Name: KEY **Id#**

COE 205, Term 062

Computer Organization & Assembly Programming

Quiz# 1

Date: Saturday, March 3, 2007

Q1. What is the Instruction Set Architecture (ISA) of a computer?

The **Instruction Set Architecture** (ISA) of a computer consists of ***Instruction Set***, ***Memory***, and ***Programmer-accessible registers***.

Q2. What is Assembly Language and how it is different from Machine Language?

Assembly language is the set of instructions a CPU can execute using symbolic names to represent operations, registers and memory locations. Each assembly language instruction corresponds to a single machine language instruction. Machine language is the set of instructions that a CPU can execute represented in binary format.

Q3. Give two advantages for programming in Assembly Language and two advantages for programming in High-Level Language.

Two advantages for programming in assembly language are:

1. Accessibility to system hardware
2. Space and time efficiency

Two advantages for programming in High-Level language are:

1. Program development is faster and program maintenance is easier
2. Programs are portable

Q4. Fill the blanks in the following questions:

(i) Assuming **8-bit 2`s complement** representation, the smallest (negative) number is 10000000 in binary and -128 in decimal and the largest (positive) number is 01111111 in binary and +127 in decimal**.**

(ii) Consider an **8-bit** register that has the binary number 10010110. The decimal value of this number as a signed number in sign-magnitude representation is -22 while in 1's complement representation it is -105 and in 2's complement representation it is -106.

(iii) Assuming **8-bit 2`s complement** representation, the number F0 represents the decimal number -16.

(iv) The binary number 01100100 represents character **d** and uses an **ODD** parity bit. Note that the ASCII code of character **A** is 41H and that of character **a** is 61H.

Q5. Perform the following arithmetic operations assuming that numbers are represented using **8-bit 2's complement** representation. Indicate in your answer when an *overflow* occurs.

i. 8F + FC = 8B

There is no overflow since we are adding two negative numbers and we got a negative number.

ii. 6E – E0= 6E + 20 = 8E

There is overflow since we are adding two positive numbers and we got a negative number.