

Nov. 3, 2007

COMPUTER ENGINEERING DEPARTMENT

COE 205

COMPUTER ORGANIZATION & ASSEMBLY PROGRAMMING

Major Exam I

Second Semester (071)

Time: 7:30-9:30 PM

Student Name : _____

Student ID. : _____

Question	Max Points	Score
Q1	50	
Q2	10	
Q3	15	
Q4	25	
Total	100	

Dr. Aiman El-Maleh

[50 Points]

(Q1) Indicate whether the following is **true** or **false**, and if it is false **correct** it (correct the answer and not the question):

- (1) (True, False) The smallest (negative) number that can be represented using 16-bit 2's complement in hexadecimal is FFFF and the largest positive number in hexadecimal is 7FFF.

- (2) (True, False) Assume that the CPU has just read a 64-bit instruction from the linear address 00404000H. Then, the linear address of the next instruction that this CPU is going to read is 00404002H.

- (3) (True, False) The 8086 processor is a 16-bit machine with an address and data bus of 16 bits while the Pentium IV processor is a 32-bit machine with an address and data bus of 32 bits.

- (4) (True, False) For addition operations, an end carry out of the most significant bit indicates incorrect result for both signed and unsigned numbers.

- (5) (True, False) With a 32-bit address bus and 64-bit data bus, the maximum memory size that can be accessed by a processor is 4GByte and the maximum number of bytes that can be read or written in a single cycle is 8 Bytes.

(6) (True, False) The addressing mode of the source operand in the instruction `MOV AX, [ESI]` is register addressing mode.

(7) (True, False) The addressing mode of the source operand in the instruction `MOV EAX, offset MSG` is direct addressing mode.

(8) (True, False) Assuming 8-bit representation of numbers, the binary number 10100100 is equal to -36 in sign-magnitude representation, -91 in 1's complement representation, and -92 in 2's complement representation.

(9) (True, False) Assuming variable `Array` is defined as shown below:

Array WORD 10h, 20h

The content of register `AX` after executing the instruction `MOV AX, Array+1` will be `20h`.

(10) (True, False) The assembler allocates 17 bytes for the variable `Array` defined below:

Array WORD 5, 4 dup(2, 3 dup(0))

- (11) (True, False) Assume that DS=12FF, CS=E6F0, ES=F135, SS=ABCD, IP=0016, and SI=526F. Based on 16-bit addressing in real mode, the linear address of the next instruction to be fetched from memory is E6F16.
- (12) (True, False) Assuming that AL=FEh and BL=00h, the two instructions ADD AL, 1 and SUB BL, 1 produce the same result in registers AL and BL and the same effect on flags.
- (13) (True, False) The instruction set architecture of a processor consists of its control unit, data path, memory, and the instruction set.
- (14) (True, False) Assume that AX=8111h and BX=F265h. Executing the instruction *ADD AX, BX* sets the overflow flag and the carry flag to 1, while it sets the sign flag, the zero flag, the parity flag, and the auxiliary flag to 0.
- (15) (True, False) Assume that AX=8111h and BX=F265h. Executing the instruction *SUB AX, BX* sets the parity flag, the auxiliary flag, the carry flag and the sign flag to 1, while it sets the zero flag and the overflow flag to 0.

(16) (True, False) Assume that $AX=F0F0h$. Executing the instruction *NEG AX* produces the result $AX=0F0Fh$.

(17) (True, False) Assume that $AX=00FFh$. Executing the instruction *INC AL* produces the result $AX=0100h$.

(18) (True, False) Assume that $AX=009Fh$. Executing the instruction *MOVSX BX, AL* produces the result $BX=FF9Fh$.

(19) (True, False) Assuming that $AX=4$ and given the following definition of *ARRAY*:

ARRAY WORD 1, 2, 3

Execution the code below does not change the content of *AX* and changes the content of *ARRAY* to:

ARRAY WORD 3, 2, 1

```
XCHG AX, ARRAY[0]
XCHG AX, ARRAY[4]
XCHG AX, ARRAY[0]
```

(20) (True, False) After executing the code shown below, the content of register *AX* will be 32 and the content of register *CX* will be 0.

```
MOV CX, 4
MOV AX, 2
NEXT:
ADD AX, AX
LOOP NEXT
```

(21) (True, False) Given a magnetic disk with the following properties:

- Rotation speed = 7200 RPM (rotations per minute)
- Average seek = 8 ms, Sector = 512 bytes, Track = 200 sectors

The average time to access a block of 64 consecutive sectors is 13.5 ms.

(22) (True, False) As part of the instruction set architecture of the Pentium-IV processor, it has Six 32-bit general-purpose registers, Four 16-bit segment registers in addition to Processor Status Flags (EFLAGS), Instruction Pointer (EIP) and Instruction Register (IR).

(23) (True, False) Assuming that the instruction execution cycle is composed of a five-stage pipeline as follows: Instruction Fetch, Instruction Decode, Operand Fetch, Instruction Execute, Result Writeback. Assume that each stage requires one clock cycle to complete. Then, executing 10 instructions using this pipeline will require 50 clock cycles.

(24) (True, False) Assuming the following data segment, and assuming that the first variable X is given the linear address **00404000h**, then the linear address for variable Y will be **00404003h**.

```
.DATA
X      BYTE 10,11,12
ALIGN 2
Y      WORD 13
```

(25) (True, False) Assuming the following data segment, the content of register ECX=0000000C after executing the instruction **MOV ECX, SIZEOF MSG**.

```
.DATA
MSG    WORD 10 DUP(0), 5, 10
```

[10 Points]

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

<i>I</i>	<i>EQU</i>	<i>255</i>
<i>J</i>	<i>BYTE</i>	<i>-1</i>
<i>K</i>	<i>WORD</i>	<i>I</i>

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

1. MOV AX, J+1

2. MOV BH, offset J

3. MOV DS, I+1

4. MOV ES, K

5. MOV [EBX], 1

6. ADD [ECX], AL

7. NEG [EAX]

8. MOV AH, AL+1

9. MOVSX EAX, j

10. DEC DS

[15 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>I</i>	<i>BYTE</i>	-5, 251, '5a'
	<i>WORD</i>	-5, 0EFFH
<i>J</i>	<i>DWORD</i>	-120, 120
	<i>WORD</i>	17
<i>K</i>	<i>EQU</i>	32H
<i>L</i>	<i>BYTE</i>	K+5, K*5
	<i>BYTE</i>	3, 2 dup(1,-1)

[illegible]

[25 Points]

(Q4) Write separate assembly programs to do the following using the smallest possible instructions. You do not need to show the full structure of the program, just show the needed assembly instructions in your solution.

- (i) Assume that you have an Array of integers, declared as IntArray, with each integer defined as a **Word**. Write an assembly program to **reverse** the content of IntArray. **Your program should work for any array size.**

For example, assume the following array definition:

IntArray Word 1, 2, 3, 4, 5, 6

After executing the program, the content of IntArray will be:

IntArray Word 6, 5, 4, 3, 2, 1

- (ii) Assume that you have a two-dimensional array of integers, declared as TwoDArray, with each integer defined as a **Byte**. Write an assembly program to **increment each integer in row i by the value i+1**. Assume that the number of rows and number of columns in the array are defined in the constants NRow and NCol, respectively. **Your program should work for any array size.**

For example, assume the following array definition:

NRow EQU 4

NCol EQU 5

TwoDArray Byte 1, 2, 3, 4, 5
 Byte 6, 7, 8, 9, 10
 Byte 11, 12, 13, 14, 15
 Byte 16, 17, 18, 19, 20

After executing the program, the content of TwoDArray will be:

TwoDArray Byte 2, 3, 4, 5, 6
 Byte 8, 9, 10, 11, 12
 Byte 14, 15, 16, 17, 18
 Byte 20, 21, 22, 23, 24

