CHAPTER 3

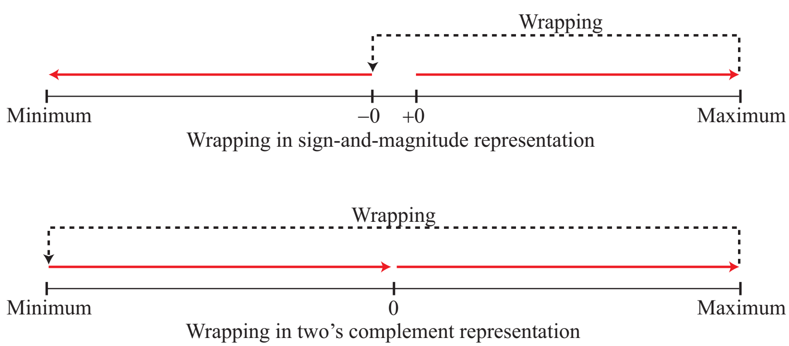
Data Storage

(Solutions to Practice Set)

Review Questions

1. We discussed five data types: numbers, text, audio, images, and video.
2. If the length of the bit pattern is L bits, the number of symbols that can be represented by the bit pattern is 2^L.
3. In the bitmap graphic method each pixel is represented by a bit pattern.
4. In vector graphic method, the size of the file is smaller and the image can be easily rescaled. However, vector graphic can not be used to represent the details of colors in a photo.
5. The three steps are sampling, quantization, and encoding.
6. Representations are the same except that the representable range of positive integers in unsigned method is twice the other methods.
7. In both representations, the upper half of the range represents the negative numbers. However, the wrapping is different as shown in Figure S3.7. In addition, there are two zeros in sign-and-magnitude but only one in two's complement.

Figure S3.7 *Question 7*



8. In the signed-and-magnitude representation, there are two zeros. In two's complement representation there is only one zero. In the excess representation, zero is represented by a positive number (bias) such as +127 and +1023.
9. In both systems, the leftmost bit represents the sign. If the leftmost bit is 0, the number is positive; if it is 1, the number is negative.
10.
    a. Normalization is necessary to make calculations easier.
    b. Mantissa is the bit sequence to the right of the decimal point after normalization.
    c. The computer stores the sign of the number, the exponent, and the mantissa.

Multiple-Choice Questions

11. c	12. c	13. d	14. d	15. b	16. d
17. a	18. b	19. a	20. d	21. d	22. d
23. c	24. a	25. d	26. c	27. b	

Exercises

28. $2^5 = 32$ patterns.
29. $10^2 = 100$ if zero is allowed. $9^2 = 81$ if zero is not allowed.
30.
    a. If zero is allowed, (10^2 for numbers) $\times$ (26^3 for letters) = 1757600.
    b. If zero is not allowed, (9^2 for numbers) $\times$ (26^3 for letters) = 1423656.
31. $2^n = 8 \rightarrow n = 3$ or $\log_2 8 = 3$.
32. $2^n = 7 \rightarrow n \approx 3$ or $\log_2 7 = 2.81 \rightarrow 3$.
33. $2^n = 900 \rightarrow n \approx 10$ or $\log_2 900 = 9.81 \rightarrow 10$. With $n = 10$ we can uniquely assign $2^{10} = 1024$ bit pattern. Then $1024 - 900 = 124$ patterns are unassigned. These unassigned patterns are not sufficient for extra 300 employees. If the company hires 300 new employees, it is needed to increase the number of bits to 11.
34. $2^4 - 10 = 6$ are wasted.
35. 256 level can be represented by 8 bits because $2^8 = 256$. Therefore, the number of bits per seconds is

$$(8000 \text{ sample/ sec}) \times (8 \text{ bits / sample}) = 64{,}000 \text{ bits /seconds}$$

36.
    a. $23 = 16 + 4 + 2 + 1 = (0000\ 1011)_2$
    b. $121 = 64 + 32 + 16 + 8 + 1 = (0111\ 1001)_2$

c. 34 = 32 +2 = $(0010\ 0010)_2$.

d. Overflow occurs because 342 > 255.

37.

a. 41 = 32 + 8 +1 = $(0000\ 0000\ 0010\ 1001)_2$.

b. 411 = 256 + 128 + 16 + 8 + 2 + 1 = $(0000\ 0001\ 1001\ 1011)_2$.

c. 1234 = 1024 + 128 + 64 + 16 + 2 = $(0000\ 0100\ 1101\ 0010)_2$.

d. 342 = 256 + 64 + 16 + 4 + 2 = $(0000\ 0001\ 0101\ 0110)_2$.

38.

a. –12 =

Convert 12 to binary	0	0	0	0	1	1	0	0
	↓	↓	↓	↓	↓	↓	↓	↓
Apply two’s complement operation	**1**	**1**	**1**	**1**	**0**	1	0	0

b. Overflow occurs because –145 is not in the range –128 to +127.

c. 56 =

Convert 56 to binary	0	0	1	1	1	0	0	0

d. Overflow occurs because 142 is not in the range –128 to +127.

39.

a. 102 =

Convert 102 to binary	0	0	0	0	0	0	0	0	0	1	1	0	0	1	1	0

b. –179 =

Convert 179 to binary	0	0	0	0	0	0	0	0	1	0	1	1	0	0	1	1
	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
Apply two’s complement operation	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**0**	**1**	**0**	**0**	**1**	**1**	**0**	1

c. 534 =

Convert 534 to binary	0	0	0	0	0	0	1	0	0	0	0	1	0	1	1	0

d. Over flow because 62,056 is not in the range (–32768, +32767).

40.

a. 0110 1011 = 64 + 32 + 8 + 2 +1 = 107.

b. 1001 0100 = 128 + 16 + 4 = 148.

c. 0000 0110 = 4 + 2 = 6.

d. 0101 0000 = 64 + 16 = 80.

41.

a. 0111 0111 =

Leftmost bit is 0. The sign is +	0	1	1	1	0	1	1	1
Integer changed to decimal								119
Sign is added								+ 119

b. 1111 1100 =

Leftmost bit is 1. The sign is −	1	1	1	1	1	1	0	0
	↓	↓	↓	↓	↓	↓	↓	↓
Apply two's complement operation	**0**	**0**	**0**	**0**	**0**	**1**	0	0
Integer changed to decimal								4
Sign is added								−4

c. 0111 0100 =

Leftmost bit is 0. The sign is +	0	1	1	1	0	1	0	0
Integer changed to decimal								116
Sign is added								+ 116

d. 1100 1110 =

Leftmost bit is 1. The sign is −	1	1	0	0	1	1	1	0
	↓	↓	↓	↓	↓	↓	↓	↓
Apply two's complement operation	**0**	**0**	**1**	**1**	**0**	**0**	1	0
Integer changed to decimal								50
Sign is added								−50

42. We change the sign of the number by applying the two's complement operation.

a. 01110111 → 10001001

b. 11111100 → 00000100

c. 01110111 → 10001001

d. 11001110 → 00110010

43.

a. 01110111 → 10001001 → 01110111

b. 11111100 → 00000100 → 11111100

c. 01110100 → 10001100 → 01110100

d. 11001110 → 00110010 → 11001110

44.

a. $1.10001 = 2^0 \times 1.10001$

b. $2^3 \times 111.1111 = 2^5 \times 1.111111$

c. $2^{-2} \times 101.110011 = 2^0 \times 1.01001100$

d. $2^{-5} \times 101101.00000110011000 = 2^0 \times 1.0110100000110011000$

45. Answers are shown with space between the three parts for clarity:

a. S = 1,
E = 0 + 127 = 127 = $(01111111)_2$,
M = 10001 (plus 18 zero added at the right to make the number of bits 23)
→ **1 01111111 10001000000000000000000**

b. S = 0,
E = 3 + 127 = 130 = $(10000010)_2$,
M = 111111 (plus 17 zero added at the right)
→ **0 10000010 11111100000000000000000**

c. S = 0
E = –4 + 127 = 123 = $(01111011)_2$,
M = 01110011 (plus 15 zero added at the right)
→ **0 01111011 01110011000000000000000**

d. S = 1
E = –5 + 127 = 122 = $(01111010)_2$,
M = 01101000 (plus 15 zero added at the right)
→ **1 01111010 01101000000000000000000**

46. Answers are shown with spaces between the three parts for clarity:

a. S = 1
E = 0 + 1023 = 1023 = $(01111111111)_2$,
M = 10001 (plus 47 zero added at the right)
→ **1 01111111111 1000100000000000000000000000000000000000000000000000**

b. S = 0
E = 3 + 1023 = 1026 = $(10000000010)_2$
M = 111111 (plus 46 zero added at the right)
→ **0 10000000010 1111110000000000000000000000000000000000000000000000**

c. S = 0
E = –4 + 1023 = 1019 = $(01111111011)_2$
M = 01110011 (plus 44 zero added at the right)
→ **0 01111111011 0111001100000000000000000000000000000000000000000000**

d. S = 1
E = –5 + 1023 = $(01111111010)_2$
M = 01101000 (plus 44 zero added at the right)
→ **1 01111111010 0110100000000000000000000000000000000000000000000000**

47. Answers are shown with spaces between the three parts for clarity:

a. 7.1875 = $(111.0011)_2 = 2^2 \times 1.110011$
S = 0
E = 2 + 127 = 129 = $(10000001)_2$
M = 110011 (plus 17 zero at the right)
→ **0 10000001 11001100000000000000000**

b. −12.640625 = $(–1100.101001)_2 = – 2^3 \times 1.100101001$
S = 1
E = 3 + 127 = 130 = $(10000010)_2$

M = 100101001 (plus 14 zero at the right)

→ **1 10000010 10010100100000000000000**

c. $11.40625 = (1011.01101)_2 = 2^3 \times 1.01101101$

S = 0

$E = 3 + 127 = 130 = (10000010)_2$

M = 01101101 (plus 15 zero at the right)

→ **0 10000010 01101101000000000000000**

d. $-0.375 = -0.011 = -2^{-2} \times 1.1$

S = 1

$E = -2 + 127 = 125 = (01111101)_2$

M = 1 (plus 22 zero at the right)

→ **1 01111101 10000000000000000000000**

48.

a. $(01110111)_2 =$

0	1	1	1	0	1	1	1		
↓	↓	↓	↓	↓	↓	↓	↓		
+	64	32	16	0	4	2	1	→	**+119**

b. $(11111100)_2 =$

1	1	1	1	1	1	0	0		
↓	↓	↓	↓	↓	↓	↓	↓		
–	64	32	16	8	4	2	1	→	**–124**

c. $(01110100)_2 =$

0	1	1	1	0	1	0	0		
↓	↓	↓	↓	↓	↓	↓	↓		
+	64	32	16	0	4	0	0	→	**+116**

d. $(11001110)_2 =$

1	1	0	0	1	1	1	0		
↓	↓	↓	↓	↓	↓	↓	↓		
–	64	0	0	8	4	2	0	→	**–78**

49.

a. 53 = 32 + 16 + 4 + 1 =

+	0	32	16	0	4	0	1		
↓	↓	↓	↓	↓	↓	↓	↓		
0	0	1	1	0	1	0	1	=	**0011 0101**

b. –107 = – (64 + 32 + 8 + 2 + 1) =

–	64	32	0	8	0	2	1		
↓	↓	↓	↓	↓	↓	↓	↓		
1	1	1	0	1	0	1	1	=	**1110 1011**

c. –5 = – (4+1) = 10000101

–	0	0	0	0	4	0	1		
↓	↓	↓	↓	↓	↓	↓	↓		
1	0	0	0	0	1	0	1	=	**1000 0101**

d. 154 creates overflow because 154 is not in the range –127 to +127.

50.

a. $(53)_{16}$ =

Convert 53 to binary	0	1	0	1	0	0	1	1

b. $(-107)_{16}$ =

Convert 107 to binary	0	1	1	0	1	0	1	1
	↓	↓	↓	↓	↓	↓	↓	↓
Apply one's complement operation	**1**	**0**	**0**	**1**	**0**	**1**	**0**	**0**

c. $(-5)_{16}$ =

Convert 5 to binary	0	0	0	0	0	1	0	1
	↓	↓	↓	↓	↓	↓	↓	↓
Apply one's complement operation	**1**	**1**	**1**	**1**	**1**	**0**	**1**	**0**

d. $(154)_{16}$ = Overflow because 154 is not in the range of –127 to 127

51.

a. $(01110111)_2$ =

Leftmost bit is 0. The sign is +	0	1	1	1	0	1	1	1
Integer changed to decimal								119
Sign is added								+119

b. $(11111100)_2$ =

Leftmost bit is 1. The sign is –	1	1	1	1	1	1	0	0
	↓	↓	↓	↓	↓	↓	↓	↓
Apply one's complement operation	**0**	**0**	**0**	**0**	**0**	**0**	1	1
Integer changed to decimal								3
Sign is added								–3

c. $(01110100)_2$ =

Leftmost bit is 0. The sign is +	0	1	1	1	0	1	0	0
Integer changed to decimal								116
Sign is added								+116

d. $(11001110)_2$ =

Leftmost bit is 1. The sign is –	1	1	0	0	1	1	1	0
	↓	↓	↓	↓	↓	↓	↓	↓
Apply one's complement operation	0	0	1	1	0	0	0	1
Integer changed to decimal								49
Sign is added								–49

52.

a. 01110111 → 10001000 → 01110111

b. 11111100 → 00000011 → 11111100

c. 01110100 → 10001011 → 01110100

d. 11001110 → 00110001 → 11001110

53.

a. $(01110111)_2$

One's complement =	10001000	Two's complement =	10001001
	+1		
	10001001		

b. $(11111100)_2$

One's complement =	00000011	Two's complement =	00000100
	+1		
	00000100		

c. $(01110100)_2$

One's complement =	10001011	Two's complement =	10001100
	+1		
	10001100		

d. $(11001110)_2$

One's complement =	00110001	Two's complement =	00110010
	+1		
	00110010		

54.

a. With 3 digits we can express $10^3 = 1000$ integers, 500 for positives and 500 negatives. Then we can represent numbers in the range of –499 to 499.

b. The first digit determine the sign of the number. The number is positive if the first digit is 0 to 4 and negative if the first digit is 5 to 9.

c. We have two zeros, one positive and one negative.

d. +0 = 000 and –0 = 999.

55.

a. +234 → 234.

b. +560 → Overflow because 560 is not in the range –499 to 499.

c. –125 → 874.

d. –111 → 888.

56.

a. With 3 digits we can represent $10^3 = 1000$ integers, 500 for zero and positives and 500 for negatives. Then we can represent numbers in the range of –500 to 499.

b. The first digit determine the sign of the number. The number is zero or positive if the first digit is 0 to 4 and negative if the first digit is 5 to 9.

c. No, there is only one representation for zero (0 = 000).

d. NA.

57.

a. +234 → 234.

b. +560 → Overflow because 560 is not in the range –500 to 499.

c. –125 → 874 + 1 = 875.

d. –111 → 888 + 1 = 889.

58.

a. With 3 digits we can represent $16^3 = 4096$ integers, 2048 for positives and 2048 for negatives. Then we can represent numbers in the range of $(–7FF)_{16}$ to $(7FF)_{16}$.

b. The fifteen's complement of a positive number is itself. To find the fifteen complement of negative numbers, we subtract each digit from 15.

c. We have two zeros, a positive zero and a negative zero.

d. $+0 = (000)_{16}$ and $–0 = (EEE)_{16}$.

59.

a. $(+B14)_{16} \rightarrow (B14)_{16}$.

b. $(+FE1)_{16} \rightarrow$ Overflow because it is not in the range $(–7FF)_{16}$ to $(7FF)_{16}$.

c. $(–1A)_{16} = (–01A)_{16} \rightarrow (FE5)_{16}$.

d. $(–1E2)_{16} \rightarrow (E1D)_{16}$.

60.

a. With 3 digits we can represent $16^3 = 4096$ integers, 2048 for zero and positives and 2048 for negatives. Then we can represent numbers in the range of $(–800)_{16}$ to $(7FF)_{16}$.

b. If the number is positive, the complement of the number is itself. If the number is negative we find the fifteen's complement and add 1 to it.

c. No, there is only one zero, $(000)_{16}$.
d. NA.

61.

a. $(+B14)_{16} \rightarrow (B14)_{16}$.
b. $(+FE1)_{16} \rightarrow$ Overflow occurs because it is not in the range $(–800)_{16}$ to $(7FF)_{16}$.
c. $(–1A)_{16} = (–01A)_{16} \rightarrow (FE5 +1)_{16} = (FE6)_{16}$.
d. $(–1E2)_{16} \rightarrow (E1D +1)_{16} = (E1E)_{16}$.