Oct. 28, 2010

COMPUTER ENGINEERING DEPARTMENT

COE 205

COMPUTER ORGANIZATION & ASSEMBLY PROGRAMMING

Major Exam I

First Semester (101)

Time: 1:00-3:00 PM

Student Name : \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Student ID. : \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |  |  |
| --- | --- | --- |
| **Question** | **Max Points** | **Score** |
| **Q1** | **76** |  |
| **Q2** | **12** |  |
| **Q3** | **12** |  |
| **Total** | **100** |  |

Dr. Aiman El-Maleh

#  **[76 Points]**

# **(Q1)** Fill the blank in each of the following:

##

## Two advantages of programming in assembly language include \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ and \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ while two disadvantages include\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ and \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.

## There is one-to-one correspondence between \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ instructions and \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ instructions.

## The \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_combines program's object filewith other object files and link libraries, and produces a single executable program.

## The \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ provides a hardware/software interface.

## The 8086 processor is a \_\_\_\_\_\_\_\_\_\_\_\_\_ bit machine with an address bus of \_\_\_\_\_\_\_\_\_\_\_ bits and a data bus with \_\_\_\_\_\_\_\_\_\_\_\_ bits.

## With a 36-bit address bus, the physical address space is \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.

## \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ is used to bridge the CPU-memory performance gap.

## In protected mode, the linear address is computed based on adding \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ with \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ obtained from \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ indexed by \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.

## A segment descriptor table stores for each segment \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_, \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ and \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.

## Given that variables I and J are **Dword** variables, the equation I = I + 2\*J can be implemented using the following minimal instructions:

## \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## Given that variable I is defined as **Qword**, the equation I = I + 5 can be implemented using the following minimal instructions:

## \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## Given a magnetic disk with the following properties: Rotation Speed = 4000 RPM (rotations per minute), Average Seek = 4 ms, Sector = 512 bytes, Track = 500 sectors. The average time to access a block of 100 consecutive sectors is \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## The integer number -1000 is represented in hexadecimal using 16-bit 2’s complement representation as \_\_\_\_\_\_\_\_\_\_\_\_\_\_.

## Assuming 16-bit 2’s complement representation, the hexadecimal number FE00 represents the decimal number\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.

## Assuming 8-bit 2’s complement representation, the largest number that can be stored is \_\_\_\_\_\_\_\_\_\_\_\_ in decimal and \_\_\_\_\_\_\_\_\_\_\_ in binary and the smallest number that can be stored is \_\_\_\_\_\_\_\_\_\_\_\_ in decimal and \_\_\_\_\_\_\_\_\_\_in binary.

## Given that the number A80h is represented using 16-bit 2’s complement representation, the equivalent number represented using 32-bit 2’s complement representation is \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.

## Given that register AL=E6 stores an ASCII character, then the stored character is \_\_\_\_\_\_\_\_ and the used parity is \_\_\_\_\_\_\_\_\_\_. Note that ‘A’=41h and ‘a’=61h.

## Given that the instruction MOV EAX, 10 (having the machine code B8 0000000A) is stored at address 00000008, then the address of the next instruction to be fetched from memory is \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.

## Given a processor with a 5-stage pipeline and clock frequency of 4 GHZ, the time that will be required to execute a program of 8 billion instructions assuming that there will be no pipeline stalls is nearly \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ seconds.

## Assume that in real mode the range of addresses from 00000 to 00109 is used by another program. Given that a program has a code segment of 4 Kbyte and a data segment of 2 Kbyte, the code segment number allocated is \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ and the data segment number allocated is \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.

## Assume that DS=0010, CS=0030, ES=0060, SS=0080, IP=0035, BX=00AF, and SP=020C. Based on 16-bit real-mode addressing, the linear address of the next instruction to be fetched from memory is \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.

## Assume that DS=0010, CS=0030, ES=0060, SS=0080, IP=0035, BX=00AF, and SP=020C. Based on 16-bit real addressing mode, the linear address of the source operand in the instruction MOV AX, [BX] is \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.

## The addressing mode of the source operand in the instruction MOV EAX, Array[10] is \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.

## Assume that AX=0000h. Executing the instruction DEC AL produces the result AX=\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.

## The addressing mode of the source operand in the instruction MOV EAX, [ESI] is \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.

## The assembler allocates \_\_\_\_\_\_\_\_ bytes for the variable *Array* defined below:

## *Array Dword 10 dup(10 dup(0))*

##

## Assuming the following data segment and assuming that variable X is given the linear address 00404000h, then the linear address for variables Y and Z will be \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ and \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.

## .DATA

## X BYTE 1, 2, 3

## ALIGN 2

Y WORD 4, 5

## ALIGN 4

## Z DWORD 6, 7

## Assuming the following data segment and assuming that variable X is given the linear address 00404000h, then the content of register EAX after executing the instruction MOV EAX, Dword PTR X+2 is \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.

## .DATA

## X WORD 1, 2, 3, 4

## Assuming the following data segment and assuming that variable X is given the linear address 00404000h , after executing the code given below, the content of register EAX=\_\_\_\_\_\_\_\_\_\_\_ and EBX=\_\_\_\_\_\_\_\_\_\_\_\_\_\_.

## .DATA

## X Byte 10, 20, 30

## Y WORD 30, 40, 50

## .CODE

## MOV EAX, TYPE X

## MOV EBX, OFFSET Y

##  After executing the code given below, the content of registers EAX and EBX will be \_\_\_\_\_\_\_\_\_\_\_\_\_\_ and \_\_\_\_\_\_\_\_\_\_\_\_\_.

## .DATA

## ARRAY WORD 10, 20

 WORD 30, 40

## .CODE

## MOV EAX, LENGTHOF ARRAY

## MOV EBX, SIZEOF ARRAY

## Assume that AL=A0h. Executing the instruction MOVSX EAX, ALproduces the result EAX=\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.

## Assume that AX=0F90h. Executing the instruction *NEG AX* produces the following results: AX=\_\_\_\_\_\_, overflow flag=\_\_\_, sign flag=\_\_\_, zero flag=\_\_\_, carry flag=\_\_\_, auxiliary flag=\_\_\_ and parity flag=\_\_\_.

## Assume that AX=800Fh and BX=FFF1h. Executing the instruction *ADD AX, BX* produces the following results: AX=\_\_\_\_\_\_, overflow flag=\_\_\_, sign flag=\_\_\_, zero flag=\_\_\_, carry flag=\_\_\_, auxiliary flag=\_\_\_ and parity flag=\_\_\_.

##  Assume that AX=800Fh and BX=00F1h. Executing the instruction *SUB AX, BX* produces the following results: AX=\_\_\_\_\_\_, overflow flag=\_\_\_, sign flag=\_\_\_, zero flag=\_\_\_, carry flag=\_\_\_, auxiliary flag=\_\_\_ and parity flag=\_\_\_.

#  **[12 Points]**

# **(Q2)** Consider a program that has the following data segment assuming a flat memory model:

*A EQU 10*

*B BYTE 11, 12*

*C WORD 13*

*D DWORD 14*

Indicate whether the following are valid **IA-32** instructions or not. **If invalid, give the reason**:

## 1. MOV AX, B

## 2. MOV B, A

## 3. MOV EAX, C+1

## 4. MOV AL, [EBX]

## 5. MOV AX, OFFSET C

## 6. MOVSX EAX, AX

## 7. ADD [EAX], A+1

## 8. MOV DS, ES

**[12 Points]**

# **(Q3)** Suppose that the following directives are declared in the data segment with a starting linear address of 00404000. Show the linear addresses of allocated memory and their corresponding content in hexadecimal. Note that the ASCII code for character ‘a’ is 61h and that of character ‘A’ is 41h. The ASCII code of character ‘0’ is 30h.

I BYTE -1, 100, “AB”

 WORD -16

J DWORD -30

K EQU 10

L BYTE K-15

 BYTE 5, 3 dup(1,2)

|  |  |  |
| --- | --- | --- |
| Variable | Linear Address (Hex.) | Content (Hex.) |
| I | 00404000 |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

##