## 1. Introduction

In this assignment, you need to implement a function in IA-32 assembly language which adds two floating-point numbers. The overall requirement is same as in the first programming assignment. The only difference is that now you have to implement the function in IA-32 assembly language. This assignment aims at introducing various primitive instructions provided by IA-32 assembly language.

## 2. Problem specification

### 2.1 Overview

Complete the file `fpadd2.s` which implements the function `fpadd2()` in IA-32 assembly language. The function `fpadd2()` receives two floating-point numbers and returns their sum. The prototype of `fpadd2()` is as follows:

```
unsigned fpadd2 (unsigned x, unsigned y);
```

The function `fpadd2()` works exactly the same way as `fpadd()` in the first programming assignment. Two arguments, `x` and `y`, indicate the floating-point numbers to be added. Each parameter is given by unsigned integer type, which corresponds to the binary representation of a single-precision floating-point number as defined in the IEEE 754 standard. The return value is also of unsigned integer type, and it should be the binary representation of the sum.

To make the problem simpler, we assume that `x` and `y` are non-negative numbers, i.e.,  $x \geq 0$  and  $y \geq 0$ . If any of the arguments is a negative number, `fpadd2()` simply returns NaN.

In addition, we use the rounding scheme "Round toward 0" in this assignment. Under this rounding scheme, you can just truncate those bits that cannot fit into the significand field (i.e., 23 bits in the single-precision FP representation).

### 2.2 Restrictions

- In the main body of `fpadd2()`, you should not use `%ebx`, `%ecx`, and `%edx` registers. Use `%eax`, `%esi`, and `%edi` registers only. If you are running out of registers, use stack as temporary storage.
- Among the registers you can use, `%esi` and `%edi` registers are callee-save registers. Therefore, you have to save and restore the original values of those registers in `fpadd2()`.
- No IA-32 floating-point instructions (such as `flds`, `fadds`, etc.) are allowed.

### 3. Skeleton codes

The following skeleton codes are provided for this assignment.

Makefile:	This is a file used by the GNU make utility
main.c:	This is a C program which calls <code>fpadd2()</code> . It also prints out the result.
fpadd2.s	This is a skeleton assembly code for <code>fpadd2()</code> . Do not change the first two instructions ( <code>pushl</code> & <code>movl</code> ) and the last three instructions ( <code>movl</code> , <code>popl</code> , & <code>ret</code> ). You are supposed to fill only the main body of this file.

### 4. Hand in instructions

- Make sure you have included your name and the student ID in the header comment of your program.
- Rename the file name of `fpadd2.s` to "`YourStudentID.s`" (e.g., `2008310123.s`).
- Prepare a separate document in PDF format (most preferred, but other formats, such as `.txt`, `.doc`, and `.hwp`, are also allowed), which explains the design and implementation of your code. The document should be named "`YourStudentID.pdf`". The document can be written in Korean if you wish.
- Send a mail to [`cse2003skku@cs.skku.edu`] AND [`cse2003skku@gmail.com`] with attaching two files, "`YourStudentID.s`" and "`YourStudentID.pdf`". The subject line of the mail should be

[`CSE2003-A`] PA#3, YourStudentID, YourName

if you belong to the Class 41 (Korean class), or

[`CSE2003-B`] PA#3, YourStudentID, YourName

if you belong to the Class 42 (English class).

### 5. Logistics

- You will work on this assignment alone.
- The submission status will be posted on the course homepage at <http://cs.skku.edu/CSE2003S10>.
- Only the assignments submitted before the deadline will receive the full credit. 25% of the credit will be deducted for every single day delay.
- Any attempt to copy others' work will result in heavy penalty (for both the copier and the originator). Don't take a risk.

Good luck!

---

Jin-Soo Kim  
Computer Systems Laboratory  
School of Information and Communication Engineering  
Sungkyunkwan University