Solution to Questions in section 5.2 ➔ Problems11-25

11. 110011000_2, 198H, 408_{10}.
12. (a) 00000011_2. **(b)** 10001101_2.
13. (a) 00001001_2. **(b)** 01101001_2.
14. 00001111_2, 0FH, 15_{10}.
15.
(a) 00FFH is added to the value in AX.
(b) Contents of AX and CF are added to the contents of SI.
(c) Contents of DS:100H are incremented by 1.
(d) Contents of BL are subtracted from the contents of DL.
(e) Contents of DS:200H and CF are subtracted from the contents of DL.
(f) Contents of the byte-wide data segment storage location pointed to by (DS)0 + (DI) + (BX) are decremented by 1.
(g) Contents of the byte-wide data segment storage location pointed to by (DS)0 + (DI) + 10H are replaced by its negative.
(h) Contents of word register DX are signed-multiplied by the word contents of AX. The double word product that results is produced in DX,AX.
(i) Contents of the byte storage location pointed to by (DS)0 + (BX) + (SI) are multiplied by the contents of AL.
(j) Contents of AX are signed-divided by the byte contents of the data segment storage location pointed to by (DS)0 + (SI) + 30H.
(k) Contents of AX are signed-divided by the byte contents of the data segment storage location pointed to by (DS)0 + (BX) + (SI) + 30H.

16.
(a) (AX) = 010FH
(b) (SI) = 0111H
(c) (DS:100H) = 11H
(d) (DL) = 20H
(e) (DL) = 0FH
(f) (DS:220H) = 2FH
(g) (DS:210H) = C0H
(h) (AX) = 0400H
(DX) = 0000H
(i) (AL) = F0H
(AH) = FFH
(j) (AL) = 02H
(AH) = 00H
(k) (AL) = 08H
(AH) = 00H
17. ADC DX,111FH
18. SBB AX,[BX]
19. ADD SI,2H,
or
INC SI
INC SI
20. (AH) = remainder = 3_{16}, (AL) = quotient = 12_{16}, therefore, (AX) = 0312_{16}.
21. DAA.
22. AAS.

23. (AX) = FFA0H.
24. (AX) = 7FFFH, (DX) = 0000H.
25. Let us assume that the memory locations NUM1, NUM2, and NUM3 are any memory locations in the same data segment.

```
MOV AX, DATA_SEG ;Establish data segment
MOV DS, AX
MOV AL, [NUM2] ;Get the second BCD number
SUB AL, [NUM1] ;Subtract the binary way
DAS ;Apply BCD adjustment
MOV [NUM3], AL ;Save the result.
```

Note that storage locations NUM1, NUM2, and NUM3 are assumed to have been declared as byte locations.

Solution to Questions in section 5.3 ➔ Problems 26-37

Section 5.3

26. **(a)** 00010000_2. **(b)** 01001100_2.
27. **(a)** 00011101_2. **(b)** 11011111_2.
28. 01010101_2, 55H.
29. 00011000_2, 18H.
30.
(a) 0FH is ANDed with the contents of the byte-wide memory address DS:300H.
(b) Contents of DX are ANDed with the contents of the word storage location pointed to by (DS)0 + (SI).
(c) Contents of AX are ORed with the word contents of the memory location pointed to by (DS)0 + (BX) + (DI).
(d) F0H is ORed with the contents of the byte-wide memory location pointed to by (DS)0 + (BX) + (DI) + 10H.
(e) Contents of the word-wide memory location pointed to by (DS)0 + (SI) + (BX) are exclusive-ORed with the contents of AX.
(f) The bits of the byte-wide memory location DS:300H are inverted.
(g) The bits of the word memory location pointed to by (DS)0 + (BX) + (DI) are inverted.
31.
(a) (DS:300H) = 0AH
(b) (DX) = A00AH
(c) (DS:210H) = FFFFH
(d) (DS:220H) = F5H
(e) (AX) = AA55H
(f) (DS:300H) = 55H
(g) (DS:210H) = 55H, (DS:211H) = 55H

32. AND DX,0080H
33. AND WORD PTR [100H],0080H.
34. The new contents of AX are the 2's complement of its old contents.
35. XOR AH,80H.
36. MOV AL,[CONTROL_FLAGS]
AND AL,81H
MOV [CONTROL_FLAGS],AL

37. The first instruction reads the byte of data from memory location CONTROL_FLAGS and loads it into BL. The AND instruction masks all bits but B_3 to 0; the XOR instruction toggles bit B_3 of this byte. That is, if the original value of B_3 equals logic 0, it is switched to 1 or if it is logic 1 it is switched to 0. Finally, the byte of flag information is written back to memory. This instruction sequence can be used to selectively complement one or more bits of the control flag byte.

Solution to Questions in section 5.4 ➔ Problems 38-45

Section 5.4

38.

(a) Contents of DX are shifted left by a number of bit positions equal to the contents of CL. LSBs are filled with zeros, and CF equals the value of the last bit shifted out of the MSB position.

(b) Contents of the byte-wide memory location DS:400H are shifted left by a number of bit positions equal to the contents of CL. LSBs are filled with zeros, and CF equals the value of the last bit shifted out of the MSB position.

(c) Contents of the byte-wide memory location pointed to by (DS)0 + (DI) are shifted right by 1 bit position. MSB is filled with zero, and CF equals the value shifted out of the LSB position.

(d) Contents of the byte-wide memory location pointed to by (DS)0 + (DI) + (BX) are shifted right by a number of bit positions equal to the contents of CL. MSBs are filled with zeros, and CF equals the value of the last bit shifted out of the LSB position.

(e) Contents of the word-wide memory location pointed to by (DS)0 + (BX) + (DI) are shifted right by 1 bit position. MSB is filled with the value of the original MSB and CF equals the value shifted out of the LSB position.

(f) Contents of the word-wide memory location pointed to by (DS)0 + (BX) + (DI) + 10H are shifted right by a number of bit positions equal to the contents of CL. MSBs are filled with the value of the original MSB, and CF equals the value of the last bit shifted out of the LSB position.

39.

(a) (DX) = 2220H, (CF) = 0

(b) (DS:400H) = 40H, (CF) = 1

(c) (DS:200H) = 11H, (CF) = 0

(d) (DS:210H) = 02H, (CF) = 1

(e) (DS:210H,211H) = D52AH, (CF) = 1

(f) (DS:220H,221H) = 02ADH, (CF) = 0

40. SHL CX,1

41. MOV CL,08H

SHL WORD PTR [DI],CL

42. The original contents of AX must have the four most significant bits equal to 0.

43. (AX) = F800H; CF =1.

44. The first instruction reads the byte of control flags into AL. Then all but the flag in the most significant bit location B_7 are masked off. Finally, the flag in B_7 is shifted to the left and into the carry flag. When the shift takes place, B_7 is shifted into CF; all other bits in AL move one bit position to the left, and the LSB locations are filled with zeros. Therefore, the contents of AL become 00H.

45. MOV AX, [ASCII_DATA] ;Get the word into AX

MOV BX,AX ;and BX

MOV CL,08H ;(CL) = bit count
SHR BX,CL ;(BX) = higher character
AND AX,00FFH ;(AX) = lower character
MOV [ASCII_CHAR_L],AX ;Save lower character
MOV [ASCII_CHAR_H],BX ;Save higher character

Solution to Questions in section 5.5 ➔ Problems 46-54

Section 5.5
46.
(a) Contents of DX are rotated left by a number of bit positions equal to the contents of CL. As each bit is rotated out of the MSB position, the LSB position and CF are filled with this value.
(b) Contents of the byte-wide memory location DS:400H are rotated left by a number of bit positions equal to the contents of CL. As each bit is rotated out of the MSB position, it is loaded into CF, and the prior contents of CF are loaded into the LSB position.
(c) Contents of the byte-wide memory location pointed to by (DS)0 + (DI) are rotated right by 1 bit position. As the bit is rotated out of the LSB position, the MSB position and CF are filled with this value.
(d) Contents of the byte-wide memory location pointed to by (DS)0 + (DI) + (BX) are rotated right by a number of bit positions equal to the contents of CL. As each bit is rotated out of the LSB position, the MSB position and CF are filled with this value.
(e) Contents of the word-wide memory location pointed to by (DS)0 + (BX) + (DI) are rotated right by 1 bit position. As the bit is rotated out of the LSB location, it is loaded into CF, and the prior contents of CF are loaded into the MSB position.
(f) Contents of the word-wide memory location pointed to by (DS)0 + (BX) + (DI) + 10H are rotated right by a number of bit positions equal to the contents of CL. As each bit is rotated out of the LSB position, it is loaded into CF, and the prior contents of CF are loaded into the MSB position.
47.
(a) (DX) = 2222H, (CF) = 0
(b) (DS:400H) = 5AH, (CF) = 1
(c) (DS:200H) = 11H, (CF) = 0
(d) (DS:210H) = AAH, (CF) = 1
(e) (DS:210H,211H) = D52AH, (CF) = 1
(f) (DS:220H,221H) = AAADH, (CF) = 0
48. RCL WORD PTR [BX],1
49. MOV BL,AL ; Move bit 5 to bit 0 position
MOV CL,5
SHR BX,CL
AND BX,1 ; Mask the other bit
50. MOV AX,[ASCII_DATA] ;Get the word into AX
MOV BX,AX ;and BX
MOV CL,08H ;(CL) = bit count
ROR BX,CL ;Rotate to position the higher character
AND AX,00FFH ;(AX) = lower character
AND BX,00FFH ;(BX) = higher character
MOV [ASCII_CHAR_L],AX ;Save lower character

MOV [ASCII_CHAR_H],BX ;Save higher character
Advanced Problems:
51. MOV AX,DATA_SEG ;Establish the data segment
MOV DS,AX
MOV AL,[MEM1] ;Get the given code at MEM1
MOV BX,TABL1
XLAT ;Translate
MOV [MEM1],AL ;Save new code at MEM1
MOV AL,[MEM2] ;Repeat for the second code at MEM2
MOV BX,TABL2
XLAT
MOV [MEM2],AL
52. MOV AX,0 ;Set up the data segment
MOV DS,AX
MOV BX,0A10H ;Set up pointer for results
MOV DX,[0A00H] ;Generate the sum
ADD DX,[0A02H]
MOV [BX],DX ;Save the sum
MOV DX,[0A00H] ;Generate the difference
SUB DX,[0A02H]
ADD BX,2 ;Save the difference
MOV [BX],DX
MOV AX,[0A00H] ;Generate the product
MUL [0A02H]
ADD BX,2 ;Save LS part of the product
MOV [BX],AX
ADD BX,2 ;Save MS part of the product
MOV [BX],DX
MOV AX,[0A00H] ;Generate the quotient
DIV AX,[0A02H]
ADD BX,2 ;Save the quotient
MOV [BX],AX
53.
; (RESULT) = (AL) • (NUM1) + (AL) • (NUM2---) + (BL)
NOT [NUM2] ;(NUM2) ← (NUM2---)
MOV CL, AL
AND CL, [NUM2] ;(CL) ← (AL) • (NUM2---)
OR CL, BL ;(CL) ← (AL) • (NUM2---) + (BL)
AND AL, [NUM1] ;(AL) ← (AL) • (NUM1)
OR AL, CL
MOV [RESULT],AL ;(RESULT)=(AL)•(NUM1)+(AL)•(NUM2---)+(BL)
54. Assume that all numbers are small enough so that shifting to the left does not generate an overflow. Further we will accept the truncation error due to shifts to the right.
MOV DX,AX ;(DX) ← (AX)
MOV CL,3
SHL DX,CL
SUB DX,AX
MOV SI,BX ;(SI) ← 5(BX)
MOV CL,2

```
SHL SI,CL
ADD SI,BX
SUB DX,SI ;(DX) ← 7(AX) – 5(BX)
MOV SI,BX ;(SI) ← (BX)/8
MOV CL,3
SAR SI,CL
SUB DX,SI ;(DX) ← 7(AX) – 5(BX) – (BX)/8
MOV AX,DX ;(AX) ← 7(AX) – 5(BX) – (BX)/8
```