


Al-Mustaqbal University
College of Science

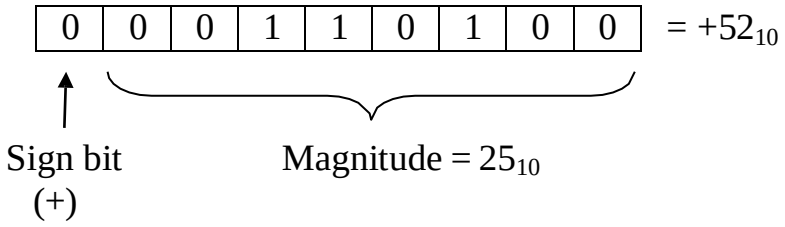
Artifical Intelligence Sciences Department

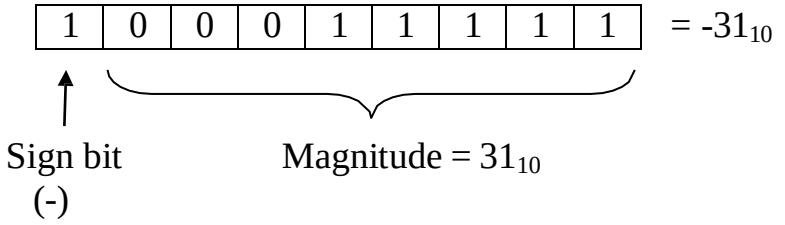
Lecture 6
Computer Organization and Logic Design

MSC.Fatima Ayad

Representation Of Signed Numbers :-

In digital computers, the binary numbers are represented by a set of binary storage device (usually flip-flops). Each device represents one bit. For example, a 8-bit FF register could store binary numbers ranging from 0000 0000 to 1111 1111 (**0** to **255**). This represents the magnitude of the number. Since most digital computers calculators handle negative as well as positive numbers, Some means of representing the sign of the number (+ or -). This is usually done by adding another bit as shown below.



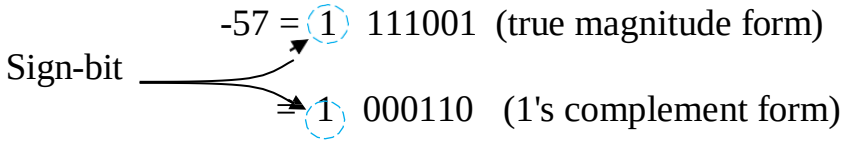


Although this true-magnitude system is straight forward and easy to understand, it is not as useful as two other systems. They are:-

1) 1's complement form.
2) 2's complement form.

1'st complement form :-

The 1's complement form of any binary number is obtained by changing each bit to it's complement for example :



Examples:-

-14 = 10001
-326 = 1010111001
-7 = 1000

2's complement form :-

The 2's complement form of a binary number is formed simply by taking the 1's complement of the number and adding 1 to the least-significant bit position. The procedure is illustrated below.

1	1	1	0	0	1	(decimal 57)
0	0	0	1	1	0	(1's complement)
				+	1	(add 1 to LSB)
0	0	0	1	1	1	(2's complement)

The three systems of representing signed numbers are summarized below:-

True Magnitude System	1's complement system	2's complement system
+57 = 0 111001 -57 = 1 111001	0 000110 1 000110	0 000111 1 000111

Notes:-

1) In al three systems the representation of positive numbers is the same.
2) While all three systems have been used in the past, most modern computers use the 2's complement system.

H.W

a) Represent each of the following signed decimal numbers as a signed binary decimal number in the 2's complement system. Use a total of five bits including the sign bit :

1) +13 2) -9 3) +3 4) -2 5) -8.

Binary-coded-decimal code :-

If each digit of a decimal number is represented by its binary equivalent, this produces a code called binary-coded-decimal (BCD). Since a decimal digit can be as large as 9, 4-bits are required to code each digit. The table below shows each decimal digit and its binary equivalent.

Decimal	0	1	2	3	4	5	6	7	8	9
BCD	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001

example: The BCD of decimal 874

8	7	4
1000	0111	0100

Excess-3-code :-

It is performed in the same manner as BCD except that 3 is added to each decimal digit before encoding it in binary. The following table shows this code.

Decimal	0	1	2	3	4	5	6	7	8	9
Ex-3-code	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100

Gray code

Binary to Gray code conversion

There are two steps in conversion from binary to gray code

1- The most significant bit (left-most) in the Gray code is the same as the corresponding MSB in the binary.

2- Going from left to right, add each adjacent pair of binary code bits to get the next Gray code bit. Discard the carries.

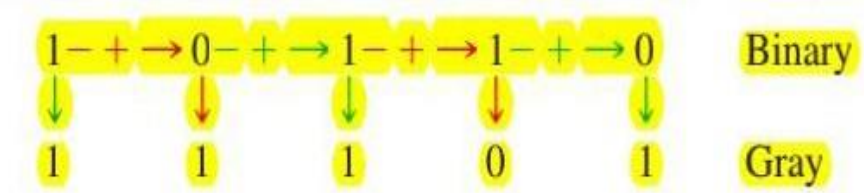
Example 1

(a) Convert the binary number 10110 to Gray code

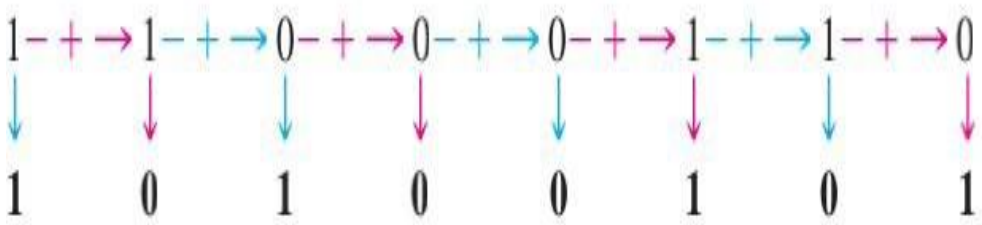
(b) Convert the binary number 11000110 to Gray code

Solution

(a)



(b)



Gray to binary code conversion

There are two steps in conversion from gray code to binary

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.

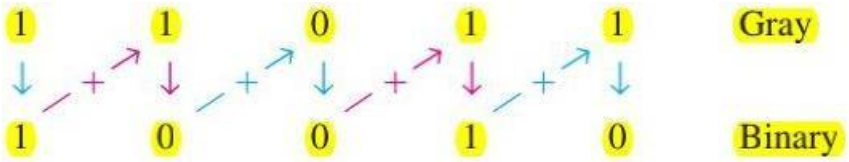
Example

(a) Convert the Gray code 11011 to binary

(b) Convert the Gray code 10101111 to binary

Solution

(a)



(b)

Gray code to binary: