

## Chapter-01: Introduction to the Microprocessor and Computer.

1. Who developed the Analytical Engine?

Charles Babbage

2. The 1890 census used a new device called a punched card. Who developed the punched card?

Herman Hollerith

3. Who was the founder of IBM Corporation?

Herman Hollerith

4. Who developed the first electronic calculator?

Konrad Zuse

5. The first electronic computer system was developed for what purpose?

To decode the Enigma code during World War II

6. The first general-purpose, programmable computer was called the \_\_\_\_\_.

ENIAC

7. The world's first microprocessor was developed in 1971 by \_\_\_\_\_.

Intel Corporation

8. Who was the Countess of Lovelace?

Augusta Ada Byron

9. Who developed the first high-level programming language called FLOWMATIC?

Grace Hopper

10. What is a von Neumann machine?

A machine that stores the instructions of a program in the memory system.

11. Which 8-bit microprocessor ushered in the age of the microprocessor?

8080

12. The 8085 microprocessor, introduced in 1977, has sold \_\_\_\_\_ copies.

200 million

13. Which Intel microprocessor was the first to address 1M bytes of memory?

8026/8088

14. The 80286 addresses \_\_\_\_\_ bytes of memory.

16M bytes

15. How much memory is available to the 80486 microprocessor?

4G bytes

16. When did Intel introduce the Pentium microprocessor?

1993

17. When did Intel introduce the Pentium Pro processor?

1995

18. When did Intel introduce the Pentium 4 microprocessor?

2000

19. Which Intel microprocessor addresses 1T of memory?

80486 through the Core2

20. What is the acronym MIPS?

Millions of instructions per second

21. What is the acronym CISC?

Complex Instruction Set Computer

22. A binary bit stores a(n) 1 or a(n) 0. A binary bit stores a 1 or a 0.

23. A computer K (pronounced kay) is equal to 1024 bytes.

1024

24. A computer M (pronounced meg) is equal to 1024 K bytes.

1024K

25. A computer G (pronounced gig) is equal to 1024 M bytes.

1024

26. A computer P (pronounced peta) is equal to 1024 T bytes.

1024

27. How many typewritten pages of information are stored in a 4Gbyte memory?

1,000,000

28. The first 1M byte of memory in a DOS-based computer system contains a(n) System area and a(n) Transient program area.

System area and Transient program area

29. How large is the Windows application programming area? 2Giga or 3Giga for 32-bit mode and currently 8Giga for 64-bit mode

30. How much memory is found in the DOS transient program area?

640K

31. How much memory is found in the Windows systems area?

1Giga

32. The 8086 microprocessor addresses \_\_\_\_\_ bytes of memory.

1M

33. The Core2 microprocessor addresses \_\_\_\_\_ bytes of memory.

Currently **1 Tera** byte using a 40-bit address

34. Which microprocessors address 4G bytes of memory?

**80386, 80486, Pentium, Pentium Pro, PII, PIII, P4, and Core2**

35. Memory above the first 1M byte is called \_\_\_\_\_ memory. Protected memory or extended memory

36. What is the system BIOS?

The basic I/O system

37. What is DOS?

An early operating system called the **Disk Operating System**

38. What is the difference between an XT and an AT computer system?

**The XT was used with the 8088 and 8086 and beginning with the 80286, the AT became the name of the system.**

39. What is the VESA local bus?

**Video Electronics Standards Association**

40. The ISA bus holds \_\_\_\_\_ -bit interface cards.

8-bit and 16-bit

→ 8 bit and 16 bit.

41. What is the USB?

**Universal Serial Bus**

42. What is the AGP?

**The advanced graphics port** is designed to support video cards.

43. What is the XMS?

**Extended Memory System**

44. What is the SATA interface and where is it used in a system? The serial ATA interface is designed to support disk drive memory

45. A driver is stored in the \_\_\_\_\_ area.

**System Area**

46. The personal computer system addresses \_\_\_\_\_ bytes of I/O space.

**64K**

47. What is the purpose of the BIOS?

**The BIOS controls the computer at its most basic level and provides for compatibility between computers.**

48. draw the block diagram of a computer system.

See Figure 1-6.

CHAPTER 1

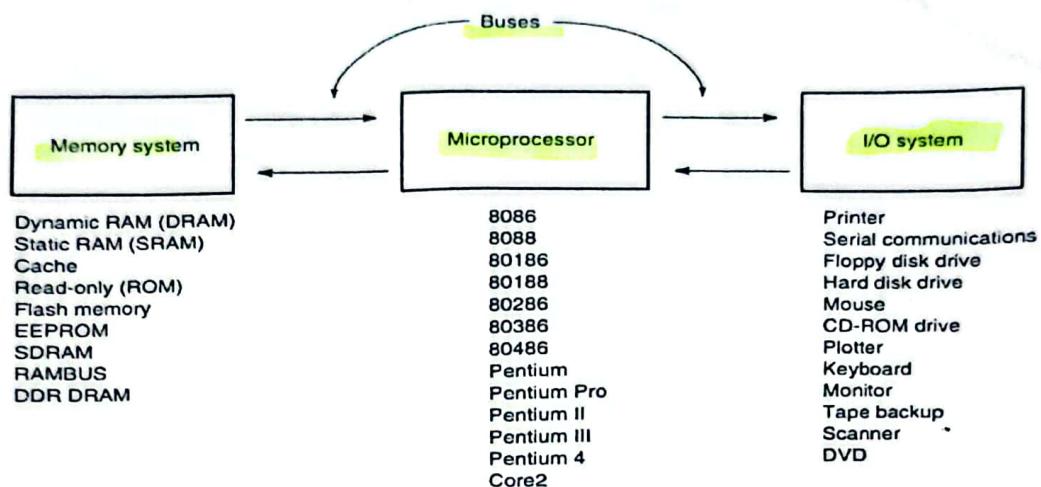


FIGURE 1-6 The block diagram of a microprocessor-based computer system.

49. What is the purpose of the microprocessor in a microprocessorbased computer?

The microprocessor is the **controlling element** in a computer system.

50. List the three buses found in all computer systems.

**Address, data, and control buses**

51. Which bus transfers the memory address to the I/O device or to the memory?

**Address bus**

52. Which control signal causes the memory to perform a read operation?

**MRDC**

53. What is the purpose of the IORC signal?

**The I/O read signal causes an I/O device to be read.**

54. If the MRDC signal is a logic 0, which operation is performed by the microprocessor?

**Memory read operation**

55. Define the purpose of the following assembler directives:

(a) DB

(b) DQ

(c) DW

(d) DD

(a) defines a **byte** or bytes of memory

(b) defines a **quadword** or quadwords of memory

83. Use the Internet to write a short report about any one of the following computer pioneers:

- (a) Charles Babbage
- (b) Konrad Zuse
- (c) Joseph Jacquard
- (d) Herman Hollerith

84. Use the Internet to write a short report about any one of the following computer languages:

- (a) COBOL → Common Business Oriented Language, Developed in 1959
- (b) ALGOL → Algorithmic Language, dev.: 1950
- (c) FORTRAN → FORM Formula Translation, dev: 1950
- (d) PASCAL → B., dev : 1960 .

85. Use the Internet to write a short report detailing the features of the Itanium 2 microprocessor.

86. Use the Internet to detail the Intel 45 nm (nanometer) fabrication technology

## Chapter-02: The Microprocessor and it's Architecture. - - -

1. What are program-visible registers?

Program visible register are the registers that are directly used in an instruction, [pentium - 4]

2. The 80286 addresses registers that are 8 and \_\_\_\_\_ bits wide.

16

3. The extended registers are addressable by which microprocessors?

The 80386 through the Core2

4. The extended BX register is addressed as \_\_\_\_\_.

EBX

5. Which register holds a count for some instructions?

CL, CX, ECX, or RCX

6. What is the purpose of the IP/EIP register (Exam)

Holds the offset address of the next step in the program. The IP/EIP register is used to store the address of the next instruction. The instruction pointer, which points to the next instruction in a program, is used by the microprocessor to find the next sequential instruction in a program located within the code segment.

7. The carry flag bit is not modified by which arithmetic operations?

INC and DEC. The carry flag bit is not modified by multiplication and division operations.

8. Will an overflow occur if a signed FFH is added to a signed 01H?

No, if you add +1 and -1 you have zero, which is a valid number

A number that contains 3 one bits is said to have Odd parity.

Odd

10. Which flag bit controls the INTR pin on the microprocessor?

The I-flag. / The I(interrupt) flag controls the INTR(interrupt request) pin. If I = 1, INTR pins enabled, if I = 0, the INPR is disabled.

11. Which microprocessors contain an FS segment register?

The 80386 through the Core2

12. What is the purpose of a segment register in the real mode operation of the microprocessor?

The segment register addresses the lowest address in a 64K memory segment. /

The Segment Registers are the additional registers which generates addresses when combined with other registers in the microprocessor. In protected mode, segment registers simply point to data structures called segment descriptors that contain the information needed to access a physical memory location.

The Segment registers are the following :

- CS(code) : The code segment register defines the starting address of the section of memory holding code. In real mode operation, it defines the start of a 64K- byte section of memory; in protected mode, it selects a descriptor that describes the starting address and length of a section of memory holding code
- 2. DS(Data): The data segment is a section of memory that contains most data used by a program. Data are accessed in the data segment by an offset address or the contents of other registers that hold the offset address.
- 3. ES (Extra) : The extra segment is an additional data segment that is used by some of the string instructions to hold destination data.
- 4. SS (Stack) : The stack segment defines the area of memory used for the stack. The stack entry point is determined by the stack segment and stack pointer registers
- 5. FS and GS: The FS and GS segments are supplemental segment registers available in the 80386-Core2 microprocessors to allow two additional memory segments for access by programs.

13. In the real mode, show the starting and ending addresses of each segment located by the following segment register values:

- (a) 1000H
- (b) 1234H
- (c) 2300H
- (d) E000H
- (e) ABOOH

(a) 1000H

starting address = 10000H ending address = 10000H +

FFFFH ending address = 1FFFFH

(b) 1234H

starting address = 12340H ending address = 12340H +

FFFFH ending address = 2233FH

(c) 2300H

starting address = 23000H ending address = 23000H +

FFFFH ending address = 32FFFH

(d) E000H

starting address = E0000H ending address = E0000H +

FFFFH ending address = EFFFFH

(e) AB00H

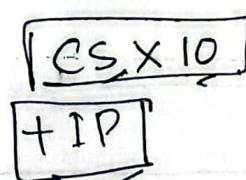
starting address = AB000H ending address = AB000H +

FFFFH ending address = BAFFFFH

(Starting  $\times 10$ )  
(FFFF + Starting)

14. Find the memory address of the next instruction executed by the microprocessor, when operated in the real mode, for the following CS:IP combinations:

(a) CS = 1000H and IP = 2000H



(b) CS = 2000H and IP = 1000H

(c) CS = 2300H and IP = 1A00H

(d) CS = 1A00H and IP = B000H

(e) CS = 3456H and IP = ABCDH

(a) CS = 1000H and IP = 2000H

CS : 1000H

IP : +2000H

Next instruction memory address: 12000H

(b) CS = 2000H and IP = 1000H

CS : 2000H

IP : +1000H

Next instruction memory address : 21000H

$CS = 2300H$  and  $IP = 1A00H$

$CS : 2300H$

$IP + 1A00H$

Next instruction memory address :  $24A000H$

(d)  $CS = 1A00H$  and  $IP = B000H$

$CS : 1A000H$

$IP : +B000H$

Next instruction memory address :  $25000H$

(e)  $CS = 3456H$  and  $IP = ABCDH$

$CS : 34560H$

$IP : +ABCDH$

Next instruction memory address :  $3F12DH$

15. Real mode memory addresses allow access to memory below which memory address?

$100000H$  1 MB or less

$1 MB or Less$

F-S

16. Which register or registers are used as an offset address for the string instruction destination in the microprocessor?

The Destination Index(DI) register is used as an offset address for the string instruction destination in the microprocessor. DI is a 16-bit register. It is used in the pointer addressing of data and as a destination in some string related operations.

17. Which 32-bit register or registers are used to hold an offset address for data segment data in the Pentium 4 microprocessor?

EAX, EBX, ECX, EDX, EBP, ESI, and EDI

18. The stack memory is addressed by a combination of the SS segment plus ESP / ESP offset.

The stack memory is addressed by a combination of the Stack segment plus SP or ESP offset.

19. If the base pointer (BP) addresses memory, the Stack segment contains the data.

Stack

20. Determine the memory location addressed by the following real mode 80286 register combinations: (e)  $SS = 2900H$  and  $SP = 3A00H$

(d)  $DS = A000H$  and  $BX = 1000H$

(c)  $SS = 2300H$  and  $BP = 3200H$

(b)  $DS = 2000H$  and  $SI = 1002H$

CS IP  
DS EAX, EBX, ECX, EDX,  
EBP, ESI, EDI.

(a) DS = 1000H and DI = 2000H

(a) DS = 1000H and DI = 2000H

Ans. DS = 10000H, DI = 2000H

Memory Location Accessed = DS + DI

$$= 10000H + 2000H$$

$$= 12000H$$

(b) DS = 2000H and SI = 1002H

Ans. DS = 20000H, DI = 1002H

Memory Location Accessed = DS + DI

$$= 20000 + 1002H$$

$$= 21002H$$

(c) SS = 2300H and BP = 3200H

Ans. DS = 23000H, DI = 3200H

Memory Location Accessed = DS + DI

$$= 23000 + 3200H$$

$$= 26200H$$

(d) DS = A000H and BX = 1000H

Ans. DS = A0000H, DI = 1000H

Memory Location Accessed = DS + DI

$$= A0000 + 1000H$$

$$= A1000H$$

(e) SS = 2900H and SP = 3A00H

Ans. DS = 29000H, DI = 3A00H

Memory Location Accessed = DS + DI

$$= 29000H + 3A00H$$

$$= 3F12DH$$

Determine the memory location addressed by the following real mode Core2 register combinations:

(e) DS = 1239H and EDX = 0000A900H

(d) SS = 8000H and ESP = 00009000H

(c) DS = C000H and ESI = 0000A000H

(b) DS = 1A00H and ECX = 00002000H

(a) DS = 2000H and EAX = 00003000H

**(a) DS = 2000H and EAX = 00003000H**

Ans. DS = 2000H, EAX = 00003000H

Memory Location Accessed = DS + EAX

$$= 2000H + 00003000H$$

$$= 23000H$$

**(b) DS = 1A00H and ECX = 00002000H**

Ans. DS = 1A00H, ECX = 00002000H

Memory Location Accessed = DS + ECX

$$= 1A00H + 00002000H$$

$$= 1C000H$$

**(c) DS = C000H and ESI = 0000A000H**

Ans. DS = C000H, ESI = 0000A000H

Memory Location Accessed = DS + ESI

$$= C000H + 0000A000H$$

$$= CA000H$$

**(d) SS = 8000H and ESP = 00009000H**

Ans. DS = 8000H, ESP = 00009000H

Memory Location Accessed = DS + ESP

$$= 8000H + 00009000H$$

$$= 89000H$$

**(e) DS = 1239H and EDX = 0000A900H**

Ans. DS = 1239H, EDX = 0000A900H

**Memory Location Accessed = DS + EDX**

$$= 12390H + 0000A9000H$$

$$= 1CC90H$$

**22.** Protected mode memory addressing allows access to which area of the memory in the 80286 microprocessor?

All 16M bytes / Below 1 MB and above 1 MB.

**23.** Protected mode memory addressing allows access to which area of the memory in the Pentium 4 microprocessor?

Any location in the memory system / 64 TeraBytes.

**24.** What is the purpose of the segment register in protected mode memory addressing?

The segment register is a selector that selects the descriptor from a descriptor table. It also sets privilege level of the request and chooses either the global or local table.

**25.** How many descriptors are accessible in the global descriptor table in the protected mode?

8,192

**26.** For an 80286 descriptor that contains a base address of A00000H and a limit of 1000H, what starting and ending locations are addressed by this descriptor?

A00000H-A01000H /

Base address = A00000H , Limit = 1000H

Starting Address = A000000H

Ending Address = Base Address + Limit

$$= \underline{\text{A00000H}} + 1000H$$

$$= \underline{\text{A01000H}}$$

Knowledge level  
of the user

**27.** For a Core2 descriptor that contains a base address of 01000000H, a limit of OFFFFH, and G = 0, what starting and ending locations are addressed by this descriptor?

01000000H-0100FFFFH /

Starting address = 01000000H

Since, G = 0, we will limit will be the same.

Ending address = 01000000H + OFFFFH

$$= \underline{\text{0100FFFFH}}$$

**28.** For a Core2 descriptor that contains a base address of 00280000H, a limit of 00010H, and G = 1, what starting and ending locations are addressed by this descriptor?

00280000H-00290FFFH /

Starting address = 00280000H

(G<sub>v</sub> = 1 means, limit is 4K bytes (append FFF to the limit))

Since, G = 1, value of the limit is multiplied 4K bytes (append FFF to the limit)

27828239

DS = 00010FFFH

Ending address = 00280000H + 00010FFFH

= 00290FFFH

29. If the DS register contains 0020H in a protected mode system, which global descriptor table entry is accessed?

4 /

In the DS register,

DS = 0020H

DS = 0000 0000 0010 0000H

So, TI = 0 4 the entry of global descriptor table will be accessed.

30. If DS = 0103H in a protected mode system, the requested privilege level is \_\_\_\_\_.

3 /

0103 = 0000 0001 0000 0011

RPL = 11

31. If DS = 0105H in a protected mode system, which entry, table, and requested privilege level are selected? Descriptor 20H, local table, a privilege ring 1 / 0105H = 0000 0001 0000 0101b.

From this, we can say that the requested privilege level is high(01), table

is local table, entry is 0 0000 0010 0000b = 20H

32th entry

32. What is the maximum length of the global descriptor table in the Pentium 4 microprocessor?

64K /

No of entries in the global descriptor table is 8192 Each descriptor is 8 bytes long,

so max length of the global descriptor table = 8192 \* 8

= 64K bytes

33. Code a descriptor that describes a memory segment that begins at location 210000H and ends at location 21001FH. This memory segment is a code segment that can be read. The descriptor is for an 80286 microprocessor.

Ans. Starting = 210000H Ending = 21001FH

Base address = 210000 0H

$$\begin{aligned} \text{Limit} &= 21001FH - 210000H \\ &= 001FH \end{aligned}$$

0000 0000 0000 0000		21H
0000H	001FH	

Access Right ->

P	D	P L	S	E	ED/C	R/W	A
1	1	1	1	1	1	1	0

We assume DPL 11 and ED/C = 1, we assume that segment not accessed yet since code is read R =1 and memory segment is in code segment E =1 and S =1

Therefore Access right = FEH

0000 0000 0000 0000	FE	21H
0000H	001FH	

34. Code a descriptor that describes a memory segment that begins at location 03000000H and ends at location 05FFFFFFH. This memory segment is a data segment that grows upward in the memory system and can be written. The descriptor is for a Pentium 4 microprocessor.

34.

0000 0011	1101 0000
1001 0010	0000 0000
0000 0000	0000 0000
0010 1111	1111 1111

/

G	D	0	AV	Limit
03H	1	1	0	1 FFF F2H 00H
0000H			2FFFH	

Assumed that D=1(32bit), AV =1(assumed that segment is available)

Access Right ->

P	D	P L	S	E	ED/C	R/W	A
1	1	1	0	0	1	0	—

0000 0000 0000 0000		21H
0000H		001FH

GDTR

S

35. Which register locates the global descriptor table?

GDTR / The GDTR (global descriptor table register) contain the base address of the global descriptor table and its limit. When the protected mode operation is desired, the address of the global descriptor table and its limit are loaded into the GDTR.

36. How is the local descriptor table addressed in the memory system?

Through a descriptor stored in the global table

37. Describe what happens when a new number is loaded into a segment register when the microprocessor is operated in the protected mode.

boa

The internal cache is loaded with the base address, offset address, and access rights byte

38. What are the program-invisible registers?

The program invisible registers are the cache portions of the segment registers and also the GDTR, LDTR, and IDTR registers.

[ALI]

39. What is the purpose of the GDTR?

The GDTR address the Global Descriptor Table

40. How many bytes are found in a memory page?

4Kilo

41. What register is used to enable the paging mechanism in the 80386, 80486, Pentium, Pentium Pro, Pentium 4, and Core2 microprocessors?

4096 CR0

$$\begin{array}{r} 1024 \\ \times 4 \\ \hline 4096 \end{array}$$

42. How many 32-bit addresses are stored in the page directory?

1024 PD

43. Each entry in the page directory translates how much linear memory into physical memory?

4M

44. If the microprocessor sends linear address 00200000H to the paging mechanism, which paging directory entry is accessed, and which page table entry is accessed?

Entry zero or the first entry

45. What value is placed in the page table to redirect linear address 20000000H to physical address 30000000H?

30000000H

46. What is the purpose of the TLB located within the Pentium class microprocessor?

The TLB caches the most recent memory accesses through the paging mechanism.

47. Using the Internet, write a short report that details the TLB. Hint:

You might want to go to the Intel Web site and search for information.

48. Locate articles about paging on the Internet and write a report detailing how paging is used in a variety of systems.

49. What is the flat mode memory system?

The flat mode memory system is used with 64-bit operation of the Core2

50. A flat mode memory system in the current version of the 64-bit Pentium 4 and Core2 allow these microprocessors to access \_\_\_\_\_ bytes of memory

1Tera

## Chapter-03: Addressing modes.

1. What do the following MOV instructions accomplish?

- (a) MOV AX,BX
- (b) MOV BX,AX
- (c) MOV BL,CH
- (d) MOV ESP,EBP
- (e) MOV RAX,RCX

- (a) the contents of BX is copied into AX
- (b) The contents of AX are copied into BX
- (c) the contents of CH are copied into BL
- (d) the contents of EBP are copied into ESP
- (e) the contents of RCX are copied into RAX

2. List the 8-bit registers that are used for register addressing.

AL, AH, BL, BH, CL, CH, DL, and DH

3. List the 16-bit registers that are used for register addressing.

Exam

AX, BX, CX, DX, SP, BP, SI, DI, CS, DS, ES, SS, FS, and GS

4. List the 32-bit registers that are used for register addressing in the 80386 through the Core2 microprocessors. EAX, EBX, ECX, EDX, ESP, EBP, EDI, and ESI

5. List the 64-bit registers available to the 64-bit mode of the Pentium 4 and Core2.

RAX, RBX, RCX, RDX, RSP, RBP, RSI, RDI and R8—R15

6. List the 16-bit segment registers used with register addressing by MOV, PUSH, and POP.

CS, DS, ES, SS, FS, and GS

7. What is wrong with the MOV BL,CX instruction?

BL is of 8bits and CX is of 16 bits, The register sizes must be equal, 16bit cannot be fit into 8-bits.

8. What is wrong with the MOV DS,SS instruction?

You may not specify mixed register sizes. because segment to segment transfer is not allowed

9. Select an instruction for each of the following tasks:

- (a) copy EBX into EDX
- (b) copy BL into CL

- (c) copy SI into BX
  - (d) copy DS into AX
  - (e) copy AL into AH
  - (f) copy R8 into R10
- (a) MOV EDX,EBX
  - (b) MOV CL,BL
  - (c) MOV BX,SI
  - (d) MOV AX,DS
  - (e) MOV AH,AL
- (f) MOV R10,R8

10. Select an instruction for each of the following tasks:

(a) move 12H into AL

(b) move 123AH into AX

(c) move OCDH into CL

(d) move 1000H into RAX

(e) move 1200A2H into EBX

(a) MOV AL,12H

(b) MOV AX,123AH

(c) MOV CL,OCDH

(d) MOV RAX,1000H

(e) MOV EBX,1200A2H

11. What special symbol is sometimes used to denote immediate data?

#

12. What is the purpose of the .MODEL TINY statement?

Selects an assembly language programming model that contains a single segment that compiles as a .COM program.

13. What assembly language directive indicates the start of the CODE segment?

.CODE

14. What is a label?

A label is a symbolic memory address.

15. The MOV instruction is placed in what field of a statement?

Opcode

16. A label may begin with what characters?

A label may begin with a letter and some special characters, but not with a number.

17. What is the purpose of the .EXIT directive?

It ends the program by exiting to the operating system

18. Does the .MODEL TINY statement cause a program to assemble as an execute (.EXE) program?

The .TINY model creates a .COM program

19. What tasks does the .STARTUP directive accomplish in the small memory model?

The .STARTUP directive loads the DS register

20. What is a displacement? How does it determine the memory address in a MOV DS:[2000H],AL instruction?

A displacement is a distance and in MOV DS:[2000H],AL the displacement of 2000H is added to the contents of DS times 10H to form the memory address.

21. What do the symbols [ ] indicate?

Indirect addressing

22. Suppose that DS = 0200H, BX = 0300H, and DI = 400H. Determine the memory address accessed by each of the following instructions, assuming real mode operation:

(a) MOV AL,[1234H]

(b) MOV EAX,[BX]

(c) MOV [DI],AL

$$(a) DS \times 10 + 1234H = 2000H + 1234H = 3234H$$

$$(b) DS \times 10 + 300H = 2000H + 300H = 2300H$$

$$(c) DS \times 10 + 400H = 2000H + 400H = 2400H$$

23. What is wrong with a MOV [BX],[DI] instruction?

Memory to memory transfers are not allowed with the MOV instruction

24. Choose an instruction that requires BYTE PTR.

MOV BYTE PTR [2000H],6

25. Choose an instruction that requires WORD PTR.

MIC WORD PTR [EDI]

27

26. Choose an instruction that requires DWORD PTR.

MOV DWORD PTR DATA1, 5

27. Select an instruction that requires QWORD PTR.

DEC QWORD PTR [RAX]

28. Explain the difference between the MOV BX,DATA instruction and the MOV BX,OFFSET DATA instruction.

The MOV BX,DATA instruction copies the word from memory location data into the BX register where the MOV BX,OFFSET DATA instruction copies the offset address of DATA into BX.

29. Suppose that DS = 1000H, SS = 2000H, BP = 1000H, and DI = 0100H. Determine the memory address accessed by each of the following instructions, assuming real mode operation:

(a) MOV AL,[BP+DI]

(b) MOV CX,[DI]

(c) MOV EDX,[BP]

(a)  $SS \times 10 + BP + DI = 20000H + 1000H + 100H = 21100H$

(b)  $DS \times 10 + 100H = 10000H + 100H = 10100H$

(c)  $SS \times 10 + BP = 20000H + 1000H = 21000H$

Handwritten notes:  
Data seg --  
Code seg IP  
BP  $\rightarrow$  (on SS)  
ES  
Ex seg = DI RTA, DS LTA  
SS = SP, BP

30. What, if anything, is wrong with a MOV AL,[BX][SI] instruction?

Nothing is wrong with the instruction; it just uses an alternative addressing style

31. Suppose that DS = 1200H, BX = 0100H, and SI = 0250H. Determine the address accessed by each of the following instructions, assuming real mode operation: (a) MOV [100H],DL

(b) MOV [SI+100H],EAX

(c) MOV DL,[BX+100H]

(a)  $DS \times 10 + 100H = 12000H + 100H = 12100H$

(b)  $DS \times 10 + SI + 100H = 12000H + 250H + 100H = 12350H$

(c)  $DS \times 10 + BX + 100H = 12000H + 100H + 100H = 12000H$

32. Suppose that DS = 1100H, BX = 0200H, LIST = 0250H, and SI = 0500H. Determine the address accessed by each of the following instructions, assuming real mode operation:

(a) MOV LIST[SI],EDX  $(DS \times 10) + LIST + SI$

(b) MOV CL,LIST[BX+SI]  $(DS \times 10) + LIST + BX + SI$

(c) MOV CH,[BX+SI]  $(DS \times 10) + BX + SI$

(a) 11750H (b) 11950H (c) 11700H

Suppose that DS = 1300H, SS = 1400H, BP = 1500H, and SI = 0100H. Determine the address accessed by each of the following instructions, assuming real mode operation: (a) MOV EAX,[BP+200H]

- (b) MOV AL,[BP+SI-200H]  
(c) MOV AL,[SI-0100H]  
(a)  $SS \times 10 + BP + 200H = 14000H + 1500H + 200H = 15700H$   
(b)  $SS \times 10 + BP + SI - 200H = 14000H + 1500H + 100H - 200H = 15400H$   
(c)  $DS \times 10 + SI - 200 = 13000H + 100H - 200H = 12F00H$

(BP 2170H SS 3C0H)

34. Which base register addresses data in the stack segment?

~~BP or as an extended version EBP~~

35. Suppose that EAX = 00001000H, EBX = 00002000H, and DS = 0010H. Determine the addresses accessed by the following instructions, assuming real mode operation:

- (a) MOV ECX,[EAX+EBX]  
(b) MOV [EAX+2\*EBX]  
(c) MOV DH,[EBX+4\*EAX+1000H]

(a) 3100H (b) 05100H (c) 07100H

36. Develop a data structure that has five fields of one word each named F1, F2, F3, F4, and F5 with a structure name of FIELDS.

FIELDS STRUC

- F1 DW ?  
F2 DW ?  
F3 DW ?  
F4 DW ?  
F5 DW ?

Fields  
F1  
F2  
F3  
F4  
F5  
Data  
Data  
Data  
Data  
Data

FIELDS ENDS

37. Show how field F3 of the data structure constructed in question 36 is addressed in a program.

38. What are the three program memory-addressing modes?

Direct, relative, and indirect

39. How many bytes of memory store a far direct jump instruction?

What is stored in each of the bytes?

5. the first byte is the opcode, followed by a two byte segment address, followed by a two byte offset address

- 1st - opcode
- 2nd byte - segment
- 3rd byte - offset

40. What is the difference between an intersegment and intrasegment jump?

The intersegment jump allows jumps between segments or to anywhere in the memory system while the intrasegment jump allows a jump to any location within the current code segment.

41. If a near jump uses a signed 16-bit displacement, how can it jump to any memory location within the current code segment?

32Kilo

42. The 80386 and above use a 32-bit displacement to jump to any location within the 4G-byte code segment.

32

43. What is a far jump?

A far jump always a jump to any location in the memory map

44. If a JMP instruction is stored at memory location 100H within the current code segment, it cannot be a \_\_\_\_\_ jump if it is jumping to memory location 200H within the current code segment.

Short

(100H, jump 200H)

45. Show which JMP instruction assembles (short, near, or far) if the JMP THERE instruction is stored at memory address 10000H and the address of THERE is:

Chyon

- (a) 10020H
- (b) 11000H ✓
- (c) OFFFEH
- (d) 30000H

a If the memory address of THERE is 10020H,

10020H - 65568

-10000H - 65536

00020H → 0000 0000 0000 0010 0000

1 byte

1 byte displacement → short jump

b If the memory address of THERE is 11000H,

11000H 69632

-10000H 65536

01000H → 0000 0001 0000 0000 0000

2 byte

2 byte displacement → near jump

- (c) short (d) far (short 1 byte, near 2 byte, far > near)

Form a JMP instruction that jumps to the address pointed to by the BX register.

JMP BX

47. Select a JMP instruction that jumps to the location stored in memory at the location TABLE.  
Assume that it is a near JMP.

JMP NEAR

48. How many bytes are stored on the stack by a PUSH AX? 2

49. Explain how the PUSH [DI] instruction functions.

PUSH [DI] places the 16-bit contents of the location addressed by DS and DI onto the stack

50. What registers are placed on the stack by the PUSHA instruction? In what order?

AX, CX, DX, BX, SP, BP, DI, and SI in the same order as listed

51. What does the PUSHAD instruction accomplish?

Places the 32-bit contents of the register array onto the stack

52. Which instruction places the EFLAGS on the stack in the Pentium 4 microprocessor?

PUSHFD

53. Is a PUSHA available in the 64-bit mode of the Pentium 4 or the Core2?

No

## Chapter-04: Data Movement Instructions.

1. The first byte of an instruction is the \_\_\_\_\_, unless it contains one of the override prefixes.

Opcode

2. Describe the purpose of the D- and W-bits found in some machine language instructions.

The D-bit indicates the direction of flow for the data (REG to R/M or R/M to REG) and the W-bit indicates the size of the data (byte or word/doubleword). (W = Data size)

3. In a machine language instruction, what information is specified by the MOD field?

The MOD field specifies the type of access for the R/M field and the size of the displacement

4. If the register field (REG) of an instruction contains 010 and W = 0, what register is selected, assuming that the instruction is a 16-bit mode instruction?

DL

5. How are the 32-bit registers selected for the Pentium 4 microprocessor?

If operated in the 16-bit mode, a register-size and/or address-size prefix is used to specify a 32-bit register

6. What memory-addressing mode is specified by R/M = 001 with MOD = 00 for a 16-bit instruction?

DS:[BX+DI]

register size  
addr size prefix  
R/M field  
size of disp  
displacement

Ch 10

Identify the default segment registers assigned to the following:

- (a) SP
- (b) EBX
- (c) DI
- (d) EBP
- (e) SI

DS  
DS  
SS  
DS  
DS  
SS  
SS

(a) SS (b) DS (c) DS (d) SS (e) DS

8. Convert an 8B07H from machine language to assembly language.

MOV AX,[BX]

9. Convert an 8B9E004CH from machine language to assembly language.

MOV BX,[BP+4C00H]

10. If a MOV SI,[BX+2] instruction appears in a program, what is its machine language equivalent?

8B 77 02

11. If a MOV ESI,[EAX] instruction appears in a program for the Core2 microprocessor operating in the 16-bit instruction mode, what is its machine language equivalent?

67 66 8B 30

67 66 8B 30

- ✓ 67: Address-size override prefix (32-bit address size in a 16-bit code segment).
- ✓ 66: Operand-size override prefix (32-bit operand size).
- ✓ 8B: Opcode for the MOV instruction.
- ✓ 30: Mod R/M byte, where:

3 is the code for the ESI register (destination operand).

0 is the code for the EAX register (source operand).

11 (Alternative). If a MOV ESI,[EAX] instruction appears in a program for the Core2 microprocessor operating in the 32-bit instruction mode, what is its machine language equivalent?

8B 30

12. What is the purpose of REX?

The REX prefix, which is used in the 64-bit flat mode, is the register extension that allows the 64-bit registers to be addressed in an instruction.

13. What is wrong with a MOV CS, AX instruction?

Me~

The contents of CS will change causing an unpredictable jump. You should never change CS without also changing IP. This instruction would most likely cause the system to crash because only the segment portion of the address of the next instruction is changed.

14. Form a short sequence of instructions that load the data segment register with a 1000H.

MOV AX,1000H

MOV DS,AX

DS

The PUSH and POP instructions always transfer a(n) \_\_\_\_\_ -bit number between the stack and a register or memory location in the 80386-Core2 microprocessors when operated in the 32-bit mode.

32)

16. Create an instruction that places RAX onto the stack in the 64-bit mode for the Pentium 4.

PUSH RAX

17. What segment register may not be popped from the stack?

CS

18. Which registers move onto the stack with the PUSHA instruction?

AX, CX, DX, BX, SP, BP, SI, and DI

19. Which registers move onto the stack for a PUSHAD instruction? EAX, EBX, ECX, EDX, ESP, EBP, EDI and ESI

20. Describe the operation of each of the following instructions:

(a) PUSH AX

(b) POP ESI

(c) PUSH [BX]

(d) PUSHFD

(e) POP DS

(f) PUSHD 4

(a) AX is copied to the stack.

2

(b) A 32-bit number is retrieved from the stack and placed into ESI.

(c) The word contents of the data segment memory location addressed by BX is pushed onto the stack.

(d) EFLAGS are pushed onto the stack.

(e) A word is retrieved from the stack and placed into DS.

(f) A 32-bit number 4 is pushed onto the stack.

21. Explain what happens when the PUSH BX instruction executes.

Make sure to show where BH and BL are stored. (Assume that SP = 0100H and SS = 0200H.)

The BH register is moved to memory location 020FFH and the BL register is moved to location 020FEH then SP is changed to 00FEH.

22. Repeat question 21 for the PUSH EAX instruction.

Bits 24–31 of EAX are stored in location 020FFH, bits 16–23 of EAX are stored into location 020FEH, bits 8–15 of EAX are stored into location 020FDH, and bits 0–7 of EAX are stored into location 020FCH. SP is then decremented by 4 to a value of 00FCH.

23. The 16-bit POP instruction (except for POPA) increments SP by \_\_\_\_\_.

2

24. What values appear in SP and SS if the stack is addressed at memory location 02200H?

There are many possible locations, but SP = 0200H and SS = 0200H is one of them.

25. Compare the operation of a MOV DI,NUMB instruction with an LEA DI,NUMB instruction.

The MOV DI,NUMB instruction copies the 16-bit number in the data segment location NUMB into DI while the LEA DI,NUMB loads DI with the offset address of location NUMB.

26. What is the difference between an LEA SI,NUMB instruction and a MOV SI,OFFSET NUMB instruction?

Both instruction load the address of NUMB into DI. The difference is that the MOV DI,OFFSET NUMB assembles as a move immediate and the LEA DI,NUMB assembles as an LEA instruction.

27. Which is more efficient, a MOV with an OFFSET or an LEA instruction?

The MOV with the OFFSET directive

28. Describe how the LDS BX,NUMB instruction operates.

The LDS BX,NUMB instruction loads BX with the word stored at data segment memory location NUMB and DS is loaded from the data segment memory location addressed by NUMB+2.

29. What is the difference between the LDS and LSS instructions?

LDS loads DS and LSS loads SS along with another 16-bit register for the offset address

30. Develop a sequence of instructions that moves the contents of data segment memory locations NUMB and NUMB+1 into BX, DX, and SI.

MOV BX,NUMB

MOV DX,BX

MOV SI,DX

31. What is the purpose of the direction flag?

If the direction flag is cleared it selects auto-increment for the string instructions and if the direction flag is set it selects auto-decrement.

32. Which instructions set and clear the direction flag?

CLD clears the direction flag and STD sets the direction flag.

Set → -1 → STD

CLD → +1 → Clear

33. Which string instruction(s) use both DI and SI to address memory data?

MOVS.

4. Explain the operation of the LODSB instruction.

The LODSB instruction copies a byte of data from the data segment memory location addressed by SI into the AL register and then increments SI by one if the direction flag is cleared.

35. Explain the operation of the LODSQ instruction for the 64-bit mode of the Pentium 4 or Core2.

A 4-bit number is loaded into RAZ from the data segment memory location addressed by ESI and then ESI is either incremented or decremented by 8 depending on the setting of the direction flag.

36. Explain the operation of the OUTSB instruction.

The OUTSB instruction sends the contents of the data segment memory location addressed by SI to the I/O port address by DX, then SI is incremented by one if the direction flag is cleared.

37. Explain the operation of the STOSW instruction.

The STOSW instruction copies AX into the extra segment memory location addressed by DI then DI is either incremented or decremented by two as dictated by the direction flag.

38. Develop a sequence of instructions that copy 12 bytes of data from an area of memory addressed by SOURCE into an area of memory addressed by DEST.

MOV SI,OFFSET SOURCE

MOV DI,OFFSET DEST

MOV CX,12

REP MOVS

39. What does the REP prefix accomplish and what type of instruction is it used with?

The REP prefix repeats a string instruction CX number of times.)

40. Select an assembly language instruction that exchanges the contents of the EBX register with the ESI register.

XCHG EBX,ESI

41. Where is the I/O address (port number) stored for an INSB instruction?

DX register

42. Would the LAHF and SAHF instructions normally appear in software?

The LAHF and SAHF instructions in non-64-bit application with the arithmetic coprocessor.

43. Write a short program that uses the XLAT instruction to convert the BCD numbers 0-9 into ASCII-coded numbers 30H-39H. Store the ASCII-coded data in a TABLE located within the data segment.

43.

TABLE	DB	30H, 31H, 32H, 33H
	DB	34H, 35H, 36H, 37H, 38H, 39H
BCD2A	PROC	NEAR
	MOV	BX,OFFSET TABLE
	XLAT	
	RET	
BCD2A	ENDP	

**44.** Explain how the XLAT instruction transforms the contents of the AL register.

The XLAT instruction passes the contents of AL to BX to form an offset address that accesses a memory location whose content is then copied into AL.

**45.** Explain what the IN AL,12H instruction accomplishes. IN AL, 12H copies the byte from I/O device 12H into AL

**46.** Explain how the OUT DX,AX instruction operates.

The OUT DX,AX instruction copies the 16-bit contents of AX into the data segment memory location addressed by the DX register.

**47.** What is a segment override prefix?

The segment override prefix allows the default segment to be changed to any segment

**48.** Select an instruction that moves a byte of data from the memory location addressed by the BX register in the extra segment into the AH register.

~~MOV AH,ES:[BX]~~

**49.** Develop a sequence of instructions that exchanges the contents of AX with BX, ECX with EDX, and SI with DI.

**49.**

XCHG	AX, BX
XCHG	ECX, EDX
XCHG	SI, DI

**50.** What is an assembly language directive?

→ [Code generate ?? ??]

An assembly language directive is a special command to the assembler that may or may not generate code or data for the memory

**51.** What is accomplished by the CMOVNE CX,DX instruction in the Pentium 4 microprocessor?

DX is copied into CX if a not zero or not equal condition exists.

**52.** Describe the purpose of the following assembly language directives: DB, DW, and DD.

The directives, DB, DW, and DD, are used to define memory as a byte (DB), a word (DW), and a doubleword (DD).

**53.** Select an assembly language directive that reserves 30 bytes of memory for array LIST1.

~~LIST1 DB 30 dup(?)~~

**54.** Describe the purpose of the EQU directive.

The EQU (equate) directive allows a memory location to be equated to another memory location.

**55.** What is the purpose of the .686 directive?

The .686 directive informs the assembler that a Pentium Pro or newer microprocessor is the target of the assembled program.

**56.** What is the purpose of the .MODEL directive?

The .MODEL directive specifies the type of memory model used for a program.

- If the start of a segment is identified with .DATA, what type of memory organization is in effect?
- models
58. If the SEGMENT directive identifies the start of a segment, what type of memory organization is in effect?

Full segment definitions

59. What does the INT 21H accomplish if AH contains a 4CH?

The program terminates and control is passed back to the operating system.

60. What directives indicate the start and end of a procedure?

PROC indicates the start of a procedure and ENDP indicates its end.

61. Explain the purpose of the USES statement as it applies to a procedure with version 6.x of MASM.

The uses directive specifies which registers are saved on the stack at the beginning of a procedure and popped at the end of the procedure.

62. Develop a near procedure that stores AL in four consecutive memory locations within the data segment, as addressed by the DI register.

```
62. STORE PROC NEAR
    MOV [DI], AL
    MOV [DI+1], AL
    MOV [DI+2], AL
    MOV [DI+3], AL
    RET
STORE ENDP
```

63. How is the Pentium 4 microprocessor instructed to use the 16-bit instruction mode?

If the model statement precedes the processor directive the code generated is 16-bit.

64. Develop a far procedure that copies contents of the word-sized memory location CS:DATA4 into AX, BX, CX, DX, and SI.

```
64. COPY PROC FAR
    MOV AX, CS:DATA4
    MOV BX, AX
    MOV CX, AX
    MOV DX, AX
    MOV SI, AX
    RET
COPY ENDP
```

## Chapter-05: Arithmetic and Logic instructions.

[Click here and read from this site.](#)

## Chapter-06: Program Control Instructions.

1. What is a short JMP?

A short jump allows a program to branch forward 127 bytes or backwards 128 bytes from the next instruction's address in the program.

2. Which type of JMP is used when jumping to any location within the current code segment?

A near JMP instruction

3. Which JMP instruction allows the program to continue execution at any memory location in the system?

Far jump

4. Which JMP instruction is 5 bytes long?

A far jump

5. What is the range of a near jump in the 80386–Core2 microprocessors?

±2G

6. Which type of JMP instruction (short, near, or far) assembles for the following:

(a) if the distance is 0210H bytes

(b) if the distance is 0020H bytes

(c) if the distance is 10000H bytes

(a) near (b) short (c) far

7. What can be said about a label that is followed by a colon?

A label followed by a single colon is a short or near address and a double colon denotes a far address.

8. The near jump modifies the program address by changing which register or registers?

The IP or EIP register

9. The far jump modifies the program address by changing which register or registers?

The code segment register and the instruction address register

10. Explain what the JMP AX instruction accomplishes. Also identify it as a near or a far jump instruction.

The JMP AX instruction jumps to the offset address stored in AX. This can only be a near jump

11. Contrast the operation of a JMP DI with a JMP [DI].

A JMP DI copies the contents of DI into the instruction address register and a JMP [DI] copies the 16-bit number from the data segment memory location addressed by DI into the instruction address register.

12. Contrast the operation of a JMP [DI] with a JMP FAR PTR [DI].

The JMP [DI] instruction jumps to the memory location addressed by the offset address stored in the data segment memory location addressed by DI. The JMP FAR PTR[DI] instruction jumps to the new offset address stored in the data

segment memory location addressed by DI and the new segment addressed by the data segment memory location address by DI+2. JMP [DI] is a near jump and JMP FAR PTR [DI] is a far jump.

13. List the five flag bits tested by the conditional jump instructions.

Sign (S), Zero (Z), Carry (C), Overflow (O), and Parity (P)

14. Describe how the JA instruction operates.

JA tests the condition of an arithmetic or logic instruction to determine if the outcome is above. If the outcome is above a jump occurs, otherwise no jump occurs.

15. When will the JO instruction jump?

A JO instruction jumps on an overflow condition

16. Which conditional jump instructions follow the comparison of signed numbers?

JNE, JE, JG, JGE, JL, or JLE

17. Which conditional jump instructions follow the comparison of unsigned numbers?

JNZ, JNE, JZ, JE, JB, JBE, JA, JAE

18. Which conditional jump instructions test both the Z and C flag bits?

JA and JBE

19. When does the JCXZ instruction jump?

Tests the contents of CX and jumps if it is zero

20. Which SET instruction is used to set AL if the flag bits indicate a zero condition?

SETZ or SETE       $0 \rightarrow Z \text{ or } E$

21. The 8086 LOOP instruction decrements register \_\_\_\_\_ and tests it for a 0 to decide if a jump occurs.

CX

22. The Pentium 4 LOOPD instruction decrements register \_\_\_\_\_ and tests it for a 0 to decide if a jump occurs.

ECX

23. The Core2 operated in 64-bit mode for a LOOP instruction decrements register \_\_\_\_\_ and tests it for a 0 to decide if a jump occurs.

RCX

24. Develop a short sequence of instructions that stores 00H into 150H bytes of memory, beginning at extra segment memory location DATAZ. You must use the LOOP instruction to help perform this task.

**MOV DI, OFFSET DATAZ**

**MOV CX, 150H**

**CLD**

**MOV AL, 00H**

**L1: STOSB**

**LOOP L1**

25. Explain how the LOOPE instruction operates.

The LOOPE instruction jumps if an equal condition exists and CX is not zero and it also decrements CX on each iteration of the loop.

26. Show the assembly language instructions are generated by the following sequence:

**CMP AL, 3**

**JNE @C0001**

**ADD AL, 2**

**@C0001:**

27. Develop a sequence of instructions that searches through a block of 100H bytes of memory. This program must count all the unsigned numbers that are above 42H and all that are below 42H. Byte-sized data segment memory location UP must contain the count of numbers above 42H, and data segment location DOWN must contain the count of numbers below 42H.

```
MOV SI, OFFSET BLOCK
MOV UP, 0
MOV DOWN, 0
MOV CX, 100H
MOV AL, 42H
CLD
L1: SCASB
JE L3
JA L2
INC DOWN
JMP L3
L2: INC UP
L3: LOOP L1
```

28. Develop a short sequence of instructions that uses the REPEAT-UNTIL construct to copy the contents of byte-sized memory **BLOCKA** into byte-sized memory **BLOCKB** until 00H is moved.

**MOV SI, OFFSET BLOCKA**

**MOV DI, OFFSET BLOCKB**

**CLD**

**.REPEAT**

LODSB

STOSB

.UNTIL AL == 0

29. What happens if the WHILE 1 instruction is placed in a program?

An infinite loop is created.

30. Using the WHILE construct, develop a sequence of instructions that add the byte-sized contents of BLOCKA to BLOCKB while the sum is not 12H.

```
MOV AL, 0
MOV SI, OFFSET BLOCKA
MOV DI, OFFSET BLOCKB
CLD
.WHILE AL != 12H
    LODSB
    ADD AL, [DI]
    MOV [DI], AL
    INC DI
.ENDW
```

31. What is the purpose of the BREAK directive?

A BREAK can be used to break out of a .WHILE construct.

32. What is a procedure?

A procedure is a reusable group of instructions that ends with a RET.

33. Explain how the near and far CALL instructions function.

The main difference between a near and a far call is the distance from the call and the type of call and return that assembles.

34. The last executable instruction in a procedure must be a(n) \_\_\_\_\_.

RET

35. How does the near RET instruction function?

The near return <sup>RET</sup> retrieves the return address from the stack and places it into the instruction address register.

36. How is a procedure identified as near or far?

By using NEAR or FAR to the right of the PROC directive.

37. Which directive identifies the start of a procedure?

PROC

38. Write a near procedure that cubes the contents of the CX register. This procedure may not affect any register except CX.

Near RET picks the return address from stack  
and place it to the instruction address register.

```
CUBE PROC NEAR USES AX DX
    MOV AX, CX
    MUL CX
    MUL CX
    RET
CUBE ENDP
```

39. Explain what the RET 6 instruction accomplishes.

The RET 6 deletes 6 bytes from the stack before returning from a procedure.

40. Write a procedure that multiplies DI by SI and then divides the result by 100H. Make sure that the result is left in AX upon returning from the procedure. This procedure may not change any register except AX.

```
SUMS PROC NEAR
    MOV EDI, 0
    ADD EAX, EBX
    ADD EAX, ECX
    ADD EAX, EDX
    ADC EDI, 0
    RET
SUMS ENDP
```

41. Write a procedure that sums EAX, EBX, ECX, and EDX. If a carry occurs, place a logic 1 in EDI. If no carry occurs, place a 0 in EDI. The sum should be found in EAX after the execution of your procedure.

```
SUMS PROC NEAR
    MOV EDI, 0
    ADD EAX, EBX
    JNC SUMA1
    MOV EDI, 1
SUMA1: ADD EAX, ECX
    JNC SUMS2
    MOV EDI, 1
SUMS2: ADD EAX, EDX
    JNC SUM3
    MOV EDI, 1
SUM3:
SUMS ENDP
```

42. What is an interrupt?

An interrupt is a hardware-initiated function call.

43. Which software instructions call an interrupt service procedure?

INT

44. How many different interrupt types are available in the microprocessor?

INT 0 through INT 255

45. Illustrate the contents of an interrupt vector and explain the purpose of each part.

An interrupt vector contains the offset address followed by the segment address in 4 bytes of memory.

46. What is the purpose of interrupt vector type number 0?

The interrupt vector is used to detect and respond to divide errors.