Nov. 10, 2009

COMPUTER ENGINEERING DEPARTMENT

COE 205

COMPUTER ORGANIZATION & ASSEMBLY PROGRAMMING

Major Exam I

First Semester (091)

Time: 8:00-10:00 PM

Student Name : _KEY_____

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:

(1) Assembly language is a programming language that uses symbolic names to represent operations, registers and memory locations.

(2) In an assembly instruction, the opcode specifies the particular operation to be performed.

(3) One of the main advantages of programming in high level language is that programs are portable.

(4) One of the main advantages of programming in assembly language is having smaller code size and faster execution.

(5) Development of a business application for multiple platforms is better done using high level language.

(6) The linker combines program's object file with other object files and libraries, and produces a single executable program.

(7) The instruction set architecture of a processor is composed of the instruction set, programmer-accessible registers, and memory.

(8) With the Pentium IV processor having a 36 bit address bus and a 64 bit data bus, the physical address space is 2^{36}=64 G bytes.

(9) The processor is interfaced with memory and input/output devices using data bus, address bus and control bus.

(10) The processor is composed of data path unit and control unit.

(11) Given a processor with 2 GHz clock frequency and a program executing in 1 billion clock cycles, the execution time of the program is $(1/2x10^9) *10^9=0.5$ sec.

(12) Cache memory is useful in reducing the memory access time, i.e. the time needed for reading instructions and data from memory (DRAM).

(13) Given a magnetic disk with the following properties: Rotation Speed = 6000 RPM (rotations per minute), Average Seek = 5 ms, Sector = 512 bytes, Track = 200 sectors. The average time to access a block of 100 consecutive sectors is
Average access time= Seek Time + Rotation Latency + Transfer Time
Rotations per second=6000/60 =100 RPS
Rotation time in milliseconds=1000/100=10 ms
Time to transfer 100 sectors=(100/200)* 10=5 ms
Average access time=5 + 5 + 5 =15 ms.

(14) The integer number -1500 is represented in hexadecimal using 16-bit 2's complement representation as FA24.

(15) Assuming 16-bit 2's complement representation, the hexadecimal number FE10 represents the decimal number -496.

(16) Assuming 8-bit 2's complement representation, the largest number that can be stored is +127 in decimal and 7F in hexadecimal and the smallest number that can be stored is -128 in decimal and 80 in hexadecimal.

(17) Given that the number F6 is represented using 8-bit 2's complement representation, the equivalent number represented using 16-bit 2's complement representation is FFF6.

(18) Given that register AL=E5 stores an ASCII character, then the stored character is e and the used parity is odd. Note that ‘A’=41h and ‘a’=61h.

(19) The Pentium-IV processor contains the segment registers CS, SS, DS, ES, FS, GS.

(20) The EIP register holds the next instruction to be fetched from memory.

(21) After an instruction is fetched from memory, the EIP register is incremented by the size of the fetched instruction.

(22) During the execution phase of an instruction, the following tasks are performed: instruction decoding, operand fetch, executing the instruction, writing back the result.

(23) Given a 10-stage pipeline where each stage executes in one clock cycle, executing 100 instructions without any pipeline stalls will require 109 cycles.

(24) A superscalar processor has multiple execution pipelines.

(25) In real addressing mode, assume that the code segment occupies the address range from 0A6F0 to 0E007. Then, the code segment number is 0A6F, the size of the code segment is 3918h=14616 bytes and the next available free segment is 0E01.

(26) Assume that DS=1234, CS=5678, ES=9ABC, SS=DEF0, IP=01F4, BX=E678, and SP=2314. Based on 16-bit real-mode addressing, the linear address of the next instruction to be fetched from memory is 56780+01F4=56974.

(27) Assume that DS=1234, CS=5678, ES=9ABC, SS=DEF0, IP=01F4, BX=E678, and SP=2314. Given that offset of variable I is 001F, based on 16-bit real addressing mode, the linear address of the source operand in the instruction MOV AX, I is 12340+001F=1235F.

(28) In protected mode, the segment unit translates logical address to linear address while the paging unit translates linear address to physical address.

(29) In protected mode, the segment descriptor table stores the following information for each segment base address, segment limit, access rights.

(30) The assembler allocates 10*(10+2)+1=121 bytes for the variable *String* defined below:

String Byte 10 dup(10 dup(0),10,13),0

(31) The content of register EAX after executing the following code segment is 00000017.

```
I=10
MOV EAX, I
I=I+3
ADD EAX, I
```

(32) 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 00404004h and 0040400Ch.

```
.DATA
X       BYTE 1, 2, 3
ALIGN 2
Y       WORD 4, 5, 6
ALIGN  4
Z       DWORD  7,  8, 9
```

(33) 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 AX, Y-8 is 0001.

```
.DATA
X       BYTE "MAJOR EXAM I"
        DWORD 1, 2
Y       WORD  3, 4
```

(34) 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=2 and EBX=00404004h.

```
.DATA
X       WORD  1, 2
Y       DWORD 3, 4
.CODE
MOV EAX, TYPE X
MOV EBX, OFFSET Y
```

(35) After executing the code given below, the content of registers EAX and EBX will be 2 and 4.

```
.DATA
ARRAY       WORD  1, 2
            WORD  3, 4
            WORD  5, 6
.CODE
MOV EAX, LENGTHOF ARRAY
MOV EBX, SIZEOF ARRAY
```

(36) After executing the code given below, the content of register EAX will be 00040003.

```
.DATA
ARRAY WORD 1, 2, 3, 4, 5, 6, 7, 8
.CODE
MOV EAX, DWORD PTR ARRAY+4
```

(37) Assuming variable ARRAY is defined as shown below:

```
ARRAY  DWORD 1, 2, 3, 4, 5, 6, 7, 8
```

The content of register AX after executing the instruction MOV AX, WORD PTR ARRAY+1 will be 0000.

(38) Assume that AX=AB4D. Executing the instruction MOVSX EAX, AX produces the result EAX=FFFFAB4D.

(39) Assume that AX=EABF and BX=FF6F. Executing the instruction *ADD AX, BX* produces the following results: AX=EA2E, overflow flag=0, sign flag=1, zero flag=0, carry flag=1, auxiliary flag=1 and parity flag=1.

(40) Assume that AX= EABF and BX= FF6F. Executing the instruction *SUB AX, BX* produces the following results: AX=EB50, overflow flag=0, sign flag=1, zero flag=0, carry flag=1, auxiliary flag=0 and parity flag=1.

[12 Points]

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

```
X       EQU     1
Y       EQU     AH
Z       BYTE    1, 2
W       WORD    X-2, X+2
```

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

1. MOV AL, Z+2

Valid. (memory to register of the same size)

2. MOV Z, Y

Valid. (register to memory of the same size)

3. MOV DS, ES

Invalid. Segment register cannot be moved directly to another segment register.

4. MOV ES, X+1

Invalid. Immediate value cannot be moved directly to a segment register.

5. MOV W, OFFSET Z

Invalid. Size mismatch as offset is 32-bits while W is 16-bits.

6. MOVSX W, AL

Invalid. The destination in MOVSX must be a register.

7. MOV W, Word PTR Z

Invalid. Memory to memory is not allowed.

8. ADD DS, AX

Invalid. Segment registers cannot be operands for the ADD instruction.

[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      “EXAM  I”,0
        WORD      15, -15
J       DWORD     100, -100
K       EQU       1Fh
L       BYTE      K+1
        BYTE      2, 3 dup(-2)
```

Variable	Linear Address (Hex.)	Content (Hex.)
I	00404000	‘E’
	00404001	‘X’
	00404002	‘A’
	00404003	‘M’
	00404004	‘ ‘
	00404005	‘I’
	00404006	0
	00404007	0F
	00404008	00
	00404009	F1
	0040400A	FF
J	0040400B	64
	0040400C	00
	0040400D	00
	0040400E	00
	0040400F	9C
	00404010	FF
	00404011	FF
	00404012	FF
L	00404013	20
	00404014	02
	00404015	FE
	00404016	FE
	00404017	FE