CONCORDIA UNIVERSITY

DEPARTMENT OF
COMPUTER SCIENCE AND SOFTWARE ENGINEERING

COMP 228 Winter 2008

ASSIGNMENT 4

Issued: March 4, 2008 Due: March 21, 2008

---

1. Addressing modes [10%]

Suppose a 1-address instruction machine (such as MARIE) has an Add* S instruction where * can refer to (a) immediate, (b) direct, (c) indirect or (d) indexed operand (with a default index register X). Specifically, if * = indexed operand, then the operand address is given by the sum of the index register and the value of S. Consider the following known memory contents:

100	600
…	
400	500
…	
500	400
…	
600	700
…	
700	100

If initially AC = 100, X = 400, determine the content of AC after executing Add* 100 for each of the four cases of *.

Explain briefly the usefulness of these four addressing modes in assembly language programming.

2. Comparison of ISA's [20%]

Referring to the example 5.1 in your text but with a different arithmetic expression:

$Z = (X - Y) * (X + Y + Z)$

(a) Rewrite the program fragments that implement Z for each of the four ISA's (three-address, two-address, one-address and stack computer).

(b) Count the number of memory operand accesses (read or write) required in each case in (a).

(c) Suppose each opcode is 1-byte, each register address is 1 byte and each memory address is 3 bytes in these four computers. Count the number of bytes required in the program code for each of the ISA's.

(d) Compare the performance implications of these different ISA using the time aspect reflected in (a) and the space aspect reflected in (b). Assume typically a memory access takes multiple (say 5) clock cycles.

3. Pipelined processing [10%]

Pipelining is an effective technique in promoting concurrent execution of instructions in a processor. Suppose a nonpipelined processor takes 200ns to process an instruction. The same instruction can be processed in a 4-segment pipeline with a clock cycle of 60ns.

(a) Determine the maximum speedup ratio of the pipeline in executing 200 instructions.

(b) Determine the maximum speedup ratio of the pipeline in executing an infinite number of instructions.

(c) Explain two key factors that make the maximum speedup ratio (in (b)) unattainable in practice.

4. NASM Program Understanding [20%]

Consider the following NASM program fragment:

```
    mov   ecx, 100
    mov   ebx, X
    sub   ax, ax
S1  add   ax, [ebx]
    add   ebx, 2
    dec   ecx
    jnz   S1
    mov   [Y], ax
```

(a) How many times is the instruction *'jnz S1'* executed?
(b) What is the value of ebx when the instruction *'mov [Y], eax'* is executed?
(c) Explain in a simple sentence what is the result computed in ax by the given program fragment.
(d) Rewrite the program fragment using MARIE instruction set ***or*** explain the difficulty you encounter in trying to do so.

5. Input/Output in Intel assembly language [Do this program in the lab.] [40%]

In this programming assignment, you are going to modify the Hello-World program example (class slide #41) ***in the code segment only*** so that the resulting program will display the same output message but without the spaces in between the words in the message [i.e., the display will be HelloWorld!PleaseTypeYourID]. You are not allowed to change the data segment in the original program.

Assemble your program using the NASM assembler and run the resulting executable. Debug the program until it works correctly. Submit a correctly working version of your NASM program along with its output illustrating its correctness.