COM 410, 1799, Computer Architecture
Task #2, Simple Problems (Signed Numbers)

Toksaitov Dmitrii Alexandrovich
toksaitov.d@gmail.com

March 3, 2011

1 General Information

This time you need to solve one task, which was inspired by problem #2 from the *Project Euler* web site. The solution should be presented in *x86* assembly for *real address mode*. You have one week for this task. The solution should be packed and sent to toksaitov.d@gmail.com before the deadline.

2 Problem Description

Part #1

Each next *Fibonacci* number is generated by adding two previous numbers in the sequence.

The first 10 numbers of this sequence are:

$$0, 1, 1, 2, 3, 5, 8, 13, 21, 34, ...$$

Find the sum of the even-valued numbers for the first 29 *Fibonacci* terms.

Part #2

NegaFibonacci numbers are the negatively indexed elements of the *Fibonacci* sequence.

$$F_{-1} = 1, F_{-2} = -1, F_n = F_{n+2} - F_{n+1}$$

The first 10 *negaFibonacci* numbers are:

$$1, -1, 2, -3, 5, -8, 13, -21, 34, -55, ...$$

Find the sum of the odd-valued numbers for the first 28 *negaFibonacci* terms. Use the *sub* and *sbb* instructions to calculate each next *negaFibonacci* term.

Part #3

Implement each solution from part #1 and #2 as a separate procedure in one file. Calculate the sum from results of this procedures.

3 Notes

In this problem it is still not possible to fit solutions into 16-bit registers. To overcome this limitation you need to use multi-precision arithmetic techniques described in the book *The Art of Assembly Language* by Randall Hyde ("Chapter 9: Advanced Arithmetic").

This time you will also need to use the *sbb* instruction (subtract with borrow).

sbb destination, source

Subtracts `source` + carry flag from `destination`.

```
1       sub [a], ax     ; Subtract the value in AX
2                       ; from the lower word in the memory location
3                       ; specified by 'a'
4
5       sbb [a+2], bx   ; Subtract the value in BX and the carry flag
6                       ; from the upper word in the memory location
7                       ; specified by 'a'
```

4 Links

Project Euler
`http://projecteuler.net`

The Art of Assembly Language
`http://bit.ly/asm-art`

x86 Instruction Set Reference
`http://siyobik.info/index.php?module=x86`

Intel® Software Developer's Manuals
`http://www.intel.com/products/processor/manuals/`