Complements

Complements are used in digital computers to simplify the subtraction operation and for logical manipulations.

One's complement representation

In a binary number, if each 1 is replaced by 0 and each 0 by 1, the resulting number is known as the one's complement of the first number. It is calculated by subtracting each digit from 1.

eg:- 1's complement of 1011000 is 0100111

Two's complement representation

If 1 is added to 1's complement of a binary number, the resulting number is known as the 2's complement of the binary number. It can also be obtained by leaving all least significant 0's and the first 1 unchanged and replacing 1's with 0's and 0's with 1's in all other higher significant digits.

eg:- 2's complement of 1011000 is 0101000

If MSB is 0, the number is positive, whereas if the MSB is 1 the number is negative.

Subtraction using 2's complement

Binary subtraction can be performed by adding 2's complement of the subtrahend to the minuend.

(1) If a final carry is generated, discard the carry and the answer is given by the remaining bits which is positive

(2) If the final carry is 0, the answer is negative and is in 2's complement form.

2's complement of 2's complement of a binary number is the number itself.

1) Given the two binary numbers X = 1010100 and Y = 1000011, perform the subtraction (a) X−Y and (b) Y−X using 2's complement.

(a)
$$
\begin{array}{lr}
X = & 1010100 \\
\text{2's comp. of } Y = & +0111101 \\
\hline
 & 10010001 \\
\text{Discard end carry} & -10000000 \\
\hline
 & 0010001
\end{array}
$$

$\therefore X - Y = 0010001$

(b)
$$
\begin{array}{lr}
Y = & 1000011 \\
\text{2's comp. of } X = & +0101100 \\
\hline
 & 1101111
\end{array}
$$

There is no end carry. Therefore, the answer is −(2's complement of 1101111) = −(0010001) Ans

2) Repeat using 1's complement.

(a)
$$
\begin{array}{lr}
X = & 1010100 \\
\text{1's comp. of } Y = & +0111100 \\
\hline
 & 10010000 \\
\text{End-around carry} & +1 \\
\hline
 & 0010001
\end{array}
$$

(b)
$$
\begin{array}{lr}
Y = & 1000011 \\
\text{1's comp of } + X = & +0101011 \\
\hline
 & 1101110
\end{array}
$$

No end carry. Therefore, the answer is −(1's complement of 1101110) = −(0010001) Ans

BCD addition

$$\begin{array}{r} 4 \\ +\ 5 \\ \hline 9 \end{array} \qquad \begin{array}{r} 0100 \\ +\ 0101 \\ \hline 1001 \end{array} \qquad \begin{array}{r} 4 \\ +\ 8 \\ \hline 12 \end{array} \qquad \begin{array}{rr} & 0100 \\ + & 1000 \\ \hline & 1100 \\ + & 0110\ (6) \\ \hline 0001 & 0010 \\ 1 & 2 \end{array}$$

The addition of $6 = (0110)_2$ to the binary sum converts it to the correct digit and also produces a carry as required. This is because a carry in the most significant bit position of the binary sum and a decimal differ by $16 - 10 = 6$.

$$\begin{array}{lr} \text{BCD} & 184 \\ + & 576 \end{array} \qquad \begin{array}{rrrr} & 0001 & 1000 & 0100 \\ + & 0101 & 0111 & 0110 \\ \hline & 0111 & 10000 & 1010 \\ & & 0110 & 0110 \\ \hline & 0111 & 0110 & 0000 \\ & 7 & 6 & 0 \end{array}$$

$$\begin{array}{r} 5\ 3\ 2 \\ +\ 2\ 8\ 9 \\ \hline 8\ 2\ 1 \end{array} \qquad \begin{array}{rrrr} & 0101 & 0011 & 0010 \\ + & 0010 & 1000 & 1001 \\ \hline & 1000 & 1100 & 1011 \\ & & 0110 & 0110 \\ \hline & 1000 & 0010 & 0001 \\ & 8 & 2 & 1 \end{array}$$

$$\begin{array}{r} 5\ 3\ 2 \\ +\ 6\ 8\ 9 \end{array} \qquad \begin{array}{rrrrr} & & 0101 & 0011 & 0010 \\ + & & 0110 & 1000 & 1001 \\ \hline & & 1100 & 1100 & 1011 \\ & & 0110 & 0110 & 0110 \\ \hline & 0001 & 0010 & 0010 & 0001 \\ & 1 & 2 & 2 & 1 \end{array}$$

GRAY CODE OR REFLECTED CODE

The gray code is unweighted and is not an arithmetic code. There are no specific weights assigned to the bit positions. The important feature of the gray code is that it exhibits only a single bit change from one codeword to the next in sequence.

The gray code is a reflected code and is constructed as follows :-

(i) A 1-bit gray code has two code words 0 and 1 representing decimal numbers 0 and 1 respectively.

(ii) An n-bit ($n \geq 2$) gray code will have first 2^{n-1} gray codes of $(n-1)$-bits written in order with a leading 0 appended.

(iii) The last 2^{n-1} gray codes will be equal to the gray codes of an $(n-1)$-bit gray code, written in reverse order (assuming a mirror placed between first 2^{n-1} and last 2^{n-1} gray codes) with a leading 1 appended.

Binary to Gray code conversion

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:-

$$\begin{array}{ccccccccc} 1 & \xrightarrow{+} & 0 & \xrightarrow{+} & 1 & \xrightarrow{+} & 1 & \xrightarrow{+} & 0 \\ \downarrow & & \downarrow & & \downarrow & & \downarrow & & \downarrow \\ 1 & & 1 & & 1 & & 0 & & 1 \end{array}$$

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 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 11101 to binary is as follows:-

$$\begin{array}{ccccc} 1 & 1 & 1 & 0 & 1 \\ \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ 1 & 0 & 1 & 1 & 0 \end{array}$$

Thus, the binary is 10110

EXCESS-3 CODE

This is another form of BCD code, in which each decimal digit is coded into a 4-bit binary code. The code for each decimal digit is obtained by adding decimal 3 to the natural BCD code of the digit.

For example :- Decimal 2 is coded as 0010 + 0011 = 0101

ALPHANUMERIC CODES

In many applications of digital systems, it is required that they handle data that may consist of numerals, letters and special symbols. An alphanumeric (sometimes abbreviated alphameric) code is a binary code of a group of elements consisting of the ten decimal digits, the 26 letters of the alphabet, and a certain number of special symbols such as $. The total number of elements in an alphanumeric group is greater than 36. Therefore, it must be coded with a minimum of six bits ($2^6 = 64$). One possible 6-bit code used in many computers to represent alphanumeric characters and symbols internally is internal code.