CMPT 334 **Spring 2021**

Assignment 5

Due: Tuesday May 04, 2021 – 8:00 AM

Submit a single pdf/word file in Moodle.

Provide short answers for the following question (100 points):

Q1. (30 points) Briefly define the following:

(a) Immediate addressing: the value of the operand is in the instruction.
(b) **Direct addressing:** the address field contents is the effective address of the operand.
(c) **Indirect addressing:** the address field refers to the address of a word in memory, which in tum contains the effective address of the operand.
(d) **Register addressing:** the address field refers to a register that contains the operand.
(e) **Register indirect addressing:** the address field refers to a register, which in turn contains the effective address of the operand.
(f) **Displacement addressing:** the instruction has two address fields, at least one of which is explicit. The value contained in one address field (value = A) is used directly. The other address field refers to a register whose contents are added to A to produce the effective address.
(g) **Relative addressing:** the implicitly referenced register is the program counter (PC). That is, the current instruction address is added to the address field to produce the EA.

Q2. (10 points) Given the following memory values and a one-address machine with an accumulator, what values do the following instructions load into the accumulator?

Address	Content
20	40
30	50
40	60
50	20

a. LOAD IMMEDIATE 40
40
b. LOAD DIRECT 40
60
c. LOAD INDIRECT 30
20
d. LOAD IMMEDIATE 50
50
e. LOAD DIRECT 50
20
f. LOAD INDIRECT 50
40

Q3. (10 points) Consider an indirect-addressing-mode, how many times does the processor need to refer to memory when it fetches and executes an instruction with a single operand. Briefly explain.

Three times to fetch instruction; fetch operand reference; fetch operand.

Q4. (10 points) Consider the following NASM code fragment, what is the value assigned to the symbol L? Briefly explain your answer.

```
msg db 'hello, world'
L equ $ - msg
```

L is assigned the constant 12. This is because $ represent the current assembly position at the beginning of the line, then we subtract the starting position of the message. This basically gets the length of this text message.

Q5. (15 points) Write an equivalent NASM code fragment to the following C code fragment:

```
if (eax == 0) ebx = ebx + 1;
else ebx = 0;
```

```
cmp eax, 0
je thenblock
add ebx, 1
jmp next
thenblock:
      mov ebx, 0
next:
```

Q6. (15 points) Write an equivalent NASM code fragment to the following C code fragment:

Hint: you will use directives (resd, dd) and instructions (idiv, mov, add)

```
int avg;
int x = 10;
int y = 5;
avg = (x + y) / 2;
```

```
avg: resd 1; integer average
x: dd 5; first number in the average
y: dd 10; second number in the average
mov avg, x
add avg, y
idiv avg, 2; get integer average
```

Q7. (10 points) List the advantages and disadvantages of using assembly language over high-level languages?

Disadvantages:

1. Development time.
2. Reliability and security.
3. Debugging and verifying.
4. Maintainability.
5. Portability.
6. System code can use intrinsic functions instead of assembly.
7. Application code can use intrinsic functions or vector classes instead of assembly.
8. Compilers have been improved a lot in recent years.

Advantages:

1. Debugging and verifying.
2. Making compilers.
3. Embedded systems.
4. Hardware drivers and system code.
5. Accessing instructions that are not accessible from high level language.
6. Self-modifying code.
7. Optimizing code for size.
8. Optimizing code for speed.