Binary Codes

In the coding, when numbers, letters or words are represented by a specific group of symbols, it is said that the number, letter or word is being encoded. The group of symbols is called as a code. The digital data is represented, stored and transmitted as group of binary bits. This group is also called as **binary code**. The binary code is represented by the number as well as alphanumeric letter.

Advantages of Binary Code

Following is the list of advantages that binary code offers.

* Binary codes are suitable for the computer applications.
* Binary codes are suitable for the digital communications.
* Binary codes make the analysis and designing of digital circuits if we use the binary codes.
* Since only 0 & 1 are being used, implementation becomes easy.

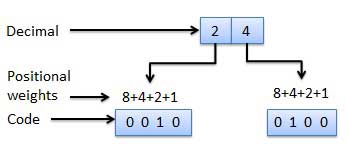
Classification of binary codes

The codes are broadly categorized into following four categories.

* Weighted Codes
* Non-Weighted Codes
* Binary Coded Decimal Code
* Alphanumeric Codes
* Error Detecting Codes
* Error Correcting Codes

Weighted Codes

Weighted binary codes are those binary codes which obey the positional weight principle. Each position of the number represents a specific weight. Several systems of the codes are used to express the decimal digits 0 through 9. In these codes each decimal digit is represented by a group of four bits.

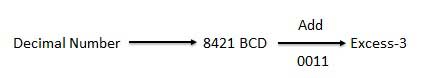


Non-Weighted Codes

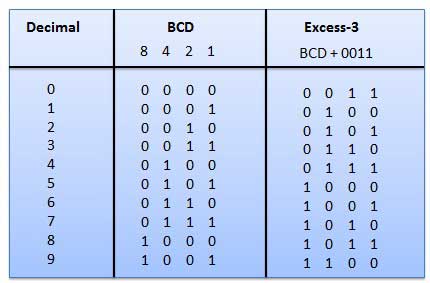
In this type of binary codes, the positional weights are not assigned. The examples of non-weighted codes are Excess-3 code and Gray code.

Excess-3 code

The Excess-3 code is also called as XS-3 code. It is non-weighted code used to express decimal numbers. The Excess-3 code words are derived from the 8421 BCD code words adding (0011)2 or (3)10 to each code word in 8421. The excess-3 codes are obtained as follows −

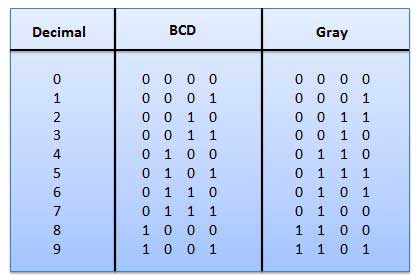


Example



Gray Code

It is the non-weighted code and it is not arithmetic codes. That means there are no specific weights assigned to the bit position. It has a very special feature that, only one bit will change each time the decimal number is incremented as shown in fig. As only one bit changes at a time, the gray code is called as a unit distance code. The gray code is a cyclic code. Gray code cannot be used for arithmetic operation.

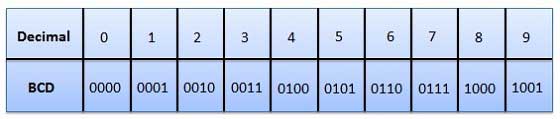


Application of Gray code

* Gray code is popularly used in the shaft position encoders.
* A shaft position encoder produces a code word which represents the angular position of the shaft.

Binary Coded Decimal (BCD) code

In this code each decimal digit is represented by a 4-bit binary number. BCD is a way to express each of the decimal digits with a binary code. In the BCD, with four bits we can represent sixteen numbers (0000 to 1111). But in BCD code only first ten of these are used (0000 to 1001). The remaining six code combinations i.e. 1010 to 1111 are invalid in BCD.



Advantages of BCD Codes

* It is very similar to decimal system.
* We need to remember binary equivalent of decimal numbers 0 to 9 only.

Disadvantages of BCD Codes

* The addition and subtraction of BCD have different rules.
* The BCD arithmetic is little more complicated.
* BCD needs more number of bits than binary to represent the decimal number. So BCD is less efficient than binary.

Alphanumeric codes

A binary digit or bit can represent only two symbols as it has only two states '0' or '1'. But this is not enough for communication between two computers because there we need many more symbols for communication. These symbols are required to represent 26 alphabets with capital and small letters, numbers from 0 to 9, punctuation marks and other symbols.

The alphanumeric codes are the codes that represent numbers and alphabetic characters. Mostly such codes also represent other characters such as symbol and various instructions necessary for conveying information. An alphanumeric code should at least represent 10 digits and 26 letters of alphabet i.e. total 36 items. The following three alphanumeric codes are very commonly used for the data representation.

* American Standard Code for Information Interchange (ASCII).
* Extended Binary Coded Decimal Interchange Code (EBCDIC).
* Five bit Baudot Code.

ASCII code is a 7-bit code whereas EBCDIC is an 8-bit code. ASCII code is more commonly used worldwide while EBCDIC is used primarily in large IBM computers.

Error Codes

There are binary code techniques available to detect and correct data during data transmission.

|  |  |
| --- | --- |
| **Error Code** | **Description** |
| [Error Detection and Correction](https://www.tutorialspoint.com/computer_logical_organization/error_codes.htm) | Error detection and correction code techniques |

Codes Conversion

Advertisements

[Previous Page](https://www.tutorialspoint.com/computer_logical_organization/binary_codes.htm)

[Next Page](https://www.tutorialspoint.com/computer_logical_organization/complement_arithmetic.htm)

There are many methods or techniques which can be used to convert code from one format to another. We'll demonstrate here the following

* Binary to BCD Conversion
* BCD to Binary Conversion
* BCD to Excess-3
* Excess-3 to BCD

Binary to BCD Conversion

Steps

* **Step 1** -- Convert the binary number to decimal.
* **Step 2** -- Convert decimal number to BCD.

Example − convert (11101)2 to BCD.

Step 1 − Convert to Decimal

Binary Number − 111012

Calculating Decimal Equivalent −

|  |  |  |
| --- | --- | --- |
| **Step** | **Binary Number** | **Decimal Number** |
| Step 1 | 111012 | ((1 × 24) + (1 × 23) + (1 × 22) + (0 × 21) + (1 × 20))10 |
| Step 2 | 111012 | (16 + 8 + 4 + 0 + 1)10 |
| Step 3 | 111012 | 2910 |

Binary Number − 111012 = Decimal Number − 2910

Step 2 − Convert to BCD

Decimal Number − 2910

Calculating BCD Equivalent. Convert each digit into groups of four binary digits equivalent.

|  |  |  |
| --- | --- | --- |
| **Step** | **Decimal Number** | **Conversion** |
| Step 1 | 2910 | 00102 10012 |
| Step 2 | 2910 | 00101001BCD |

Result

(11101)2 = (00101001)BCD

BCD to Binary Conversion

Steps

* **Step 1** -- Convert the BCD number to decimal.
* **Step 2** -- Convert decimal to binary.

Example − convert (00101001)BCD to Binary.

Step 1 - Convert to BCD

BCD Number − (00101001)BCD

Calculating Decimal Equivalent. Convert each four digit into a group and get decimal equivalent for each group.

|  |  |  |
| --- | --- | --- |
| **Step** | **BCD Number** | **Conversion** |
| Step 1 | (00101001)BCD | 00102 10012 |
| Step 2 | (00101001)BCD | 210 910 |
| Step 3 | (00101001)BCD | 2910 |

BCD Number − (00101001)BCD = Decimal Number − 2910

Step 2 - Convert to Binary

Used long division method for decimal to binary conversion.

Decimal Number − 2910

Calculating Binary Equivalent −

|  |  |  |  |
| --- | --- | --- | --- |
| **Step** | **Operation** | **Result** | **Remainder** |
| Step 1 | 29 / 2 | 14 | 1 |
| Step 2 | 14 / 2 | 7 | 0 |
| Step 3 | 7 / 2 | 3 | 1 |
| Step 4 | 3 / 2 | 1 | 1 |
| Step 5 | 1 / 2 | 0 | 1 |

As mentioned in Steps 2 and 4, the remainders have to be arranged in the reverse order so that the first remainder becomes the least significant digit (LSD) and the last remainder becomes the most significant digit (MSD).

Decimal Number − 2910 = Binary Number − 111012

Result

(00101001)BCD = (11101)2

BCD to Excess-3

Steps

* **Step 1** -- Convert BCD to decimal.
* **Step 2** -- Add (3)10 to this decimal number.
* **Step 3** -- Convert into binary to get excess-3 code.

Example − convert (0110)BCD to Excess-3.

Step 1 − Convert to decimal

(0110)BCD = 610

Step 2 − Add 3 to decimal

(6)10 + (3)10 = (9)10

Step 3 − Convert to Excess-3

(9)10 = (1001)2

Result

(0110)BCD = (1001)XS-3

Excess-3 to BCD Conversion

Steps

* **Step 1** -- Subtract (0011)2 from each 4 bit of excess-3 digit to obtain the corresponding BCD code.

Example − convert (10011010)XS-3 to BCD.

Given XS-3 number = 1 0 0 1 1 0 1 0

Subtract (0011)2 = 1 0 0 1 0 1 1 1

--------------------

BCD = 0 1 1 0 0 1 1 1

Result

(10011010)XS-3 = (01100111)BCD