CH = 02

Number Systems, Operations, and Codes

1. **Decimal-to-Binary Conversion**

- **Binary-to-Decimal Conversion**

2. **Binary-to-Hexadecimal Conversion**

- **Hexadecimal-to-Binary Conversion**
- **Hexadecimal-to-Decimal Conversion**

3. **Octal-to-Decimal Conversion**

- **Octal-to-Binary Conversion**
- **Binary-to-Octal Conversion**

Binary Numbers

- The binary number system is another way to represent quantities. It is less complicated than the decimal system because the binary system has only two digits.
- The decimal system with its ten digits is a base-ten system; the binary system with its two digits is a base-two system. The two binary digits (bits) are 1 and 0. The position of a 1 or 0 in a binary number indicates its weight, or value within the number, just as the position of a decimal digit determines the value of that digit. The weights in a binary number are based on powers of two.

Binary weights.

Positive Powers of Two (Whole Numbers)									Negative Powers of Two (Fractional Number)					
2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}
256	128	64	32	16	8	4	2	1	1/2 0.5	1/4 0.25	1/8 0.125	1/16 0.625	1/32 0.03125	1/64 0.015625

Decimal-to-Binary Conversion

Sum-of-Weights Method

EXAMPLE 2-5

Convert the following decimal numbers to binary:

(a) 12 (b) 25

(c) 58 (d) 82

Solution

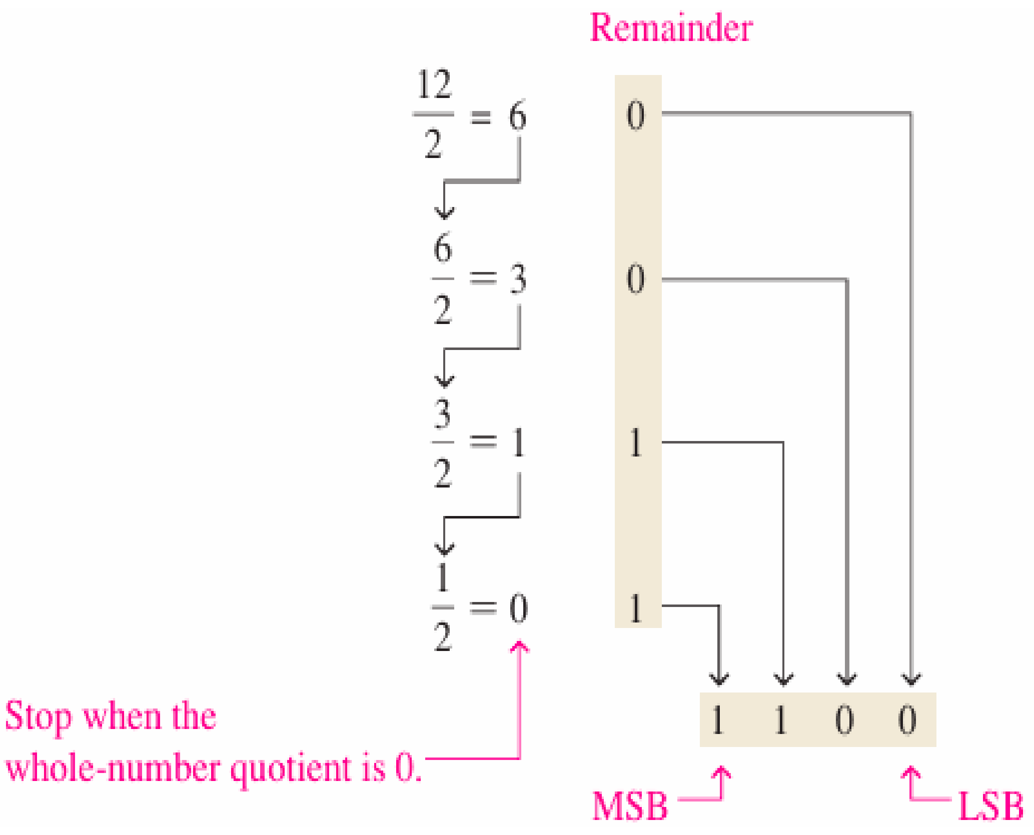
(a) $12 = 8 + 4 = 2^3 + 2^2 \longrightarrow$ **1100**

(b) $25 = 16 + 8 + 1 = 2^4 + 2^3 + 2^0 \longrightarrow$ **11001**

(c) $58 = 32 + 16 + 8 + 2 = 2^5 + 2^4 + 2^3 + 2^1 \longrightarrow$ **111010**

(d) $82 = 64 + 16 + 2 = 2^6 + 2^4 + 2^1 \longrightarrow$ **1010010**

Repeated Division-by-2 Method



Binary-to-Decimal Conversion

Convert the binary whole number 1101101 to decimal.

Solution

Determine the weight of each bit that is a 1, and then find the sum of the weights to get the decimal number.

$$
\begin{array}{rl}
\text{Weight:} & 2^6\ 2^5\ 2^4\ 2^3\ 2^2\ 2^1\ 2^0 \\
\text{Binary number:} & 1\ \ 1\ \ 0\ \ 1\ \ 1\ \ 0\ \ 1
\end{array}
$$

$$
\begin{aligned}
1101101 &= 2^6 + 2^5 + 2^3 + 2^2 + 2^0 \\
&= 64 + 32 + 8 + 4 + 1 = \mathbf{109}
\end{aligned}
$$

Convert the fractional binary number 0.1011 to decimal.

Solution

Determine the weight of each bit that is a 1, and then sum the weights to get the decimal fraction.

$$
\begin{array}{rcccc}
\text{Weight:} & 2^{-1} & 2^{-2} & 2^{-3} & 2^{-4} \\
\text{Binary number:} & 0\,.\,1 & 0 & 1 & 1
\end{array}
$$

$$
\begin{aligned}
0.1011 &= 2^{-1} + 2^{-3} + 2^{-4} \\
&= 0.5 + 0.125 + 0.0625 = \mathbf{0.6875}
\end{aligned}
$$

Binary-to-Hexadecimal Conversion

EXAMPLE 2-24

Convert the following binary numbers to hexadecimal:

(a) 1100101001010111 (b) 111111000101101001

Solution

(a) $\underbrace{1100}_{\text{C}}\underbrace{1010}_{\text{A}}\underbrace{0101}_{5}\underbrace{0111}_{7} = \mathbf{CA57_{16}}$

(b) $\underbrace{0011}_{3}\underbrace{1111}_{\text{F}}\underbrace{0001}_{1}\underbrace{0110}_{6}\underbrace{1001}_{9} = \mathbf{3F169_{16}}$

Two zeros have been added in part (b) to complete a 4-bit group at the left.

Hexadecimal-to-Binary Conversion

EXAMPLE 2-25

Determine the binary numbers for the following hexadecimal numbers:

(a) $10A4_{16}$ (b) $CF8E_{16}$ (c) 9742_{16}

Solution

(a) $\underbrace{1}_{1}\underbrace{0000}_{0}\underbrace{1010}_{\text{A}}\underbrace{0100}_{4}$ → **1000010100100**

(b) $\underbrace{1100}_{\text{C}}\underbrace{1111}_{\text{F}}\underbrace{1000}_{8}\underbrace{1110}_{\text{E}}$ → **1100111110001110**

(c) $\underbrace{1001}_{9}\underbrace{0111}_{7}\underbrace{0100}_{4}\underbrace{0010}_{2}$ → **1001011101000010**

In part (a), the MSB is understood to have three zeros preceding it, thus forming a 4-bit group.

Decimal	Binary	Hexadecimal
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

Hexadecimal-to-Decimal Conversion

Convert the following hexadecimal numbers to decimal:

(a) $1C_{16}$ **(b)** $A85_{16}$

Solution

Remember, convert the hexadecimal number to binary first, then to decimal.

(a) $\overbrace{0001}^{1}\overbrace{1100}^{C} = 2^4 + 2^3 + 2^2 = 16 + 8 + 4 = \mathbf{28}_{10}$

(b) $\overbrace{1010}^{A}\overbrace{1000}^{8}\overbrace{0101}^{5} = 2^{11} + 2^9 + 2^7 + 2^2 + 2^0 = 2048 + 512 + 128 + 4 + 1 = \mathbf{2693}_{10}$

Octal Numbers

Octal-to-Decimal Conversion

Weight: $8^3\ 8^2\ 8^1\ 8^0$

Octal number: 2 3 7 4

$$2374_8 = (2 \times 8^3) + (3 \times 8^2) + (7 \times 8^1) + (4 \times 8^0)$$
$$= (2 \times 512) + (3 \times 64) + (7 \times 8) + (4 \times 1)$$
$$= 1024 + 192 + 56 + 4 = 1276_{10}$$

Octal-to-Binary Conversion

Octal/binary conversion.

Octal Digit	0	1	2	3	4	5	6	7
Binary	000	001	010	011	100	101	110	111

Convert each of the following octal numbers to binary:

(a) 13_8 (b) 25_8 (c) 140_8 (d) 7526_8

Solution

(a) $\underbrace{001}_{1}\underbrace{011}_{3}$ = 001011

(b) $\underbrace{010}_{2}\underbrace{101}_{5}$ = 010101

(c) $\underbrace{001}_{1}\underbrace{100}_{4}\underbrace{000}_{0}$ = 001100000

(d) $\underbrace{111}_{7}\underbrace{101}_{5}\underbrace{010}_{2}\underbrace{110}_{6}$ = 111101010110

Binary-to-Octal Conversion

Convert each of the following binary numbers to octal:

(a) 110101 **(b)** 101111001 **(c)** 100110011010 **(d)** 11010000100

Solution

(a) $\underbrace{110}_{6}\underbrace{101}_{5} = \mathbf{65_8}$

(b) $\underbrace{101}_{5}\underbrace{111}_{7}\underbrace{001}_{1} = \mathbf{571_8}$

(c) $\underbrace{100}_{4}\underbrace{110}_{6}\underbrace{011}_{3}\underbrace{010}_{2} = \mathbf{4632_8}$

(d) $\underbrace{011}_{3}\underbrace{010}_{2}\underbrace{000}_{0}\underbrace{100}_{4} = \mathbf{3204_8}$

Binary Arithmetic

Binary Addition

The four basic rules for adding binary digits (bits) are as follows:

0 + 0 = 0	Sum of 0 with a carry of 0
0 + 1 = 1	Sum of 1 with a carry of 0
1 + 0 = 1	Sum of 1 with a carry of 0
1 + 1 = 10	Sum of 0 with a carry of 1

1 + 0 + 0 = 01	Sum of 1 with a carry of 0
1 + 1 + 0 = 10	Sum of 0 with a carry of 1
1 + 0 + 1 = 10	Sum of 0 with a carry of 1
1 + 1 + 1 = 11	Sum of 1 with a carry of 1

Add the following binary numbers:

(a) 11 + 11 **(b)** 100 + 10

(c) 111 + 11 **(d)** 110 + 100

Solution

The equivalent decimal addition is also shown for reference.

(a)
$$\begin{array}{r} 11 \\ +\,11 \\ \hline \mathbf{110} \end{array} \qquad \begin{array}{r} 3 \\ +\,3 \\ \hline 6 \end{array}$$

(b)
$$\begin{array}{r} 100 \\ +\,10 \\ \hline \mathbf{110} \end{array} \qquad \begin{array}{r} 4 \\ +\,2 \\ \hline 6 \end{array}$$

(c)
$$\begin{array}{r} 111 \\ +\,11 \\ \hline \mathbf{1010} \end{array} \qquad \begin{array}{r} 7 \\ +\,3 \\ \hline 10 \end{array}$$

(d)
$$\begin{array}{r} 110 \\ +\,100 \\ \hline \mathbf{1010} \end{array} \qquad \begin{array}{r} 6 \\ +\,4 \\ \hline 10 \end{array}$$

Perform the following binary subtractions:

(a) 11 − 01 **(b)** 11 − 10

Solution

(a)
$$\begin{array}{r} 11 \\ -01 \\ \hline \mathbf{10} \end{array} \qquad \begin{array}{r} 3 \\ -1 \\ \hline 2 \end{array}$$

(b)
$$\begin{array}{r} 11 \\ -10 \\ \hline \mathbf{01} \end{array} \qquad \begin{array}{r} 3 \\ -2 \\ \hline 1 \end{array}$$

No borrows were required in this example. The binary number 01 is the same as 1.

Related Problem

Subtract 100 from 111.

Binary Subtraction

- *The four basic rules for subtracting bits are as follows:*

$0 - 0 = 0$

$1 - 1 = 0$

$1 - 0 = 1$

$10 - 1 = 1 \qquad 0 - 1 \text{ with a borrow of } 1$

In binary $10 - 1 = 1$, not 9.

- When subtracting numbers, you sometimes have to borrow from the next column to the left. A borrow is required in binary only when you try to subtract a 1 from a 0.
- In this case, when a 1 is borrowed from the next column to the left, a 10 is created in the column being subtracted, and the last of the four basic rules just listed must be applied.

Perform the following binary subtractions:

(a) 11 − 01 **(b)** 11 − 10

Solution

(a)

$$\begin{array}{r} 11 \\ -01 \\ \hline \mathbf{10} \end{array} \qquad \begin{array}{r} 3 \\ -1 \\ \hline 2 \end{array}$$

(b)

$$\begin{array}{r} 11 \\ -10 \\ \hline \mathbf{01} \end{array} \qquad \begin{array}{r} 3 \\ -2 \\ \hline 1 \end{array}$$

No borrows were required in this example. The binary number 01 is the same as 1.

Related Problem

Subtract 100 from 111.

Binary Multiplication

The four basic rules for multiplying bits are as follows:

$$0 \times 0 = 0$$
$$0 \times 1 = 0$$
$$1 \times 0 = 0$$
$$1 \times 1 = 1$$

Binary multiplication of two bits is the same as multiplication of the decimal digits 0 and 1.

Perform the following binary multiplications:

(a) 11 × 11 **(b)** 101 × 111

Solution

(a)

$$\begin{array}{rr} & 11 \\ & \times\ 11 \\ \hline \text{Partial products}\left\{\begin{array}{r} \\ \end{array}\right. & \begin{array}{r} 11 \\ +11 \end{array} \\ \hline & \mathbf{1001} \end{array} \qquad \begin{array}{r} 3 \\ \times\ 3 \\ \hline 9 \end{array}$$

(b)

$$\begin{array}{rr} & 111 \\ & \times\ 101 \\ \hline \text{Partial products}\left\{\begin{array}{r} \\ \\ \\ \end{array}\right. & \begin{array}{r} 111 \\ 000 \\ +111\ \ \end{array} \\ \hline & \mathbf{100011} \end{array} \qquad \begin{array}{r} 7 \\ \times\ 5 \\ \hline 35 \end{array}$$

Related Problem

Multiply 1101 × 1010.

Binary Division

Perform the following binary divisions:

(a) $110 \div 11$ (b) $110 \div 10$

Solution

(a)
$$\begin{array}{r} \mathbf{10} \\ 11\overline{)110} \\ \underline{11} \\ 000 \end{array} \qquad \begin{array}{r} 2 \\ 3\overline{)6} \\ \underline{6} \\ 0 \end{array}$$

(b)
$$\begin{array}{r} \mathbf{11} \\ 10\overline{)110} \\ \underline{10} \\ 10 \\ \underline{10} \\ 00 \end{array} \qquad \begin{array}{r} 3 \\ 2\overline{)6} \\ \underline{6} \\ 0 \end{array}$$

Related Problem

Divide 1100 by 100.

1. Perform the following binary additions:

   (a) $1101 + 1010$ (b) $10111 + 01101$

2. Perform the following binary subtractions:

   (a) $1101 - 0100$ (b) $1001 - 0111$

3. Perform the indicated binary operations:

   (a) 110×111 (b) $1100 \div 011$

Data Representation using Signed Magnitude

- Data Representation
  - Magnitude
    - Unsigned: $\oplus^{ve}$ binary no.
    - Signed: both $\oplus^{ve}$ & $\ominus^{ve}$
  - Complement
    - 1's Complement: $\oplus^{ve}$ & $\ominus^{ve}$
    - 2's Complement: $\oplus^{ve}$ & $\ominus^{ve}$

→ In all 4 rep. $\oplus^{ve}$ no. is rep. in same way.

Unsigned Mag:-

+6 = 110

−6 = can't rep.

Signed mag. :-

Sign bit → msb

+6 = 0 110 → magnitude

−6 = 1 110

imp

sign bit = 0 → no. is $\oplus^{ve}$

— " — = 1 → — " — $\ominus^{ve}$

$\underline{+}\,\underline{13}$ 01101

$-\underline{13}$ 11101

n → no. of var.

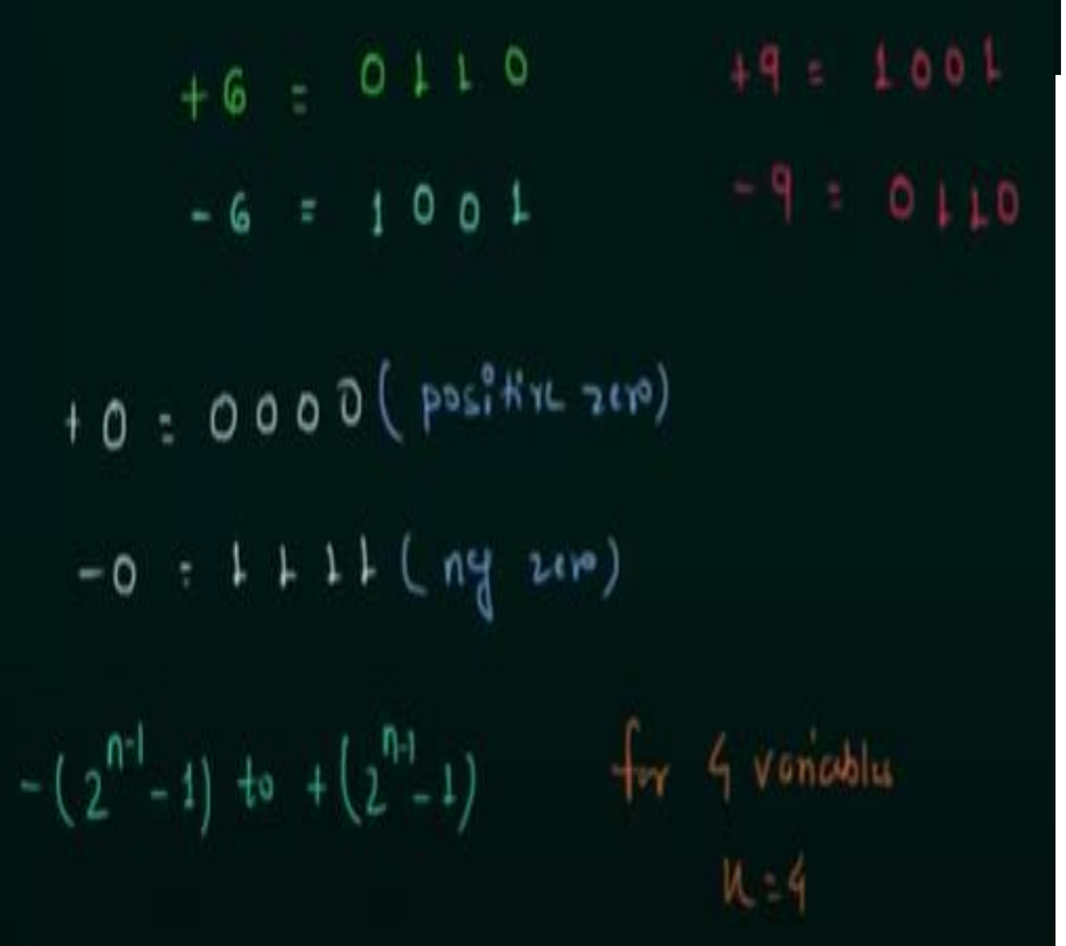
Range : $-(2^{n-1}-1)$ to $+(2^{n-1}-1)$

n : 4

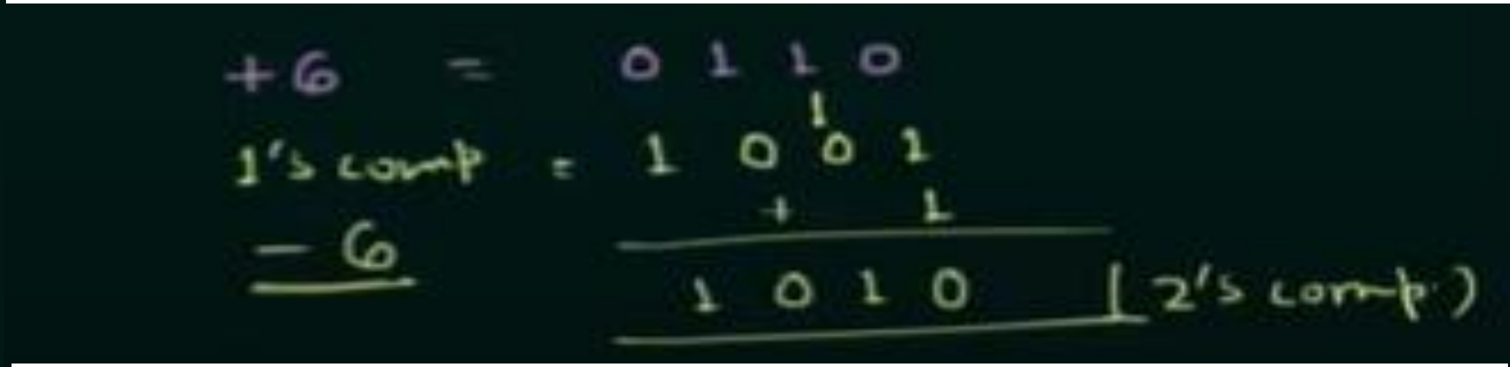
$-(2^3-1)$ to $+(2^3-1)$

−7 to +7

1's Complement



2's Complement



Express the decimal number −39 as an 8-bit number in the sign-magnitude, 1's complement, and 2's complement forms.

Solution

First, write the 8-bit number for +39.

00100111

In the *sign-magnitude form,* −39 is produced by changing the sign bit to a 1 and leaving the magnitude bits as they are. The number is

10100111

In the *1's complement form,* −39 is produced by taking the 1's complement of +39 (00100111).

11011000

In the *2's complement form,* −39 is produced by taking the 2's complement of +39 (00100111) as follows:

	11011000	1's complement
+	1	
	11011001	2's complement

Signed Numbers

- Digital systems, such as the computer, must be able to handle both positive and negative numbers.
- A signed binary number consists of both sign and magnitude information.
- The sign indicates whether a number is positive or negative, and the magnitude is the value of the number.
- There are three forms in which signed integer (whole) numbers can be represented in binary.

a) sign-magnitude

b) 1’s complement

c) and 2’s complement.

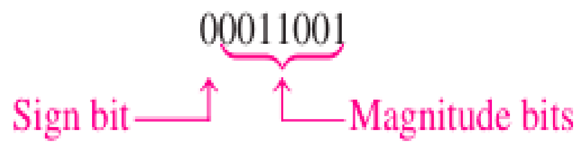
- Of these, the 2’s complement is the most important and the sign-magnitude is the least used.

The Sign Bit

- The left-most bit in a signed binary number is the sign bit, which tells you whether the number is positive or negative.
- A 0 sign bit indicates a positive number, and a 1 sign bit indicates a negative number.

Sign-Magnitude Form

When a signed binary number is represented in sign-magnitude, the left-most bit is the sign bit and the remaining bits are the magnitude bits. The magnitude bits are in true (uncomplemented) binary for both positive and negative numbers. For example, the decimal number +25 is expressed as an 8-bit signed binary number using the sign-magnitude form as



The decimal number 25 is expressed as

− 25 = 10011001

Notice that the only difference between +25 and − 25 is the sign bit because the magnitude bits are in true binary for both positive and negative numbers.

In the sign-magnitude form, a negative number has the same magnitude bits as the corresponding positive number but the sign bit is a 1 rather than a zero.

1’s Complement Form

- In the 1’s complement form, a negative number is the 1’s complement of the corresponding positive number.
- For example, using eight bits, the decimal number 25 is expressed as the 1’s complement of +25 (00011001) as

  11100110

2’s Complement Form

Again, using eight bits, let’s take decimal number 25 and express it as the 2’s complement of +25 (00011001).
Inverting each bit and adding 1, you get -25=11100111 In the 2’s complement form, a negative number is the 2’s complement of the corresponding positive number.

An alternative method of finding the 2’s complement of a binary number is as follows:

1. Start at the right with the LSB and write the bits as they are up to and including the first 1.
2. Take the 1’s complements of the remaining bits.

Find the 2's complement of 10110010.

Solution

10110010	Binary number
01001101	1's complement
+ 1	Add 1
01001110	2's complement

Find the 2's complement of 10111000 using the alternative method.

Solution

10111000	Binary number
01001000	2's complement

$$\underbrace{\mathbf{0100}}_{\text{1's complements of original bits}}\underbrace{\mathbf{1000}}_{\text{These bits stay the same.}}$$

Digital Codes

Digital codes are patterns of 0s and 1s used to represent numbers, characters, and symbols.

(a) Gray Code

- Each successive number differs by only **one bit**.
- Used in **rotary encoders** and error reduction in analog-to-digital converters.
- Example (4-bit Gray vs Binary):

Decimal	**Binary**	**Gray**
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100
8		
9		
10		
11		
12		

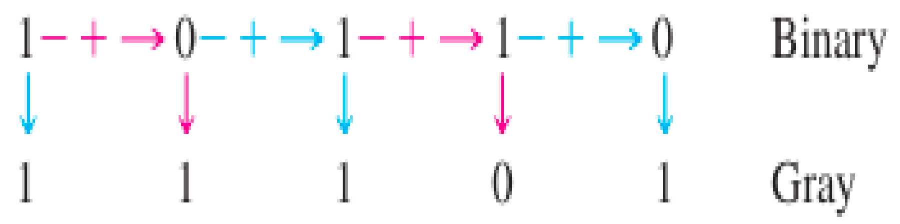
Analogy: Like climbing stairs **one step at a time** (no big jumps), to avoid tripping.

Binary-to-Gray Code Conversion

Conversion between binary code and Gray code is sometimes useful. The following rules explain how to convert from a binary number to a Gray code word:

1. The most significant bit (left-most) in the Gray code is the same as the corresponding MSB in the binary number.
2. Going from left to right, add each adjacent pair of binary code bits to get the next Gray code bit. Discard carries.

For example, the conversion of the binary number 10110 to Gray code is as follows:

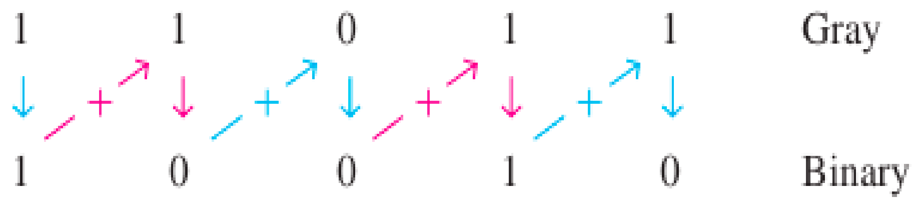


The Gray code is 11101.

Gray-to-Binary Code Conversion

- 1. The most significant bit (left-most) in the binary code is the same as the corresponding bit in the Gray code.
- 2. Add each binary code bit generated to the Gray code bit in the next adjacent position. Discard carries.

For example, the conversion of the Gray code word 11011 to binary is as follows:



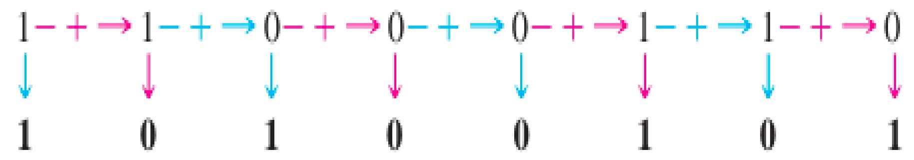
The binary number is 10010.

(a) Convert the binary number 11000110 to Gray code.

(b) Convert the Gray code 10101111 to binary.

Solution

(a) Binary to Gray code:



(b) Gray code to binary: