

*Control Techniques for Digital Communication* by A. M. Michelson and A. H. Levesque (Wiley-Interscience, 1985). Hardware applications of codes in computer systems are discussed in *Error-Detecting Codes, Self-Checking Circuits, and Applications* by John F. Wakerly (Elsevier/North-Holland, 1978).

As shown in the above reference by Wakerly, ones'-complement checksum codes have the ability to detect long bursts of unidirectional errors; this is useful in communication channels where errors all tend to be in the same direction. The special computational properties of these codes also make them quite amenable to efficient checksum calculation by software programs, important for their use in the Internet Protocol; see RFC-1071 and RFC-1141. RFCs (Requests for Comments) are archived in many places on the Web; just search for "RFC".

An introduction to coding techniques for serial data transmission, including mathematical analysis of the performance and bandwidth requirements of several codes, appears in *Introduction to Communications Engineering* by R. M. Gagliardi (Wiley-Interscience, 1988, second edition). A nice introduction to the serial codes used in magnetic disks and tapes is given in *Computer Storage Systems and Technology* by Richard Matick (Wiley-Interscience, 1977).

The structure of the 8B10B code and the rationale behind it is explained nicely in the original IBM patent by Peter Franaszek and Albert Widmer, U.S. patent number 4,486,739 (1984). This and almost all U.S. patents issued after 1971 can be found on the Web at [www.patents.ibm.com](http://www.patents.ibm.com).

## Drill Problems

2.1 Perform the following number system conversions:

- |                             |                         |
|-----------------------------|-------------------------|
| (a) $1101011_2 = ?_{16}$    | (b) $174003_8 = ?_2$    |
| (c) $10110111_2 = ?_{16}$   | (d) $67.24_8 = ?_2$     |
| (e) $10100.1101_2 = ?_{16}$ | (f) $F3A5_{16} = ?_2$   |
| (g) $11011001_2 = ?_8$      | (h) $AB3D_{16} = ?_2$   |
| (i) $101111.0111_2 = ?_8$   | (j) $15C.38_{16} = ?_2$ |

2.2 Convert the following octal numbers into binary and hexadecimal:

- |                                |                                  |
|--------------------------------|----------------------------------|
| (a) $1023_8 = ?_2 = ?_{16}$    | (b) $761302_8 = ?_2 = ?_{16}$    |
| (c) $163417_8 = ?_2 = ?_{16}$  | (d) $552273_8 = ?_2 = ?_{16}$    |
| (e) $5436.15_8 = ?_2 = ?_{16}$ | (f) $13705.207_8 = ?_2 = ?_{16}$ |

2.3 Convert the following hexadecimal numbers into binary and octal:

- |                                |                                  |
|--------------------------------|----------------------------------|
| (a) $1023_{16} = ?_2 = ?_8$    | (b) $7E6A_{16} = ?_2 = ?_8$      |
| (c) $ABCD_{16} = ?_2 = ?_8$    | (d) $C350_{16} = ?_2 = ?_8$      |
| (e) $9E36.7A_{16} = ?_2 = ?_8$ | (f) $DEAD.BEEF_{16} = ?_2 = ?_8$ |

- 2.4 What are the octal values of the four 8-bit bytes in the 32-bit number with octal representation  $12345670123_8$ ?
- 2.5 Convert the following numbers into decimal:
- (a)  $1101011_2 = ?_{10}$       (b)  $174003_8 = ?_{10}$   
 (c)  $10110111_2 = ?_{10}$       (d)  $67.24_8 = ?_{10}$   
 (e)  $10100.1101_2 = ?_{10}$       (f)  $F3A5_{16} = ?_{10}$   
 (g)  $12010_3 = ?_{10}$       (h)  $AB3D_{16} = ?_{10}$   
 (i)  $7156_8 = ?_{10}$       (j)  $15C.38_{16} = ?_{10}$
- 2.6 Perform the following number-system conversions:
- (a)  $125_{10} = ?_2$       (b)  $3489_{10} = ?_8$   
 (c)  $209_{10} = ?_2$       (d)  $9714_{10} = ?_8$   
 (e)  $132_{10} = ?_2$       (f)  $23851_{10} = ?_{16}$   
 (g)  $727_{10} = ?_5$       (h)  $57190_{10} = ?_{16}$   
 (i)  $1435_{10} = ?_8$       (j)  $65113_{10} = ?_{16}$
- 2.7 Add the following pairs of binary numbers, showing all carries:
- (a) 
$$\begin{array}{r} 110101 \\ + 11001 \\ \hline \end{array}$$
      (b) 
$$\begin{array}{r} 101110 \\ + 100101 \\ \hline \end{array}$$
      (c) 
$$\begin{array}{r} 11011101 \\ + 1100011 \\ \hline \end{array}$$
      (d) 
$$\begin{array}{r} 1110010 \\ + 1101101 \\ \hline \end{array}$$
- 2.8 Repeat Drill 2.7 using subtraction instead of addition, and showing borrows instead of carries.
- 2.9 Add the following pairs of octal numbers:
- (a) 
$$\begin{array}{r} 1372 \\ + 4631 \\ \hline \end{array}$$
      (b) 
$$\begin{array}{r} 47135 \\ + 5125 \\ \hline \end{array}$$
      (c) 
$$\begin{array}{r} 175214 \\ + 152405 \\ \hline \end{array}$$
      (d) 
$$\begin{array}{r} 110321 \\ + 56573 \\ \hline \end{array}$$
- 2.10 Add the following pairs of hexadecimal numbers:
- (a) 
$$\begin{array}{r} 1372 \\ + 4631 \\ \hline \end{array}$$
      (b) 
$$\begin{array}{r} 4F1A5 \\ + B8D5 \\ \hline \end{array}$$
      (c) 
$$\begin{array}{r} F35B \\ + 27E6 \\ \hline \end{array}$$
      (d) 
$$\begin{array}{r} 1B90F \\ + C44E \\ \hline \end{array}$$
- 2.11 Write the 8-bit signed-magnitude, two's-complement, and ones'-complement representations for each of these decimal numbers: +18, +115, +79, -49, -3, -100.
- 2.12 Indicate whether or not overflow occurs when adding the following 8-bit two's-complement numbers:
- (a) 
$$\begin{array}{r} 11010100 \\ + 10101011 \\ \hline \end{array}$$
      (b) 
$$\begin{array}{r} 10111001 \\ + 11010110 \\ \hline \end{array}$$
      (c) 
$$\begin{array}{r} 01011101 \\ + 00100001 \\ \hline \end{array}$$
      (d) 
$$\begin{array}{r} 00100110 \\ + 01011010 \\ \hline \end{array}$$
- 2.13 How many errors can be detected by a code with minimum distance  $d$ ?
- 2.14 What is the minimum number of parity bits required to obtain a distance-4, two-dimensional code with  $n$  information bits?

## Exercises

2.15 Here's a problem to whet your appetite. What is the hexadecimal equivalent of  $61453_{10}$ ?

2.16 Each of the following arithmetic operations is correct in at least one number system. Determine possible radices of the numbers in each operation.

- (a)  $1234 + 5432 = 6666$       (b)  $41 / 3 = 13$   
 (c)  $33/3 = 11$                       (d)  $23+44+14+32 = 223$   
 (e)  $302/20 = 12.1$                 (f)  $\sqrt{41} = 5$

2.17 The first expedition to Mars found only the ruins of a civilization. From the artifacts and pictures, the explorers deduced that the creatures who produced this civilization were four-legged beings with a tentacle that branched out at the end with a number of grasping "fingers." After much study, the explorers were able to translate Martian mathematics. They found the following equation:

$$5x^2 - 50x + 125 = 0$$

with the indicated solutions  $x = 5$  and  $x = 8$ . The value  $x = 5$  seemed legitimate enough, but  $x = 8$  required some explanation. Then the explorers reflected on the way in which Earth's number system developed, and found evidence that the Martian system had a similar history. How many fingers would you say the Martians had? (From *The Bent of Tau Beta Pi*, February 1956.)

2.18 Suppose a  $4n$ -bit number  $B$  is represented by an  $n$ -digit hexadecimal number  $H$ . Prove that the two's complement of  $B$  is represented by the 16's complement of  $H$ . Make and prove true a similar statement for octal representation.

2.19 Repeat Exercise 2.18 using the ones' complement of  $B$  and the 15s' complement of  $H$ .

2.20 Given an integer  $x$  in the range  $-2n^{-1} \leq x \leq 2n^{-1} - 1$ , we define  $[x]$  to be the two's-complement representation of  $x$ , expressed as a positive number:  $[x] = x$  if  $x \geq 0$  and  $[x] = 2n - |x|$  if  $x < 0$ , where  $|x|$  is the absolute value of  $x$ . Let  $y$  be another integer in the same range as  $x$ . Prove that the two's-complement addition rules given in Section 2.6 are correct by proving that the following equation is always true:

$$[x + y] = [x] + [y] \text{ modulo } 2^n$$

(Hints: Consider four cases based on the signs of  $x$  and  $y$ . Without loss of generality, you may assume that  $|x| \geq |y|$ .)

2.21 Repeat Exercise 2.20 using appropriate expressions and rules for ones'-complement addition.

2.22 State an overflow rule for addition of two's-complement numbers in terms of counting operations in the modular representation of Figure 2-3.

2.23 Show that a two's-complement number can be converted to a representation with more bits by *sign extension*. That is, given an  $n$ -bit two's-complement number  $X$ , show that the  $m$ -bit two's-complement representation of  $X$ , where  $m > n$ , can be obtained by appending  $m - n$  copies of  $X$ 's sign bit to the left of the  $n$ -bit representation of  $X$ .

- 2.24 Show that a two's-complement number can be converted to a representation with fewer bits by removing higher-order bits. That is, given an  $n$ -bit two's-complement number  $X$ , show that the  $m$ -bit two's-complement number  $Y$  obtained by discarding the  $d$  leftmost bits of  $X$  represents the same number as  $X$  if and only if the discarded bits all equal the sign bit of  $Y$ .
- 2.25 Why is the punctuation of "two's complement" and "ones' complement" inconsistent? (See the first two citations in the References.)
- 2.26 A  $n$ -bit binary adder can be used to perform an  $n$ -bit unsigned subtraction operation  $X - Y$ , by performing the operation  $X + \bar{Y} + 1$ , where  $X$  and  $Y$  are  $n$ -bit unsigned numbers and  $\bar{Y}$  represents the bit-by-bit complement of  $Y$ . Demonstrate this fact as follows. First, prove that  $(X - Y) = (X + \bar{Y} + 1) - 2^n$ . Second, prove that the carry out of the  $n$ -bit adder is the opposite of the borrow from the  $n$ -bit subtraction. That is, show that the operation  $X - Y$  produces a borrow out of the MSB position if and only if the operation  $X + \bar{Y} + 1$  *does not* produce a carry out of the MSB position.
- 2.27 In most cases, the product of two  $n$ -bit two's-complement numbers requires fewer than  $2n$  bits to represent it. In fact, there is only one case in which  $2n$  bits are needed—find it.
- 2.28 Prove that a two's-complement number can be multiplied by 2 by shifting it one bit position to the left, with a carry of 0 into the least significant bit position and disregarding any carry out of the most significant bit position, assuming no overflow. State the rule for detecting overflow.
- 2.29 State and prove correct a technique similar to the one described in Exercise 2.28, for multiplying a ones'-complement number by 2.
- 2.30 Show how to subtract BCD numbers, by stating the rules for generating borrows and applying a correction factor. Show how your rules apply to each of the following subtractions:  $9 - 3$ ,  $5 - 7$ ,  $4 - 9$ ,  $1 - 8$ .
- 2.31 How many different 3-bit binary state encodings are possible for the traffic-light controller of Table 2-12?
- 2.32 List all of the "bad" boundaries in the mechanical encoding disk of Figure 2-5, where an incorrect position may be sensed.
- 2.33 As a function of  $n$ , how many "bad" boundaries are there in a mechanical encoding disk that uses an  $n$ -bit binary code?
- 2.34 On-board altitude transponders on commercial and private aircraft use Gray code to encode the altitude readings that are transmitted to air traffic controllers. Why?
- 2.35 An incandescent light bulb is stressed every time it is turned on, so in some applications the lifetime of the bulb is limited by the number of on/off cycles rather than the total time it is illuminated. Use your knowledge of codes to suggest a way to double the lifetime of 3-way bulbs in such applications.
- 2.36 As a function of  $n$ , how many different distinct subcubes of an  $n$ -cube are there?
- 2.37 Find a way to draw a 3-cube on a sheet of paper (or other two-dimensional object) so that none of the lines cross, or prove that it's impossible.
- 2.38 Repeat Exercise 2.37 for a 4-cube.

- 2.39 Write a formula that gives the number of  $m$ -subcubes of an  $n$ -cube for a specific value of  $m$ . (Your answer should be a function of  $n$  and  $m$ .)
- 2.40 Define parity groups for a distance-3 Hamming code with 11 information bits.
- 2.41 Write the code words of a Hamming code with one information bit.
- 2.42 Exhibit the pattern for a 3-bit error that is not detected if the “corner” parity bits are not included in the two-dimensional codes of Figure 2-14.
- 2.43 The *rate of a code* is the ratio of the number of information bits to the total number of bits in a code word. High rates, approaching 1, are desirable for efficient transmission of information. Construct a graph comparing the rates of distance-2 parity codes and distance-3 and -4 Hamming codes for up to 100 information bits.
- 2.44 Which type of distance-4 code has a higher rate—a two-dimensional code or a Hamming code? Support your answer with a table in the style of Table 2-15, including the rate as well as the number of parity and information bits of each code for up to 100 information bits.
- 2.45 Show how to construct a distance-6 code with four information bits. Write a list of its code words.
- 2.46 Describe the operations that must be performed in a RAID system to write new data into information block  $b$  in drive  $d$ , so the data can be recovered in the event of an error in block  $b$  in any drive. Minimize the number of disk accesses required.
- 2.47 In the style of Figure 2-17, draw the waveforms for the bit pattern 10101110 when sent serially using the NRZ, NRZI, RZ, BPRZ, and Manchester codes, assuming that the bits are transmitted in order from left to right.