SOLUTIONS FOR CHAPTER 14 (6)

SECTION 14.1 (6.1): 8088/86 INTERRUPTS

1. Yes. It finishes the current instruction and saves the address of the next instruction on the stack before it gets the CS:IP of the interrupt service routine. The stack provides the returning address.
2. (a) For INT 05 we have logical address = 0000:0014H and physical address = 00014H.
   (b) For INT 21H, we have LA = 0000:0084H and PA = 00084H.
3. ISR stands for *interrupt service routine*, and is the program associated with the INT type. When an interrupt is invoked, it is asked to run the ISR. Another name for ISR is *interrupt handler*.
4. the interrupt vector table
5. (a) In the CALL FAR, only 4 bytes of the stack is used, 2 for the IP and 2 for the CS of the next instruction.
   (b) In the interrupt, 6 bytes are used, 2 for the flag register, 2 for IP and 2 for the CS of the next instruction.
6. INT FEH or INT 254
7. The logical address is 0000:0000 to 0000:03FFH and physical address is 00000 to 003FFH.
8. 1024 bytes, since there are 256 interrupts and each interrupt takes 4 bytes.
9. because it is set aside for the interrupt vector table
10. INT 00 for the divide error or quotient oversize
11. (a) INT 00 (b) INT 01 (c) INT 02
12. true
13. false, only the INTR
14.

```
        PUSHF
        MOV     BP,SP
        OR      BP+0,0000 0001 0000 0000B     ;binary position of TF
        POPF
```

15. (a) The IF is set to low by instruction CLI.
    (b) It is set to high by instruction STI.
16. true
17. IRET, RETF
18. In RETF, only the 4 bytes are popped from the stack, while IRET pops off 6 bytes FR,CS, and IP.
19.

	for Interrupt	for CALL FAR
SS:FFDA	IP (low)	
SS:FFDB	IP (high)	
SS:FFDC	CS (low)	IP (low)
SS:FFDD	CS (high)	IP (high)
SS:FFDE	FR (low)	CS (low)
SS:FFDF	FR (high)	CS (high)
SS:FFE0		

20. none; the sequence is IP, CS and FR

SECTION 14.2 (6.2): IBM PC AND MS DOS INTERRUPT ASSIGNMENT

21. (a) This belongs to INT 07
    (b) F000:FF47H is the logical address and FFF47H is the physical address.
22. BIOS
23. c
24. CS = F000H and IP = FE6EH
25. 0000:0070 = 6EH, 0000:0071 = FEH, 0000:0072 = 00H, 0000:0073 =F0H
26. locations 0000:0413H and 0000:0414H

SECTION 14.3 (6.3): 8259 PROGRAMMABLE INTERRUPT CONTROLLER

27. false, A0 = 0
28. All are input, except INT.
29. The port address for ICW1 is 94H while ICW2, ICW3, and ICW4 all have the port address of 95H.
30. 00010110 = 16H for ICW1. The ICW2 is always the INT # assigned to IR0; therefore, the ICW2=88H.
31.

```
        MOV    AL,16H         ;no ICW4,SNGL,x86,EDGE TRIG
        OUT    94H,AL
        MOV    AL,88H         ;INT 88h is assigned to IR0
        OUT    95H,AL         ;output the ICW2
```

32. a, c, d, since the lower nibble must be 8 or 0. The 8259 provides the 3 bit combinations for IR0 - IR7.
33. IR0 will have the INT 18H, and IR7 the INT 1FH.
34. IR0 has 30H, IR4 has INT 34H, and IR6 has INT 36H
35. OCW1
36. OCW2
37. IR0 has the highest priority, and IR7 the lowest priority.
38. OCW1 has the port address of 95H, and OCW2 and OCW3 both have the same port address of 94H.
39.

```
        MOV    AL,11101011    ;OCW1: MASK ALL EXCEPT IR2&IR4
        OUT    95H,AL         ;SEND THE OCW1
```

40. Notice that D4 = 1 in ICW1, while in OCW2 and OCW3, D4 = 0. In OCW2 we have D3 = 0, while in OCW3, D3 = 1

SECTION 14.4 (6.4): USE OF THE 8259 CHIP IN THE PBM PC/XT

41. because the 8259 must be programmed by the main CPU the 80x86, and not the DMA; this means that AEN = 0 and must be used in the address decoding circuitry to access the 8259 chip.
42. single
43. edge
44. INT 08 to INT 0FH
45. port addresses 20H and 21H
46. true
47. IRQ2 to IRQ7
48. side B
49. 0010 0000 = 20H and is sent to port address 20H for OCW2
50. IRQ0 since BIOS programs the 8259 in the default fully nested mode
51. true
52. true
53. Yes, through D7 of the port address A0H.
54. true

SECTION 14.5 (6.5): INTERRUPTS ON 80286 AND HIGHER 80x86 PCs

55. true
56. true
57. The primary (master) has 20H and 21H for the port address. The secondary (slave) has A0H and A1H as port addresses.
58. INT 70H to 77H
59. IRQ3 to IRQ7; IRQ9, IRQ10, IRQ11, IRQ12, IRQ14, IRQ15
60. The IRQ2 is used for the cascade connection of the 2nd 8259; therefore, it is not available at the expansion slot. IRQ9 is provided at the pin where IRQ2 used to be in the PC/XT. Internally, BIOS redirects IRQ9 to IRQ2.

61. (a) The port address where the EOI is issued for the primary 8259 is 20H, and for the secondary 8259 is A0H.
(b) If any of the IRQ0, IRQ1, IRQ3 - IRQ7 is activated, the EOI is issued only to the primary 8259. However, if any of IRQ8 - IRQ15 from the secondary 8259 are activated not only must the EOI be issued to the secondary, but also to the primary 8259 since these IRQs come in through IRQ2 of the primary (master) 8259.
62. true
63. false; they have higher priority
64. true
65. IRQ10 and IRQ6 both come to the CPU through the INTR; therefore, they have lower priority than NMI. It means that the CPU takes care of the NMI first. Then IRQ10 is serviced, and finally IRQ6 is serviced.
66. IRQ15 is serviced first, then IRQ3, and finally IRQ7. At the end of the interrupt service routine for the IRQ15, the issuing of the EOI will allow the lower priority IRQs to be responded to; therefore, the IRQ3 is serviced next. Again, at the end of the ISR for IRQ3, the issuing of the EOI allows the system to service IRQ7.
67. lower
68. When the IRQ is responded to via INTR, the CPU makes IF = 0, meaning that no more interrupts until the CPU finishes the ISR belonging the current IRQ. The STI instruction at the beginning of the ISR makes IF =1, thereby allowing a higher priority IRQ to interrupt the current ISR. This means that you can have an interrupt inside an interrupt, or nested interrupts.
69. This means that there is no interrupting of the present ISR by any IRQs through the INTR pin. In other words, no nesting of interrupts from the INTR pin.
70. 8259
71. true
72. a low-to-high (positive edge trigger) pulse
73. IRQ10 has the higher priority since it is cascaded to IRQ2 of the master 8259 interrupt controller.
74. Level-triggered interrupts allow the sharing of a single interrupt line among several devices. They are also more immune to noise than edge-triggered interrupts.
75. As the number of devices using interrupts increases, the system is more likely to run out of interrupt lines. Therefore, using interrupt sharing allows several devices to share a single interrupt. Then it is the job of BIOS to distinguish which device initiated the interrupt. The PC buses EISA, MCA, and PCI are equipped with interrupt sharing, but not ISA.