For exclusive use of adopters of the book *Digital Design Principles and Practices*, Fourth Edition, by John F. Wakerly, ISBN 0-13-186389-4.

3e2.1 2.1
(a) $1101011_2 = 6B_{16}$ (b) $174003_8 = 1111100000000011_2$
(c) $10110111_2 = B7_{16}$ (d) $67.24_8 = 110111.0101_2$
(e) $10100.1101_2 = 14.D_{16}$ (f) $F3A5_{16} = 1111001110100101_2$
(g) $11011001_2 = 331_8$ (h) $AB3D_{16} = 1010101100111101_2$
(i) $101111.0111_2 = 57.34_8$ (j) $15C.38_{16} = 101011100.00111_2$

3e2.3 2.3
(a) $1023_{16} = 1000000100011_2 = 10043_8$
(b) $7E6A_{16} = 111111001101010_2 = 77152_8$
(c) $ABCD_{16} = 1010101111001101_2 = 125715_8$
(d) $C350_{16} = 1100001101010000_2 = 141520_8$
(e) $9E36.7A_{16} = 1001111000110110.0111101_2 = 117066.364_8$
(f) $DEAD.BEEF_{16} = 1101111010101101.1011111011101111_2 = 157255.575674_8$

3e2.5 2.5
(a) $1101011_2 = 107_{10}$ (b) $174003_8 = 63491_{10}$
(c) $10110111_2 = 183_{10}$ (d) $67.24_8 = 55.3125_{10}$
(e) $10100.1101_2 = 20.8125_{10}$ (f) $F3A5_{16} = 62373_{10}$
(g) $12010_3 = 138_{10}$ (h) $AB3D_{16} = 43837_{10}$
(i) $7156_8 = 3694_{10}$ (j) $15C.38_{16} = 348.21875_{10}$

3e2.6 2.6
(a) $125_{10} = 1111101_2$ (b) $3489_{10} = 6641_8$
(c) $209_{10} = 11010001_2$ (d) $9714_{10} = 22762_8$
(e) $132_{10} = 1000100_2$ (f) $23851_{10} = 5D2B_{16}$
(g) $727_{10} = 10402_5$ (h) $57190_{10} = DF66_{16}$
(i) $1435_{10} = 2633_8$ (j) $65113_{10} = FE59_{16}$

3e2.18 2.20

$$h_j = \sum_{i=0} b_{4j+i} \cdot 2^j$$

Therefore,

$$B = \sum_{i-0}^{4n-1} b_i \cdot 2^i = \sum_{i=0}^{n-1} h_i \cdot 16^i$$

$$-B = 2^{4n} - \sum_{i=0}^{4n-1} b_i \cdot 2^i = 16^n - \sum_{i=0}^{n-1} h_i \cdot 16^i$$

Suppose a $3n$-bit number B is represented by an n-digit octal number Q. Then the two's-complement of B is represented by the 8's-complement of Q.

3e2.22 2.24 Starting with the arrow pointing at any number, adding a positive number causes overflow if the arrow is advanced through the +7 to –8 transition. Adding a negative number to any number causes overflow if the arrow is not advanced through the +7 to –8 transition.

For exclusive use of adopters of the book *Digital Design Principles and Practices*, Fourth Edition, by John F. Wakerly, ISBN 0-13-186389-4.

3e2.24 **2.26** Let the binary representation of X be $x_{n-1}x_{n-2}\ldots x_1x_0$. Then we can write the binary representation of Y as $x_mx_{m-1}\ldots x_1x_0$, where $m = n-d$. Note that x_{m-1} is the sign bit of Y. The value of Y is

$$Y = -2^{m-1} \cdot x_{m-1} + \sum_{i=0} x_i \cdot 2^i$$

The value of X is

$$X = -2^{n-1} \cdot x_{n-1} + \sum_{i=0}^{n-2} x_i \cdot 2^i$$

$$= -2^{n-1} \cdot x_{n-1} + Y + 2^{m-1} \cdot x_{m-1} + \sum_{i=m-1}^{n-2} x_i \cdot 2^i$$

$$= -2^{n-1} \cdot x_{n-1} + Y + 2 \cdot 2^{m-1} + \sum^{n-2} x_i \cdot 2^i$$

Case 1 $(x_{m-1} = 0)$ In this case, $X = Y$ if and only if $-2^{n-1} \cdot x_{n-1} + \sum_{i=m}^{n-2} x_i \cdot 2^i = 0$, which is true if and only if all of the discarded bits $(x_m \ldots x_{n-1})$ are 0, the same as x_{m-1}.

Case 2 $(x_{m-1} = 1)$ In this case, $X = Y$ if and only if $-2^{n-1} \cdot x_{n-1} + 2 \cdot 2^{m-1} + \sum_{i=m}^{n-2} x_i \cdot 2^i = 0$, which is true if and only if all of the discarded bits $(x_m \ldots x_{n-1})$ are 1, the same as x_{m-1}.

3e2.25 **2.27** If the radix point is considered to be just to the right of the leftmost bit, then the largest number is $1.11\cdots1$ and the 2's complement of D is obtained by subtracting it from 2 (singular possessive). Regardless of the position of the radix point, the 1s' complement is obtained by subtracting D from the largest number, which has all 1s (plural).

3e2.28 **2.30**

$$B = -b_{n-1} \cdot 2^{n-1} + \sum_{i=0}^{n-2} b_i \cdot 2^i$$

$$2B = -b_{n-1} \cdot 2^n + \sum_{i=0}^{n-2} b_i \cdot 2^{i+1}$$

Case 1 $(b_{n-1} = 0)$ First term is 0, summation terms have shifted coefficients as specified. Overflow if $b_{n-2} = 1$.

Case 2 $(b_{n-1} = 1)$ Split first term into two halves; one half is cancelled by summation term $b_{n-2} \cdot 2^{n-1}$ if $b_{n-2} = 1$. Remaining half and remaining summation terms have shifted coefficients as specified. Overflow if $b_{n-2} = 0$.

4e2.33 **2.35** 001–010, 011–100, 101–110, 111–000.

4e2.37 **2.38** The manufacturer's code fails every 4th time, for a total of 2^{n-2} for an n-bit encoding disc. A standard binary code fails when the LSB changes from 1 to 0, which changes the next bit (and possibly others), guaranteeing a problem. So, half the boundaries in a standard binary code are bad, a total of 2^{n-1} for an n-bit encoding disc. The manufacturer's code is only half as bad as a standard binary code.

3e2.36 **2.41** In the string representation, each position may have a 0, a 1, or an x, a total of three possibilities per position, and 3^n combinations in all.