

Chapter 1

Numbers Used in Digital Electronics

1-1 INTRODUCTION

The decimal number system is familiar to everyone. This system uses the symbols 0, 1, 2, 3, 4, 5, 6, 7, 8, and 9. The decimal system also has a place-value characteristic. Consider the decimal number 238. The 8 is in the 1s position or place. The 3 is in the 10s position, and therefore the three 10s stand for 30 units. The 2 is in the 100s position and means two 100s, or 200 units. Adding $200 + 30 + 8$ gives the total decimal number of 238. The decimal number system is also called the *base 10 system*. It is referred to as base 10 because it has 10 different symbols. The base 10 system is also said to have a *radix* of 10. “Radix” and “base” are terms that mean exactly the same thing.

Binary numbers (base 2) are used extensively in digital electronics and computers. Both hexadecimal (base 16) and octal (base 8) numbers are used to represent groups of binary digits. Binary and hexadecimal numbers find wide use in modern microcomputers.

All the number systems mentioned (decimal, binary, octal, and hexadecimal) can be used for counting. All these number systems also have the place-value characteristic.

1-2 BINARY NUMBERS

The binary number system uses only two symbols (0,1). It is said to have a radix of 2 and is commonly called the *base 2 number system*. Each *binary digit* is called a *bit*.

Counting in binary is illustrated in Fig. 1-1. The binary number is shown on the right with its decimal equivalent. Notice that the *least significant bit* (LSB) is the 1s place. In other words, if a 1 appears in the right column, a 1 is added to the binary count. The second place over from the right is the 2s place. A 1 appearing in this column (as in decimal 2 row) means that 2 is added to the count. Three other binary place values also are shown in Fig. 1-1 (4s, 8s, and 16s places). Note that each larger place value is an added power of 2. The 1s place is really 2^0 , the 2s place 2^1 , the 4s place 2^2 , the 8s place 2^3 , and the 16s place 2^4 . It is customary in digital electronics to memorize at least the binary counting sequence from 0000 to 1111 (say: one, one, one, one) or decimal 15.

Consider the number shown in Fig. 1-2a. This figure shows how to convert the binary 10011 (say: one, zero, zero, one, one) to its decimal equivalent. Note that, for each 1 bit in the binary number, the decimal equivalent for that place value is written below. The decimal numbers are then added ($16 + 2 + 1 = 19$) to yield the decimal equivalent. Binary 10011 then equals a decimal 19.

Consider the binary number 101110 in Fig. 1-2b. Using the same procedure, each 1 bit in the binary number generates a decimal equivalent for that place value. The *most significant bit* (MSB) of the binary number is equal to 32. Add 8 plus 4 plus 2 to the 32 for a total of 46. Binary 101110 then equals decimal 46. Figure 1-2b also identifies the binary point (similar to the decimal point in decimal numbers). It is customary to omit the binary point when working with whole binary numbers.

What is the value of the number 111? It could be one hundred and eleven in decimal or one, one, one in binary. Some books use the system shown in Fig. 1-2c to designate the base, or radix, of a number. In this case 10011 is a base 2 number as shown by the small subscript 2 after the number. The number 19 is a base 10 number as shown by the subscript 10 after the number. Figure 1-2c is a summary of the binary-to-decimal conversions in Fig. 1-2a and b.

How about converting fractional numbers? Figure 1-3 illustrates the binary number 1110.101 being converted to its decimal equivalent. The place values are given across the top. Note the value of each position to the right of the binary point. The procedure for making the conversion is the same as with whole numbers. The place value of each 1 bit in the binary number is added to form the decimal number. In this problem $8 + 4 + 2 + 0.5 + 0.125 = 14.625$ in decimal.

Decimal count	Binary count				
	16s	8s	4s	2s	1s
0					0
1					1
2				1	0
3				1	1
4			1	0	0
5			1	0	1
6			1	1	0
7			1	1	1
8		1	0	0	0
9		1	0	0	1
10		1	0	1	0
11		1	0	1	1
12		1	1	0	0
13		1	1	0	1
14		1	1	1	0
15		1	1	1	1
16	1	0	0	0	0
17	1	0	0	0	1
18	1	0	0	1	0
19	1	0	0	1	1
	2^4	2^3	2^2	2^1	2^0
	Powers of 2				

Fig. 1-1 Counting in binary and decimal

Powers of 2	2^4	2^3	2^2	2^1	2^0	
Place value	16s	8s	4s	2s	1s	
Binary	1	0	0	1	1	• ← Binary point
Decimal	16		+	2	+	1 = 19

(a) Binary-to-decimal conversion

Powers of 2	2^5	2^4	2^3	2^2	2^1	2^0	
Place value	32s	16s	8s	4s	2s	1s	
Binary	1	0	1	1	1	0	• ← Binary point
Decimal	32	+	8	+	4	+	2 = 46

(b) Binary-to-decimal conversion

$$10011_2 = 19_{10} \quad 101110_2 = 46_{10}$$

(c) Summary of conversions and use of small subscripts to indicate radix of number

Fig. 1-2

Powers of 2	2^3	2^2	2^1	2^0	$1/2^1$	$1/2^2$	$1/2^3$				
Place value	8s	4s	2s	1s	0.5s	0.25s	0.125s				
Binary	1	1	1	0	.	1	0	1			
Decimal	8	+	4	+	2	+	0.5	+	0.125	=	14.625

Fig. 1-3 Binary-to-decimal conversion

Convert the decimal number 87 to a binary number. Figure 1-4 shows a convenient method for making this conversion. The decimal number 87 is first divided by 2, leaving 43 with a remainder of 1. The remainder is important and is recorded at the right. It becomes the LSB in the binary number. The quotient (43) then is transferred as shown by the arrow and becomes the dividend. The quotients are repeatedly divided by 2 until the quotient becomes 0 with a remainder of 1, as in the last line of Fig. 1-4. Near the bottom the figure shows that decimal 87 equals binary 1010111.

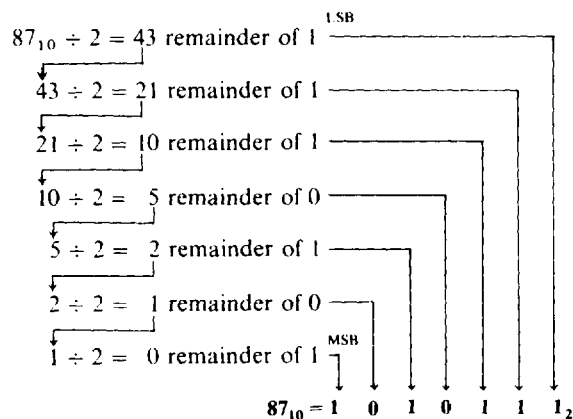


Fig. 1-4 Decimal-to-binary conversion

Convert the decimal number 0.375 to a binary number. Figure 1-5a illustrates one method of performing this task. Note that the decimal number (0.375) is being *multiplied* by 2. This leaves a product of 0.75. The 0 from the integer place (1s place) becomes the bit nearest the binary point. The 0.75 is then multiplied by 2, yielding 1.50. The carry of 1 to the integer (1s place) is the next bit in the binary number. The 0.50 is then multiplied by 2, yielding a product of 1.00. The carry of 1 in the integer place is the final 1 in the binary number. When the product is 1.00, the conversion process is complete. Figure 1-5a shows a decimal 0.375 being converted into a binary equivalent of 0.011.

Figure 1-5b shows the decimal number 0.84375 being converted into binary. Again note that 0.84375 is multiplied by 2. The integer of each product is placed below, forming the binary number. When the product reaches 1.00, the conversion is complete. This problem shows a decimal 0.84375 being converted to binary 0.11011.

Consider the decimal number 5.625. Converting this number to binary involves two processes. The integer part of the number (5) is processed by *repeated division* near the top in Fig. 1-6. Decimal 5 is converted to a binary 101. The fractional part of the decimal number (.625) is converted to binary .101 at the bottom in Fig. 1-6. The fractional part is converted to binary through the *repeated multiplication* process. The integer and fractional sections are then combined to show that decimal 5.625 equals binary 101.101.

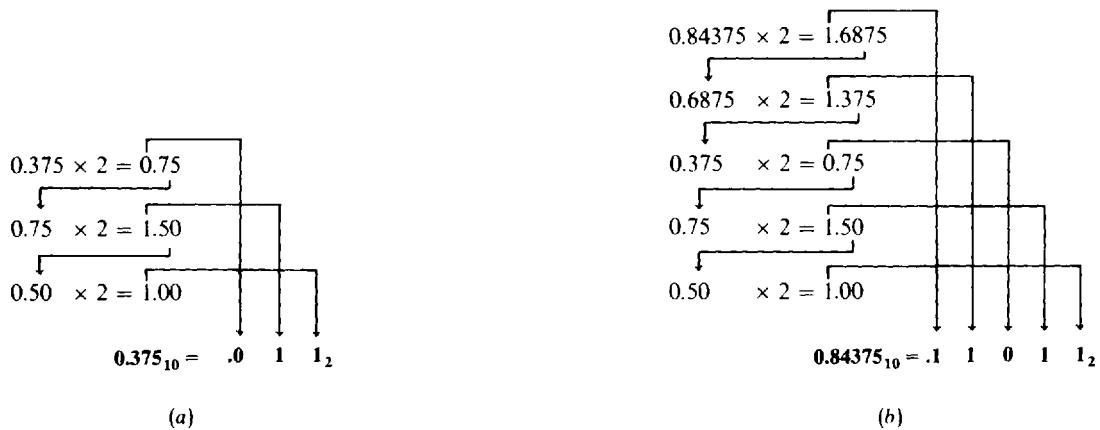


Fig. 1-5 Fractional decimal-to-binary conversions

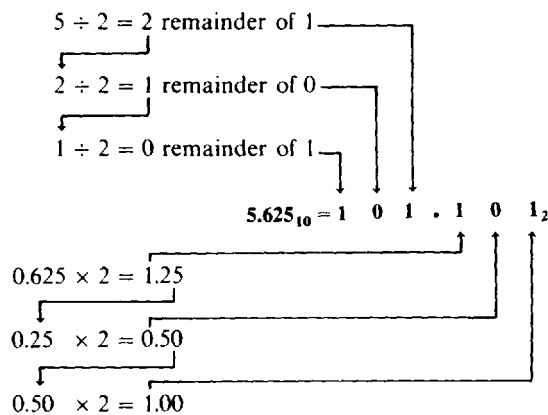


Fig. 1-6 Decimal-to-binary conversion

SOLVED PROBLEMS

- 1.1 The binary number system is the base _____ system and has a radix of _____.

Solution:

The binary number system is the base 2 system and has a radix of 2.

- 1.2 The term bit means _____ when dealing with binary numbers.

Solution:

Bit means binary digit.

- 1.3 How would you say the number 1001 in (a) binary and (b) decimal?

Solution:

The number 1001 is pronounced as follows: (a) one, zero, zero, one; (b) one thousand and one.

- 1.4 The number 110₁₀ is a base _____ number.

Solution:

The number 110₁₀ is a base 10 number, as indicated by the small 10 after the number.

- 1.5** Write the base 2 number one, one, zero, zero, one.

Solution:

$$11001_2$$

- 1.6** Convert the following binary numbers to their decimal equivalents:

(a) 001100 (b) 000011 (c) 011100 (d) 111100 (e) 101010 (f) 111111
(g) 100001 (h) 111000

Solution:

Follow the procedure shown in Fig. 1-2. The decimal equivalents of the binary numbers are as follows:

(a) $001100_2 = 12_{10}$ (c) $011100_2 = 28_{10}$ (e) $101010_2 = 42_{10}$ (g) $100001_2 = 33_{10}$
(b) $000011_2 = 3_{10}$ (d) $111100_2 = 60_{10}$ (f) $111111_2 = 63_{10}$ (h) $111000_2 = 56_{10}$

- 1.7** $11110001111_2 = \underline{\hspace{2cm}}_{10}$

Solution:

Follow the procedure shown in Fig. 1-2. $11110001111_2 = 1935_{10}$.

- 1.8** $11100.011_2 = \underline{\hspace{2cm}}_{10}$

Solution:

Follow the procedure shown in Fig. 1-3. $11100.011_2 = 28.375_{10}$.

- 1.9** $110011.10011_2 = \underline{\hspace{2cm}}_{10}$

Solution:

Follow the procedure shown in Fig. 1-3. $110011.10011_2 = 51.59375_{10}$.

- 1.10** $1010101010.1_2 = \underline{\hspace{2cm}}_{10}$

Solution:

Follow the procedure shown in Fig. 1-3. $1010101010.1_2 = 682.5_{10}$.

- 1.11** Convert the following decimal numbers to their binary equivalents:

(a) 64, (b) 100, (c) 111, (d) 145, (e) 255, (f) 500.

Solution:

Follow the procedure shown in Fig. 1-4. The binary equivalents of the decimal numbers are as follows:

(a) $64_{10} = 1000000_2$ (c) $111_{10} = 1101111_2$ (e) $255_{10} = 11111111_2$
(b) $100_{10} = 1100100_2$ (d) $145_{10} = 10010001_2$ (f) $500_{10} = 111110100_2$

- 1.12** $34.75_{10} = \underline{\hspace{2cm}}_2$

Solution:

Follow the procedure shown in Fig. 1-6. $34.75_{10} = 100010.11_2$.

1.13 $25.25_{10} = \text{_____}_2$

Solution:

Follow the procedure shown in Fig. 1-6. $25.25_{10} = 11001.01_2$.

1.14 $27.1875_{10} = \text{_____}_2$

Solution:

Follow the procedure shown in Fig. 1-6. $27.1875_{10} = 11011.0011_2$.

1-3 HEXADECIMAL NUMBERS

The hexadecimal number system has a radix of 16. It is referred to as the *base 16 number system*. It uses the symbols 0-9, A, B, C, D, E, and F as shown in the hexadecimal column of the table in Fig. 1-7. The letter A stands for a count of 10, B for 11, C for 12, D for 13, E for 14, and F for 15. The advantage of the hexadecimal system is its usefulness in converting directly from a 4-bit binary number. Note in the shaded section of Fig. 1-7 that each 4-bit binary number from 0000 to 1111 can be represented by a unique hexadecimal digit.

Decimal	Binary	Hexadecimal	Decimal	Binary	Hexadecimal
0	0000	0	16	10000	10
1	0001	1	17	10001	11
2	0010	2	18	10010	12
3	0011	3	19	10011	13
4	0100	4	20	10100	14
5	0101	5	21	10101	15
6	0110	6	22	10110	16
7	0111	7	23	10111	17
8	1000	8	24	11000	18
9	1001	9	25	11001	19
10	1010	A	26	11010	1A
11	1011	B	27	11011	1B
12	1100	C	28	11100	1C
13	1101	D	29	11101	1D
14	1110	E	30	11110	1E
15	1111	F	31	11111	1F

Fig. 1-7 Counting in decimal, binary, and hexadecimal number systems

Look at the line labeled 16 in the decimal column in Fig. 1-7. The hexadecimal equivalent is 10. This shows that the hexadecimal number system uses the place-value idea. The 1 (in 10_{16}) stands for 16 units, while the 0 stands for zero units.

Convert the hexadecimal number 2B6 into a decimal number. Figure 1-8a shows the familiar process. The 2 is in the 256s place so $2 \times 256 = 512$, which is written in the decimal line. The hexadecimal digit B appears in the 16s column. Note in Fig. 1-8 that hexadecimal B corresponds to decimal 11. This means that there are eleven 16s (16×11), yielding 176. The 176 is added into the decimal total near the bottom in Fig. 1-8a. The 1s column shows six 1s. The 6 is added into the decimal line. The decimal values are added ($512 + 176 + 6 = 694$), yielding 694_{10} . Figure 1-8a shows that $2B6_{16}$ equals 694_{10} .

hexadecimal number. The process is complete because the integer part of the quotient is 0. The process in Fig. 1-9a converts the decimal number 45 to the hexadecimal number 2D.

Convert the decimal number 250.25 to a hexadecimal number. The conversion must be done by using two processes as shown in Fig. 1-9b. The integer part of the decimal number (250) is converted to hexadecimal by using the repeated divide-by-16 process. The remainders of 10 (A in hexadecimal) and 15 (F in hexadecimal) form the hexadecimal whole number FA. The fractional part of the 250.25 is multiplied by 16 (0.25×16). The result is 4.00. The integer 4 is transferred to the position shown in Fig. 1-9b. The completed conversion shows the decimal number 250.25 equaling the hexadecimal number FA.4.

The prime advantage of the hexadecimal system is its easy conversion to binary. Figure 1-10a shows the hexadecimal number 3B9 being converted to binary. Note that each hexadecimal digit forms a group of four binary digits, or bits. The groups of bits are then combined to form the binary number. In this case $3B9_{16}$ equals 1110111001_2 .

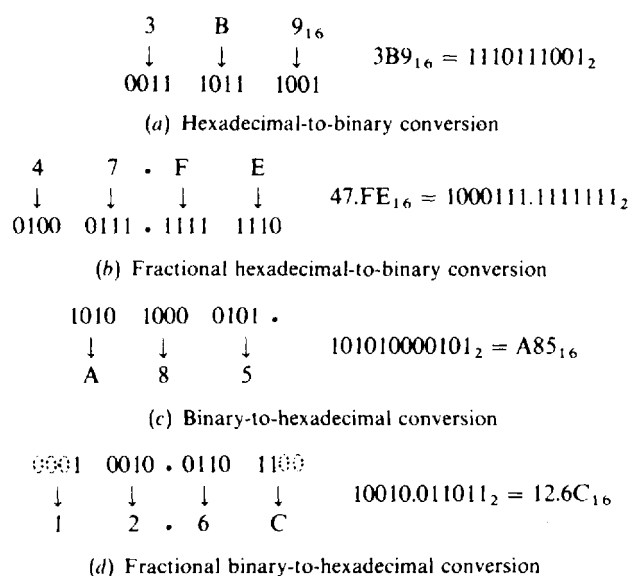


Fig. 1-10

Another hexadecimal-to-binary conversion is detailed in Fig. 1-10b. Again each hexadecimal digit forms a 4-bit group in the binary number. The hexadecimal point is dropped straight down to form the binary point. The hexadecimal number 47.FE is converted to the binary number 1000111.1111111_2 . It is apparent that hexadecimal numbers, because of their compactness, are much easier to write down than the long strings of 1s and 0s in binary. The hexadecimal system can be thought of as a shorthand method of writing binary numbers.

Figure 1-10c shows the binary number 101010000101 being converted to hexadecimal. First divide the binary number into 4-bit groups *starting at the binary point*. Each group of four bits is then translated into an equivalent hexadecimal digit. Figure 1-10c shows that binary 101010000101 equals hexadecimal A85.

Another binary-to-hexadecimal conversion is illustrated in Fig. 1-10d. Here binary 10010.011011 is to be translated into hexadecimal. First the binary number is divided into groups of four bits, starting at the binary point. Three 0s are added in the leftmost group, forming 0001. Two 0s are added to the rightmost group, forming 1100. Each group now has 4 bits and is translated into a hexadecimal digit as shown in Fig. 1-10d. The binary number 10010.011011 then equals $12.6C_{16}$.

As a practical matter, many modern hand-held calculators perform number base conversions. Most can convert between decimal, hexadecimal, octal, and binary. These calculators can also perform arithmetic operations in various bases (such as hexadecimal).

SOLVED PROBLEMS

- 1.15** The hexadecimal number system is sometimes called the base _____ system.

Solution:

The hexadecimal number system is sometimes called the base 16 system.

- 1.16** List the 16 symbols used in the hexadecimal number system.

Solution:

Refer to Fig. 1-7. The 16 symbols used in the hexadecimal number system are 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, and F.

- 1.17** Convert the following whole hexadecimal numbers to their decimal equivalents:

(a) C, (b) 9F, (c) D52, (d) 67E, (e) ABCD.

Solution:

Follow the procedure shown in Fig. 1-8a. Refer also to Fig. 1-7. The decimal equivalents of the hexadecimal numbers are as follows:

(a) $C_{16} = 12_{10}$ (c) $D52_{16} = 3410_{10}$ (e) $ABCD_{16} = 43981_{10}$
 (b) $9F_{16} = 159_{10}$ (d) $67E_{16} = 1662_{10}$

- 1.18** Convert the following hexadecimal numbers to their decimal equivalents:

(a) F.4, (b) D3.E, (c) 1111.1, (d) 888.8, (e) EBA.C.

Solution:

Follow the procedure shown in Fig. 1-8b. Refer also to Fig. 1-7. The decimal equivalents of the hexadecimal numbers are as follows:

(a) $F.4_{16} = 15.25_{10}$ (c) $1111.1_{16} = 4369.0625_{10}$ (e) $EBA.C_{16} = 3770.75_{10}$
 (b) $D3.E_{16} = 211.875_{10}$ (d) $888.8_{16} = 2184.5_{10}$

- 1.19** Convert the following whole decimal numbers to their hexadecimal equivalents:

(a) 8, (b) 10, (c) 14, (d) 16, (e) 80, (f) 2560, (g) 3000, (h) 62,500.

Solution:

Follow the procedure shown in Fig. 1-9a. Refer also to Fig. 1-7. The hexadecimal equivalents of the decimal numbers are as follows:

(a) $8_{10} = 8_{16}$ (c) $14_{10} = E_{16}$ (e) $80_{10} = 50_{16}$ (g) $3000_{10} = BB8_{16}$
 (b) $10_{10} = A_{16}$ (d) $16_{10} = 10_{16}$ (f) $2560_{10} = A00_{16}$ (h) $62\,500_{10} = F424_{16}$

- 1.20** Convert the following decimal numbers to their hexadecimal equivalents:

(a) 204.125, (b) 255.875, (c) 631.25, (d) 10 000.003 906 25.

Solution:

Follow the procedure shown in Fig. 1-9b. Refer also to Fig. 1-7. The hexadecimal equivalents of the decimal numbers are as follows:

(a) $204.125_{10} = CC.2_{16}$ (c) $631.25_{10} = 277.4_{16}$
 (b) $255.875_{10} = FF.E_{16}$ (d) $10\,000.003\,906\,25_{10} = 2710.01_{16}$

- 1.21** Convert the following hexadecimal numbers to their binary equivalents:

(a) B, (b) E, (c) 1C, (d) A64, (e) 1F.C, (f) 239.4

Solution:

Follow the procedure shown in Fig. 1-10*a* and *b*. Refer also to Fig. 1-7. The binary equivalents of the hexadecimal numbers are as follows:

$$\begin{array}{lll} (a) \ B_{16} = 1011_2 & (c) \ 1C_{16} = 11100_2 & (e) \ 1F.C_{16} = 11111.11_2 \\ (b) \ E_{16} = 1110_2 & (d) \ A64_{16} = 101001100100_2 & (f) \ 239.4_{16} = 1000111001.01_2 \end{array}$$

1.22 Convert the following binary numbers to their hexadecimal equivalents:

$$\begin{array}{lll} (a) \ 1001.1111 & (c) \ 110101.011001 & (e) \ 10100111.111011 \\ (b) \ 10000001.1101 & (d) \ 10000.1 & (f) \ 1000000.0000111 \end{array}$$

Solution:

Follow the procedure shown in Fig. 1-10*c* and *d*. Refer also to Fig. 1-7. The hexadecimal equivalents of the binary numbers are as follows:

$$\begin{array}{lll} (a) \ 1001.1111_2 = 9.F_{16} & (c) \ 110101.011001_2 = 35.64_{16} & (e) \ 10100111.111011_2 = A7.EC_{16} \\ (b) \ 10000001.1101_2 = 81.D_{16} & (d) \ 10000.1_2 = 10.8_{16} & (f) \ 1000000.0000111_2 = 40.0E_{16} \end{array}$$

1-4 2s COMPLEMENT NUMBERS

The 2s complement method of representing numbers is widely used in microprocessor-based equipment. Until now, we have assumed that all numbers are positive. However, microprocessors must process both positive and negative numbers. By using *2s complement representation*, the *sign as well as the magnitude* of a number can be determined.

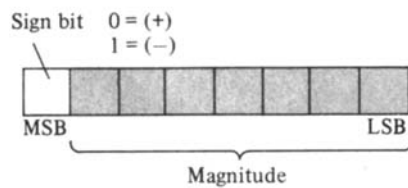
Assume a microprocessor register 8 bits wide such as that shown in Fig. 1-11*a*. The most-significant bit (MSB) is the *sign bit*. If this bit is 0, then the number is (+) positive. However, if the sign bit is 1, then the number is (−) negative. The other 7 bits in this 8-bit register represent the magnitude of the number.

The table in Fig. 1-11*b* shows the 2s complement representations for some positive and negative numbers. For instance, a +127 is represented by the 2s complement number 01111111. A decimal −128 is represented by the 2s complement number 10000000. Note that *the 2s complement representations for all positive values are the same as the binary equivalents* for that decimal number.

Convert the signed decimal −1 to a 2s complement number. Follow Fig. 1-12 as you make the conversion in the next five steps.

- Step 1. Separate the sign and magnitude part of −1. The negative sign means the sign bit will be 1 in the 2s complement representation.
- Step 2. Convert decimal 1 to its 7-bit binary equivalent. In this example decimal 1 equals 0000001 in binary.
- Step 3. Convert binary 0000001 to its 1s complement form. In this example binary 0000001 equals 1111110 in 1s complement. Note that each 0 is changed to a 1 and each 1 to a 0.
- Step 4. Convert the 1s complement to its 2s complement form. In this example 1s complement 1111110 equals 1111111 in 2s complement. Add +1 to the 1s complement to get the 2s complement number.
- Step 5. The 7-bit 2s complement number (1111111 in this example) becomes the magnitude part of the entire 8-bit 2s complement number.

The result is that the signed decimal −1 equals 11111111 in 2s complement notation. The 2s complement number is shown in the register near the top of Fig. 1-12.



(a) The MSB of an 8-bit register is the sign bit

Signed decimal	8-bit 2s complement representation	
+127	0	111 1111
+126	0	111 1110
+125	0	111 1101
+124	0	111 1100
⋮	⋮	⋮
+5	0	000 0101
+4	0	000 0100
+3	0	000 0011
+2	0	000 0010
+1	0	000 0001
+0	0	000 0000
-1	1	111 1111
-2	1	111 1110
-3	1	111 1101
-4	1	111 1100
-5	1	111 1011
⋮	⋮	⋮
-125	1	000 0011
-126	1	000 0010
-127	1	000 0001
-128	1	000 0000
	Sign	Magnitude

same as binary numbers

(b) 2s complement representations of positive and negative numbers

Fig. 1-11

Reverse the process and convert the 2s complement 11111000 to a signed decimal number. Follow Fig. 1-13 as the conversion is made in the following four steps.

- Step 1. Separate the sign bit from the magnitude part of the 2s complement number. The MSB is a 1; therefore, the sign of decimal number will be (–) negative.
- Step 2. Take the 1s complement of the magnitude part. The 7-bit magnitude 1111000 equals 0000111 in 1s complement notation.
- Step 3. Add +1 to the 1s complement number. Adding 0000111 to 1 gives us 0001000. The 7-bit number 0001000 is now in binary.

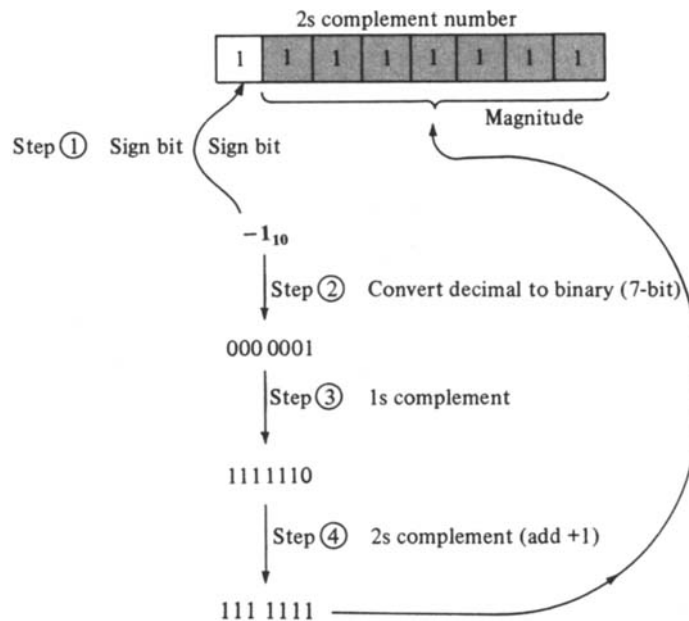


Fig. 1-12 Converting a signed decimal number to a 2s complement number

Step 4. Convert the binary number to its decimal equivalent. In this example, binary 0001000 equals 8 in decimal notation. The magnitude part of the number is 8.

The procedure in Fig. 1-13 shows how to convert 2s complement notation to negative signed decimal numbers. In this example, 2s complement 1111000 equals -8 in decimal notation.

Regular binary-to-decimal conversion (see Fig. 1-4) is used to convert 2s complements that equal positive decimal numbers. Remember that, for positive decimal numbers, the binary and 2s complement equivalents are the same.

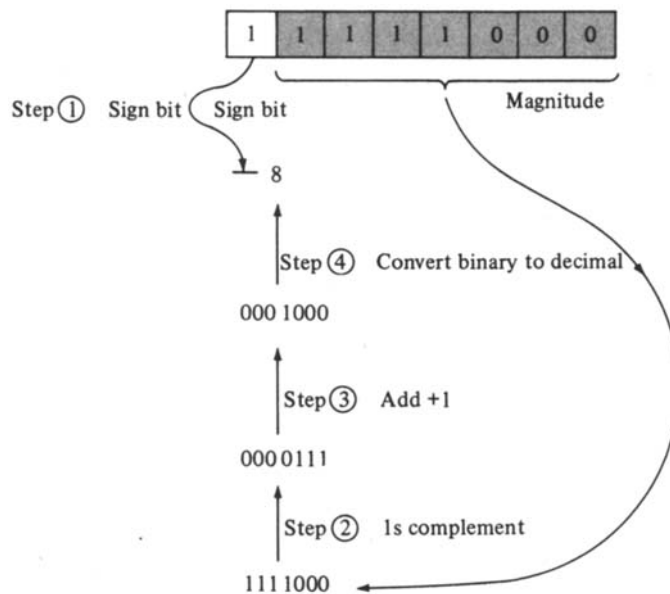


Fig. 1-13 Converting a 2s complement number to a signed decimal number

SOLVED PROBLEMS

- 1.23** The _____ (LSB, MSB) of a 2s complement number is the sign bit.

The MSB (most-significant bit) of a 2s complement number is the sign bit.

- 1.24** The 2s complement number 10000000 is equal to _____ in signed decimal.

Solution:

Follow the procedure shown in Fig. 1-13. The 2s complement number 10000000 equals -128 in decimal.

- 1.25** The number 01110000 is equal to _____ in signed decimal.

Solution:

The 0 in the MSB position means this is a positive number, and conversion to decimal follows the rules used in binary-to-decimal conversion. The number 01110000 is equal to $+112$ in signed decimal.

- 1.26** The signed decimal number $+75$ equals _____ in 8-bit 2s complement.

Solution:

Follow the procedure shown in Fig. 1-4. Decimal $+75$ equals 01001011 in 2s complement and binary.

- 1.27** The 2s complement number 11110001 is equal to _____ in signed decimal.

Solution:

Follow the procedure shown in Fig. 1-13. The 2s complement number 11110001 is equal to -15 in signed decimal.

- 1.28** The signed decimal number -35 equals _____ in 8-bit 2s complement.

Solution:

Follow the procedure shown in Fig. 1-12. Decimal -35 equals 11011101 in 2s complement.

- 1.29** The signed decimal number -100 equals _____ in 8-bit 2s complement.

Solution:

Follow the procedure shown in Fig. 1-12. Decimal -100 equals 10011100 in 2s complement.

- 1.40** The signed decimal number $+20$ equals _____ in 8-bit 2s complement.

Solution:

Follow the procedure shown in Fig. 1-4. Decimal $+20$ equals 00010100 in 2s complement and binary

Supplementary Problems

- 1.31** The number system with a radix of 2 is called the _____ number system. *Ans.* binary
- 1.32** The number system with a radix of 10 is called the _____ number system. *Ans.* decimal
- 1.33** The number system with a radix of 8 is called the _____ number system. *Ans.* octal
- 1.34** The number system with a radix of 16 is called the _____ number system. *Ans.* hexadecimal
- 1.35** A *binary digit* is sometimes shortened and called a(n) _____. *Ans.* bit
- 1.36** How would you pronounce the number 1101 in (a) binary and (b) decimal?
Ans. (a) one, one, zero, one (b) one thousand one hundred and one
- 1.37** The number 1010_2 is a base _____ (a) _____ number and is pronounced _____ (b) _____.
Ans. (a) 2 (b) one, zero, one, zero
- 1.38** Convert the following binary numbers to their decimal equivalents:
 (a) 00001110, (b) 11100000, (c) 10000011, (d) 10011010.
Ans. (a) $00001110_2 = 14_{10}$ (c) $10000011_2 = 131_{10}$
 (b) $11100000_2 = 224_{10}$ (d) $10011010_2 = 154_{10}$
- 1.39** $110011.11_2 = \text{_____}_{10}$ *Ans.* 51.75
- 1.40** $11110000.0011_2 = \text{_____}_{10}$ *Ans.* 240.1875
- 1.41** Convert the following decimal numbers to their binary equivalents:
 (a) 32, (b) 200, (c) 170, (d) 258.
Ans. (a) $32_{10} = 100000_2$ (c) $170_{10} = 10101010_2$
 (b) $200_{10} = 11001000_2$ (d) $258_{10} = 100000010_2$
- 1.42** $40.875_{10} = \text{_____}_2$ *Ans.* 101000.111
- 1.43** $999.125_{10} = \text{_____}_2$ *Ans.* 1111100111.001
- 1.44** Convert the following hexadecimal numbers to their decimal equivalents:
 (a) 13AF, (b) 25E6, (c) B4.C9, (d) 78.D3.
Ans. (a) $13AF_{16} = 5039_{10}$ (c) $B4.C9_{16} = 180.78515_{10}$
 (b) $25E6_{16} = 9702_{10}$ (d) $78.D3_{16} = 120.82421_{10}$
- 1.45** Convert the following decimal numbers to their hexadecimal equivalents:
 (a) 3016, (b) 64881, (c) 17386.75, (d) 9817.625.
Ans. (a) $3016_{10} = BC8_{16}$ (c) $17386.75_{10} = 43EA.C_{16}$
 (b) $64881_{10} = FD71_{16}$ (d) $9817.625_{10} = 2659.A_{16}$
- 1.46** Convert the following hexadecimal numbers to their binary equivalents:
 (a) A6, (b) 19, (c) E5.04, (d) 1B.78.
Ans. (a) $A6_{16} = 10100110_2$ (c) $E5.04_{16} = 11100101.000001_2$
 (b) $19_{16} = 11001_2$ (d) $1B.78_{16} = 11011.01111_2$
- 1.47** Convert the following binary numbers to their hexadecimal equivalents:
 (a) 11110010, (b) 11011001, (c) 111110.000011, (d) 10001.11111.
Ans. (a) $11110010_2 = F2_{16}$ (c) $111110.000011_2 = 3E.0C_{16}$
 (b) $11011001_2 = D9_{16}$ (d) $10001.11111_2 = 11.F8_{16}$

- 1.48** When 2s complement notation is used, the MSB is the _____ bit. *Ans.* sign
- 1.49** Convert the following signed decimal numbers to their 8-bit 2s complement equivalents:
(a) +13, (b) +110, (c) -25, (d) -90.
Ans. (a) 00001101 (b) 01101110 (c) 11100111 (d) 10100110
- 1.50** Convert the following 2s complement numbers to their signed decimal equivalents:
(a) 01110000, (b) 00011111, (c) 11011001, (d) 11001000.
Ans. (a) +112 (b) +31 (c) -39 (d) -56

Chapter 2

Binary Codes

2-1 INTRODUCTION

Digital systems process only codes consisting of 0s and 1s (binary codes). That is due to the bistable nature of digital electronic circuits. The straight binary code was discussed in Chap. 1. Several other, special binary codes have evolved over the years to perform specific functions in digital equipment. All those codes use 0s and 1s, but their meanings may vary. Several binary codes will be detailed here, along with the methods used to translate them into decimal form. In a digital system, electronic translators (called *encoders* and *decoders*) are used for converting from code to code. The following sections will detail the process of conversion from one code to another.

2-2 WEIGHTED BINARY CODES

Straight binary numbers are somewhat difficult for people to understand. For instance, try to convert the binary number 10010110_2 to a decimal number. It turns out that $10010110_2 = 150_{10}$, but it takes quite a lot of time and effort to make this conversion without a calculator.

The *binary-coded decimal* (BCD) *code* makes conversion to decimals much easier. Figure 2-1 shows the 4-bit BCD code for the decimal digits 0-9. Note that the BCD code is a weighted code. The most significant bit has a weight of 8, and the least significant bit has a weight of only 1. This code is more precisely known as the *8421 BCD code*. The 8421 part of the name gives the weighting of each place in the 4-bit code. There are several other BCD codes that have other weights for the four place values. Because the 8421 BCD code is most popular, it is customary to refer to it simply as the BCD code.

Decimal	BCD			
	8s	4s	2s	1s
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1

Fig. 2-1 The 8421 BCD code

How is the decimal number 150 expressed as a BCD number? Figure 2-2a shows the very simple technique for converting decimal numbers to BCD (8421) numbers. Each decimal digit is converted to its 4-bit BCD equivalent (see Fig. 2-1). The decimal number 150 then equals the BCD number 000101010000.

Converting BCD numbers to decimal numbers also is quite simple. Figure 2-2b shows the technique. The BCD number 10010110 is first divided into groups of 4 bits starting at the binary point. Each group of 4 bits is then converted to its equivalent decimal digit, which is recorded below. The BCD number 10010110 then equals decimal 96.

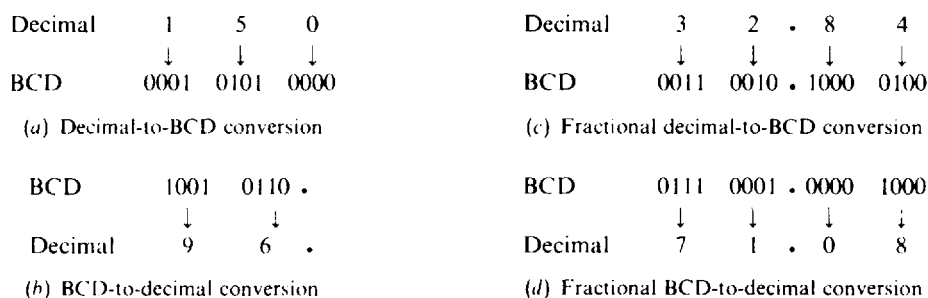


Fig. 2-2

Figure 2-2c illustrates a fractional decimal number being converted to its BCD equivalent. Each decimal digit is converted to its BCD equivalent. The decimal point is dropped down and becomes the binary point. Figure 2-2c shows that decimal 32.84 equals the BCD number 00110010.10000100.

Convert the fractional BCD number 01110001.00001000 to its decimal equivalent. Figure 2-2d shows the procedure. The BCD number is first divided into groups of 4 bits starting at the binary point. Each group of four bits is then converted to its decimal equivalent. The binary point becomes the decimal point in the decimal number. Figure 2-2d shows the BCD number 01110001.00001000 being translated into its decimal equivalent of 71.08.

Consider converting a BCD number to its straight binary equivalent. Figure 2-3 shows the three-step procedure. Step 1 shows the BCD number being divided into 4-bit groups starting from the binary point. Each 4-bit group is translated into its decimal equivalent. Step 1 in Fig. 2-3 shows the BCD number 000100000011.0101 being translated into the decimal number 103.5.

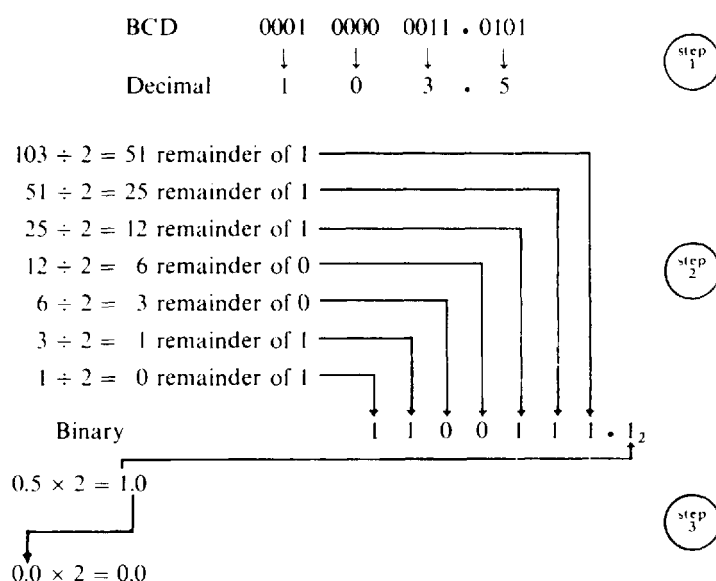


Fig. 2-3 BCD-to-binary conversion

Step 2 in Fig. 2-3 shows the integer part of the decimal number being translated into binary. The 103_{10} is converted into 1100111_2 in step 2 by the repeated divide-by-2 procedure.

Step 3 in Fig. 2-3 illustrates the fractional part of the decimal number being translated into binary. The 0.5_{10} is converted into 0.1_2 in step 3 by the repeated multiply-by-2 procedure. The integer and fractional parts of the binary number are joined. The BCD number 000100000011.0101 then equals the binary number 1100111.1 .

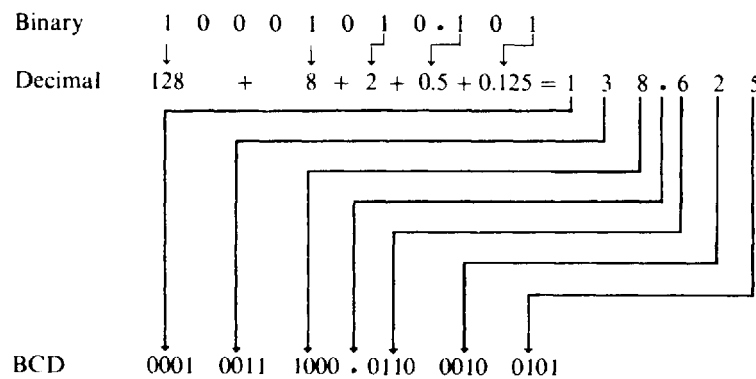


Fig. 2-4 Binary-to-BCD conversion

Note that it is usually more efficient to write down a figure in straight binary numbers than in BCD numbers. Binary numbers usually contain fewer 1s and 0s, as seen in the conversion in Fig. 2-3. Although longer, BCD numbers are used in digital systems when numbers must be easily converted to decimals.

Translate the binary number 10001010.101 into its BCD (8421) equivalent. The procedure is shown in Fig. 2-4. The binary number is first converted to its decimal equivalent. The binary number 10001010.101 then equals 138.625_{10} . Each decimal digit is then translated into its BCD equivalent. Figure 2-4 shows decimal 138.625 being converted into the BCD number 000100111000.011000100101. The entire conversion, then, translates binary 10001010.101_2 into the BCD number 000100111000.011000100101.

“Binary-coded decimal (BCD)” is a general term that may apply to any one of several codes. The most popular BCD code is the 8421 code. The numbers 8, 4, 2, and 1 stand for the weight of each bit in the 4-bit group. Examples of other weighted BCD 4-bit codes are shown in Fig. 2-5.

Decimal	8421 BCD				4221 BCD				5421 BCD							
	8s	4s	2s	1s	8s	4s	2s	1s	4s	2s	2s	1s	5s	4s	2s	1s
0					0	0	0	0								
1					0	0	0	1								
2					0	0	1	0								
3					0	0	1	1								
4					0	1	0	0								
5					0	1	0	1								
6					0	1	1	0								
7					0	1	1	1								
8					1	0	0	0								
9					1	0	0	1								
10	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1
12	0	0	0	1	0	0	1	0	0	0	1	0	0	0	1	0
13	0	0	0	1	0	0	1	1	0	0	1	1	0	0	1	1

Fig. 2-5 Three weighted BCD codes

SOLVED PROBLEMS

- 2.1** The letters BCD stand for _____ - _____ .

Solution:

The letters BCD stand for binary-coded decimal.

- 2.2** Convert the following 8421 BCD numbers to their decimal equivalents:
 (a) 1010 (b) 00010111 (c) 10000110 (d) 010101000011 (e) 00110010.10010100
 (f) 0001000000000000.0101

Solution:

The decimal equivalents of the BCD numbers are as follows:

- | | |
|---------------------------------------|------------------------------------|
| (a) 1010 = ERROR (no such BCD number) | (d) 010101000011 = 543 |
| (b) 00010111 = 17 | (e) 00110010.10010100 = 32.94 |
| (c) 10000110 = 86 | (f) 0001000000000000.0101 = 1000.5 |

- 2.3** Convert the following decimal numbers to their 8421 BCD equivalents:
 (a) 6, (b) 13, (c) 99.9, (d) 872.8, (e) 145.6 (f) 21.001.

Solution:

The BCD equivalents of the decimal numbers are as follows:

- | | | |
|-------------------|-------------------------------|------------------------------------|
| (a) 6 = 0110 | (c) 99.9 = 10011001.1001 | (e) 145.6 = 000101000101.0110 |
| (b) 13 = 00010011 | (d) 872.8 = 100001110010.1000 | (f) 21.001 = 00100001.000000000001 |

- 2.4** Convert the following binary numbers to their 8421 BCD equivalents:
 (a) 10000, (b) 11100.1, (c) 101011.01, (d) 100111.11, (e) 1010.001,
 (f) 1111110001.

Solution:

The BCD equivalents of the binary numbers are as follows:

- | | |
|-----------------------------------|--------------------------------------|
| (a) 10000 = 00010110 | (d) 100111.11 = 00111001.01110101 |
| (b) 11100.1 = 00101000.0101 | (e) 1010.001 = 00010000.000100100101 |
| (c) 101011.01 = 01000011.00100101 | (f) 1111110001 = 0001000000001001 |

- 2.5** Convert the following 8421 BCD numbers to their binary equivalents:
 (a) 00011000 (b) 01001001 (c) 0110.01110101 (d) 00110111.0101
 (e) 01100000.00100101 (f) 0001.001101110101

Solution:

The binary equivalents of the BCD numbers are as follows:

- | | |
|----------------------------|-----------------------------------|
| (a) 00011000 = 10010 | (d) 00110111.0101 = 100101.1 |
| (b) 01001001 = 110001 | (e) 01100000.00100101 = 111100.01 |
| (c) 0110.01110101 = 110.11 | (f) 0001.001101110101 = 1.011 |

- 2.6** List three weighted BCD codes.

Solution:

Three BCD codes are: (a) 8421 BCD code, (b) 4221 BCD code, (c) 5421 BCD code.

2.7 The 4221 BCD equivalent of decimal 98 is _____.

Solution:

The 4221 BCD equivalent of decimal 98 is 11111110.

2.8 The 5421 BCD equivalent of decimal 75 is _____.

Solution:

The 5421 BCD equivalent of decimal 75 is 10101000.

2.9 What kind of number (BCD or binary) would be easier for a worker to translate to decimal?

Solution:

BCD numbers are easiest to translate to their decimal equivalents.

2-3 NONWEIGHTED BINARY CODES

Some binary codes are nonweighted. Each bit therefore has no special weighting. Two such nonweighted codes are the excess-3 and Gray codes.

The *excess-3* (XS3) code is related to the 8421 BCD code because of its binary-coded-decimal nature. In other words, each 4-bit group in the XS3 code equals a specific decimal digit. Figure 2-6 shows the XS3 code along with its 8421 BCD and decimal equivalents. Note that the XS3 number is always *3 more* than the 8421 BCD number.

Decimal	8421 BCD		XS3 BCD	
	10s	1s	10s	1s
0		0000	0011	0011
1		0001	0011	0100
2		0010	0011	0101
3		0011	0011	0110
4		0100	0011	0111
5		0101	0011	1000
6		0110	0011	1001
7		0111	0011	1010
8		1000	0011	1011
9		1001	0011	1100
10	0001	0000	0100	0011
11	0001	0001	0100	0100

Fig. 2-6 The excess-3 (XS3) code

Consider changing the decimal number 62 to an equivalent XS3 number. Step 1 in Fig. 2-7a shows 3 being added to each decimal digit. Step 2 shows 9 and 5 being converted to their 8421 BCD equivalents. The decimal number 62 then equals the BCD XS3 number 10010101.

Convert the 8421 BCD number 01000000 to its XS3 equivalent. Figure 2-7b shows the simple procedure. The BCD number is divided into 4-bit groups starting at the binary point. Step 1 shows 3 (binary 0011) being added to each 4-bit group. The sum is the resulting XS3 number. Figure 2-7b shows the 8421 BCD number 01000000 being converted to its equivalent BCD XS3 number, which is 01110011.

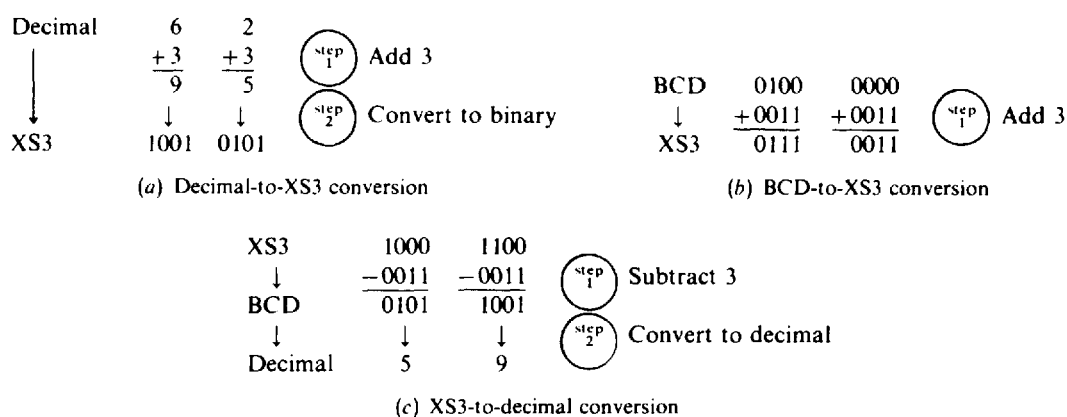


Fig. 2-7

Consider the conversion from XS3 code to decimal. Figure 2-7c shows the XS3 number 10001100 being converted to its decimal equivalent. The XS3 number is divided into 4-bit groups starting at the binary point. Step 1 shows 3 (binary 0011) being subtracted from each 4-bit group. An 8421 BCD number results. Step 2 shows each 4-bit group in the 8421 BCD number being translated into its decimal equivalent. The XS3 number 10001100 is equal to decimal 59 according to the procedure in Fig. 2-7c.

The XS3 code has significant value in arithmetic circuits. The value of the code lies in its ease of complementing. If each bit is complemented (0s to 1s and 1s to 0s), the resulting 4-bit word will be the 9s complement of the number. Adders can use 9s complement numbers to perform subtraction.

The *Gray code* is another nonweighted binary code. The Gray code is not a BCD-type code. Figure 2-8 compares the Gray code with equivalent binary and decimal numbers. Look carefully at the Gray code. Note that each increase in count (increment) is accompanied by *only 1 bit changing state*. Look at the change from the decimal 7 line to the decimal 8 line. In binary all four bits change state (from 0111 to 1000). In this same line the Gray code has only the left bit changing state (0100 to 1100). This change of a single bit in the code group per increment characteristic is important in some applications in digital electronics.

Decimal	Binary	Gray code	Decimal	Binary	Gray code
0	0000	0000	8	1000	1100
1	0001	0001	9	1001	1101
2	0010	0011	10	1010	1111
3	0011	0010	11	1011	1110
4	0100	0110	12	1100	1010
5	0101	0111	13	1101	1011
6	0110	0101	14	1110	1001
7	0111	0100	15	1111	1000

Fig. 2-8 The Gray code

Consider converting a binary number to its Gray code equivalent. Figure 2-9a shows the binary number 0010 being translated into its Gray code equivalent. Start at the MSB of the binary number. Transfer this to the left position in the Gray code as shown by the downward arrow. Now *add* the 8s bit to the next bit over (4s bit). The sum is 0 ($0 + 0 = 0$), which is transferred down and written as the second bit from the left in the Gray code. The 4s bit is now added to the 2s bit of the binary number. The sum is 1 ($0 + 1 = 1$) and is transferred down and written as the third bit from the left in the Gray

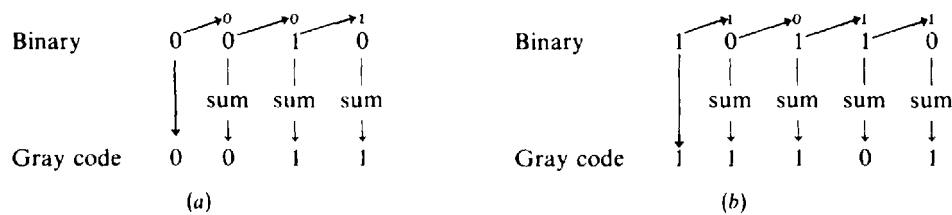


Fig. 2-9 Binary-to-Gray code conversion

code. The 2s bit is now added to the 1s bit of the binary number. The sum is 1 ($1 + 0 = 1$) and is transferred and written as the right bit in the Gray code. The binary 0010 is then equal to the Gray code number 0011. This can be verified in the decimal 2 line of the table in Fig. 2-8.

The rules for converting from any binary number to its equivalent Gray code number are as follows:

1. The left bit is the same in the Gray code as in the binary number.
2. Add the MSB to the bit on its immediate right and record the sum (neglect any carry) below in the Gray code line.
3. Continue adding bits to the bits on their right and recording sums until the LSB is reached.
4. The Gray code number will always have the same number of bits as the binary number.

Try these rules when converting binary 10110 to its Gray code equivalent. Figure 2-9b shows the MSB(1) in the binary number being transferred down and written as part of the Gray code number. The 16s bit is then added to the 8s bit of the binary number. The sum is 1 ($1 + 0 = 1$), which is recorded in the Gray code (second bit from left). Next the 8s bit is added to the 4s bit of the binary number. The sum is 1 ($0 + 1 = 1$), which is recorded in the Gray code (third bit from the left). Next the 4s bit is added to the 2s bit of the binary number. The sum is 0 ($1 + 1 = 10$) because the carry is dropped. The 0 is recorded in the second position from the right in the Gray code. Next the 2s bit is added to the 1s bit of the binary number. The sum is 1 ($1 + 0 = 1$), which is recorded in the Gray code (right bit). The process is complete. Figure 2-9b shows the binary number 10110 being translated into the Gray code number 11101.

Convert the Gray code number 1001 to its equivalent binary number. Figure 2-10a details the procedure. First the left bit (1) is transferred down to the binary line to form the 8s bit. The 8s bit of the binary number is transferred (see arrow) up above the next Gray code bit, and the two are added. The sum is 1 ($1 + 0 = 1$), which is written in the 4s bit place in the binary number. The 4s bit (1) is then added to the next Gray code bit. The sum is 1 ($1 + 0 = 1$). This 1 is written in the 2s place of the binary number. The binary 2s bit (1) is added to the right Gray code bit. The sum is 0 ($1 + 1 = 10$) because the carry is neglected. This 0 is written in the 1s place of the binary number. Figure 2-10a shows the Gray code number 1001 being translated into its equivalent binary number 1110. This conversion can be verified by looking at decimal line 14 in Fig. 2-8.

Convert the 6-bit Gray code number 011011 to its 6-bit binary equivalent. Start at the left and follow the arrows in Fig. 2-10b. Follow the procedure, remembering that a $1 + 1 = 10$. The carry of 1 is neglected, and the 0 is recorded on the binary line. Figure 2-10b shows that the Gray code number 011011 is equal to the binary number 010010.

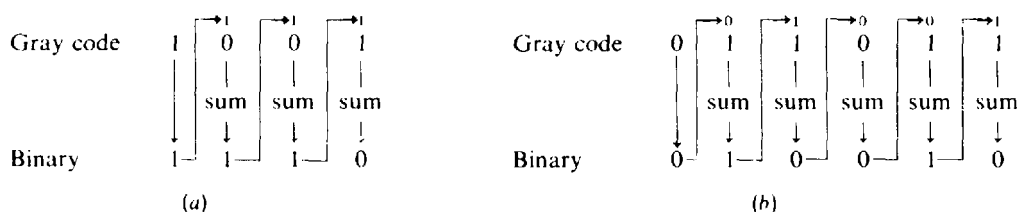


Fig. 2-10 Gray code-to-binary conversions

SOLVED PROBLEMS

2.10 The letters and numbers XS3 stand for _____ – _____ code.

Solution:

XS3 stands for the excess-3 code.

2.11 The _____ (8421, XS3) BCD code is an example of a nonweighted code.

Solution:

The XS3 BCD code is an example of a nonweighted code.

2.12 The _____ (Gray, XS3) code is a BCD code.

Solution:

The XS3 code is a BCD code.

2.13 Convert the following decimal numbers to their XS3 code equivalents:

(a) 9, (b) 18, (c) 37, (d) 42, (e) 650.

Solution:

The XS3 equivalents of the decimal numbers are as follows:

(a) $9 = 1100$ (c) $37 = 01101010$ (e) $650 = 100110000011$
(b) $18 = 01001011$ (d) $42 = 01110101$

2.14 Convert the following 8421 BCD numbers to their XS3 code equivalents:

(a) 0001, (b) 0111, (c) 01100000, (d) 00101001, (e) 10000100.

Solution:

The XS3 equivalents of the 8421 BCD numbers are as follows:

(a) $0001 = 0100$ (c) $01100000 = 10010011$ (e) $10000100 = 10110111$
(b) $0111 = 1010$ (d) $00101001 = 01011100$

2.15 Convert the following XS3 numbers to their decimal equivalents:

(a) 0011, (b) 01100100, (c) 11001011, (d) 10011010, (e) 10000101.

Solution:

The decimal equivalents of the XS3 numbers are as follows:

(a) $0011 = 0$ (c) $11001011 = 98$ (e) $10000101 = 52$
(b) $01100100 = 31$ (d) $10011010 = 67$

2.16 The _____ (Gray, XS3) code is usually used in arithmetic applications in digital circuits.

Solution:

The XS3 code is usually used in arithmetic applications.

2.17 Convert the following straight binary numbers to their Gray code equivalents:

(a) 1010, (b) 10000, (c) 10001, (d) 10010, (e) 10011.

Solution:

The Gray code equivalents of the binary numbers are as follows:

(a) $1010 = 1111$ (c) $10001 = 11001$ (e) $10011 = 11010$
(b) $10000 = 11000$ (d) $10010 = 11011$

2.18 Convert the following Gray code numbers to their straight binary equivalents:

(a) 0100, (b) 11111, (c) 10101, (d) 110011, (e) 011100.

Solution:

The binary equivalents of the Gray code numbers are as follows:

(a) 0100 = 0111 (c) 10101 = 11001 (e) 011100 = 010111

(b) 11111 = 10101 (d) 110011 = 100010

2.19 The Gray code's most important characteristic is that, when the count is incremented by 1, _____ (more than, only) 1 bit will change state.

Solution:

The Gray code's most important characteristic is that, when the count is incremented by 1, only 1 bit will change state.

2-4 ALPHANUMERIC CODES

Binary 0s and 1s have been used to represent various numbers to this point. Bits can also be coded to represent letters of the alphabet, numbers, and punctuation marks. One such 7-bit code is the *American Standard Code for Information Interchange* (ASCII, pronounced "ask-ee"), shown in Fig. 2-11. Note that the letter A is represented by 1000001, whereas B in the ASCII code is 1000010.

Character	ASCII	EBCDIC	Character	ASCII	EBCDIC
Space	010 0000	0100 0000	A	100 0001	1100 0001
!	010 0001	0101 1010	B	100 0010	1100 0010
"	010 0010	0111 1111	C	100 0011	1100 0011
#	010 0011	0111 1011	D	100 0100	1100 0100
\$	010 0100	0101 1011	E	100 0101	1100 0101
%	010 0101	0110 1100	F	100 0110	1100 0110
&	010 0110	0101 0000	G	100 0111	1100 0111
'	010 0111	0111 1101	H	100 1000	1100 1000
(	010 1000	0100 1101	I	100 1001	1100 1001
)	010 1001	0101 1101	J	100 1010	1101 0001
*	010 1010	0101 1100	K	100 1011	1101 0010
+	010 1011	0100 1110	L	100 1100	1101 0011
,	010 1100	0110 1011	M	100 1101	1101 0100
-	010 1101	0110 0000	N	100 1110	1101 0101
.	010 1110	0100 1011	O	100 1111	1101 0110
/	010 1111	0110 0001	P	101 0000	1101 0111
0	011 0000	1111 0000	Q	101 0001	1101 1000
1	011 0001	1111 0001	R	101 0010	1101 1001
2	011 0010	1111 0010	S	101 0011	1110 0010
3	011 0011	1111 0011	T	101 0100	1110 0011
4	011 0100	1111 0100	U	101 0101	1110 0100
5	011 0101	1111 0101	V	101 0110	1110 0101
6	011 0110	1111 0110	W	101 0111	1110 0110
7	011 0111	1111 0111	X	101 1000	1110 0111
8	011 1000	1111 1000	Y	101 1001	1110 1000
9	011 1001	1111 1001	Z	101 1010	1110 1001

Fig. 2-11 Alphanumeric codes

The ASCII code is used extensively in small computer systems to translate from the keyboard characters to computer language. The chart in Fig. 2-11 is not a complete list of all the combinations in the ASCII code.

Codes that can represent both letters and numbers are called *alphanumeric codes*. Another alphanumeric code that is widely used is the *Extended Binary-Coded Decimal Interchange Code* (EBCDIC, pronounced “eb-si-dik”). Part of the EBCDIC code is shown in Fig. 2-11. Note that the EBCDIC code is an 8-bit code and therefore can have more variations and characters than the ASCII code can have. The EBCDIC code is used in many larger computer systems.

The alphanumeric ASCII code is the modern code for getting information into and out of microcomputers. ASCII is used when interfacing computer keyboards, printers, and video displays. ASCII has become the standard input/output code for microcomputers.

Other alphanumeric codes that you may encounter are:

1. 7-bit BCDIC (*Binary-Coded Decimal Interchange Code*).
2. 8-bit EBCDIC (*Extended Binary-Coded Decimal Interchange Code*). Used on some IBM equipment.
3. 7-bit Selectric. Used to control the spinning ball on IBM Selectric typewriters.
4. 12-bit Hollerith. Used on punched paper cards.

SOLVED PROBLEMS

2.20 Binary codes that can represent both numbers and letters are called _____ codes.

Solution:

Alphanumeric codes can represent both numbers and letters.

2.21 The following are abbreviations for what?

(a) ASCII (b) EBCDIC

Solution:

(a) ASCII = American Standard Code for Information Interchange

(b) EBCDIC = Extended Binary-Coded Decimal Interchange Code

2.22 Refer to Fig. 2-12. The ASCII keyboard-encoder output would be _____ if the K on the typewriter-like keyboard were pressed.

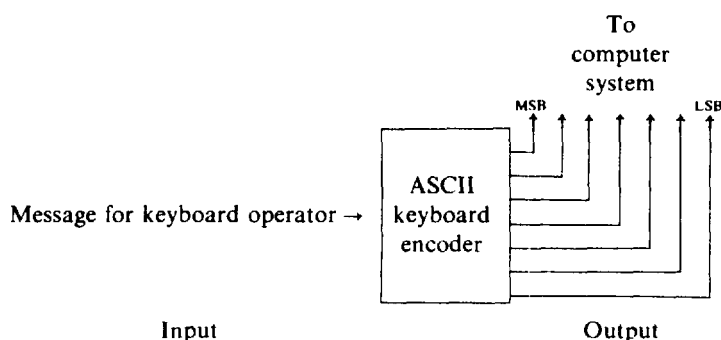


Fig. 2-12 ASCII keyboard-encoder system

Solution:

The ASCII output would be 1001011 if the K on the keyboard were pressed.

- 2.23** Refer to Fig. 2-12. List the 12 ASCII keyboard-encoder outputs for entering the message “pay \$1000.00.”

Solution:

The ASCII codes for the characters in the message are as follows:

- (a) P = 1010000 (d) Space = 0100000 (g) 0 = 0110000 (j) . = 0101110
 (b) A = 1000001 (e) \$ = 0100100 (h) 0 = 0110000 (k) 0 = 0110000
 (c) Y = 1011001 (f) 1 = 0110001 (i) 0 = 0110000 (l) 0 = 0110000

- 2.24** The _____ code is a 12-bit alphanumeric code used on punched paper cards.

Solution:

The 12-bit Hollerith code is used on punched cards.

- 2.25** The 7-bit _____ code is considered the industry standard for input/output on microcomputers.

Solution:

The 7-bit ASCII (American Standard Code for Information Interchange) code is considered the industry standard for microcomputer inputs and outputs.

Supplementary Problems

- 2.26** Electronic devices that translate from one code to another are called (a) or (b).
Ans. (a) encoders (b) decoders
- 2.27** Convert the following 8421 BCD numbers to their decimal equivalents:
 (a) 10010000, (b) 11111111, (c) 0111.0011, (d) 01100001.00000101.
Ans. (a) 10010000 = 90 (c) 0111.0011 = 7.3
 (b) 11111111 = ERROR (no such BCD number) (d) 01100001.00000101 = 61.05
- 2.28** Convert the following decimal numbers to their 8421 BCD equivalents:
 (a) 10, (b) 342, (c) 679.8, (d) 500.6.
Ans. (a) 10 = 00010000 (c) 679.8 = 011001111001.1000
 (b) 342 = 001101000010 (d) 500.6 = 010100000000.0110
- 2.29** Convert the following binary numbers to their 8421 BCD equivalents:
 (a) 10100, (b) 11011.1, (c) 100000.01, (d) 111011.11.
Ans. (a) 10100 = 00100000 (c) 100000.01 = 00110010.00100101
 (b) 11011.1 = 00100111.0101 (d) 111011.11 = 01011001.01110101
- 2.30** Convert the following 8421 BCD numbers to their binary equivalents:
 (a) 01011000, (b) 000100000000, (c) 1001.01110101, (d) 0011.0000011000100101.
Ans. (a) 01011000 = 111010 (c) 1001.01110101 = 1001.11
 (b) 000100000000 = 1100100 (d) 0011.0000011000100101 = 11.0001
- 2.31** The 4221 BCD equivalent of decimal 74 is _____. *Ans.* 11011000
- 2.32** The 5421 BCD equivalent of decimal 3210 is _____. *Ans.* 0011001000010000
- 2.33** The BCD code is convenient when translations must be made to _____ (binary, decimal) numbers.
Ans. decimal

- 2.34** "Excess-3" code is often shortened to _____. *Ans.* XS3
- 2.35** Convert the following decimal numbers to their XS3 code equivalents:
(a) 7, (b) 16, (c) 32, (d) 4089.
Ans. (a) 7 = 1010 (c) 32 = 01100101
(b) 16 = 01001001 (d) 4089 = 0111001110111100
- 2.36** Convert the following XS3 numbers to their decimal equivalents:
(a) 1100, (b) 10101000, (c) 100001110011, (d) 0100101101100101.
Ans. (a) 1100 = 9 (c) 100001110011 = 540
(b) 10101000 = 75 (d) 0100101101100101 = 1832
- 2.37** Convert the following straight binary numbers to their Gray code equivalents:
(a) 0110, (b) 10100, (c) 10101, (d) 10110.
Ans. (a) 0110 = 0101 (b) 10100 = 11110 (c) 10101 = 11111 (d) 10110 = 11101
- 2.38** Convert the following Gray code numbers to their straight binary equivalents:
(a) 0001, (b) 11100, (c) 10100, (d) 10101.
Ans. (a) 0001 = 0001 (b) 11100 = 10111 (c) 10100 = 11000 (d) 10101 = 11001
- 2.39** EBCDIC is a(n) _____-bit alphanumeric code used in some IBM equipment.
Ans. 8
- 2.40** The 7-bit alphanumeric _____ code serves as the industry standard for input/output on micro-computers.
Ans. ASCII