Binary Coded Decimal (BCD)

Binary coded decimal (BCD) is a way to express each of the decimal digits with a binary code. There are only ten code groups in the BCD system, so it is very easy to convert between decimal and BCD. Because we like to read and write in decimal, the BCD code provides an excellent interface to binary systems. Examples of such interfaces are keypad inputs and digital readouts.

The 8421 BCD Code

The 8421 code is a type of **BCD** (binary coded decimal) code. Binary coded decimal means that each decimal digit, 0 through 9, is represented by a binary code of four bits. The designation 8421 indicates the binary weights of the four bits ($2^3, 2^2, 2^1, 2^0$). The ease of conversion between 8421 code numbers and the familiar decimal numbers is the main advantage

Decimal/BCD conversion.

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

Invalid Codes

You should realize that, with four bits, sixteen numbers (0000 through 1111) can be represented but that, in the 8421 code, only ten of these are used. The six code combinations that are not used—1010, 1011, 1100, 1101, 1110, and 1111—are invalid in the 8421 BCD code.

1. **Decimal to BCD convertion**:

The same as the conversation between Hexa → Binary

Ex:

$(67.\ 9)_{10}$

BCD $(0110\ 0111.\ 1001)_{BCD}$

Ex:

$(50\ .\ 30)_{10}$

BCD $(0101\ 0000\ \ .\ 0011\ 0000\ )_{BCD}$

Less than or equal to '9'

Examples:

Convert each of the following decimal numbers to BCD:

(a) 35 (b) 98 (c) 170 (d) 2469

Solution

(a) 3 → 0011, 5 → 0101: 00110101

(b) 9 → 1001, 8 → 1000: 10011000

(c) 1 → 0001, 7 → 0111, 0 → 0000: 000101110000

(d) 2 → 0010, 4 → 0100, 6 → 0110, 9 → 1001: 0010010001101001

Related Problem

Convert the decimal number 9673 to BCD.

2. BCD to Decimal:

The same as the conversation between Binary → Hexa

Ex:

$(0110\ 0101\ .\ 0011\ )_{BCD}$

BCD $(65\ .\ 3)_{10}$

Ex:

$(0001\ 0100\ .\ 0000\ 0010\ )_{BCD}$

BCD $(14\ .\ 02)_{10}$

Ex:

$(0101\ .\ 1101\ )$ ✖ Invalid is not BCD

5 ↙ ↘ 13>9

The Gray Code

The **Gray code** is unweighted and is not an arithmetic code; that is, 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 code word to the next in sequence.* This property is important in many applications, such as shaft position encoders, where error susceptibility increases with the number of bit changes between adjacent numbers in a sequence.

Table 2–6 is a listing of the 4-bit Gray code for decimal numbers 0 through 15. Binary numbers are shown in the table for reference. Like binary numbers, *the Gray code can have any number of bits.* Notice the single-bit change between successive Gray code words. For instance, in going from decimal 3 to decimal 4, the Gray code changes from 0010 to 0110, while the binary code changes from 0011 to 0100, a change of three bits. The only bit change in the Gray code is in the third bit from the right: the other bits remain the same.

TABLE 2–6

Four-bit Gray code.

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

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:

```
1 − + → 0 − + → 1 − + → 1 − + → 0     Binary
↓       ↓       ↓       ↓       ↓
1       1       1       0       1     Gray
```

The Gray code is 11101.

Gray-to-Binary Code Conversion

To convert from Gray code to binary, use a similar method; however, there are some differences. The following rules apply:

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:

```
1       1       0       1       1     Gray
↓  + ↗  ↓  + ↗  ↓  + ↗  ↓  + ↗  ↓
1       0       0       1       0     Binary
```

The binary number is 10010.

EXAMPLE 2–37

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

(b) Convert the Gray code 10101111 to binary.

Solution

(a) Binary to Gray code:

```
1 − + → 1 − + → 0 − + → 0 − + → 0 − + → 1 − + → 1 − + → 0
↓       ↓       ↓       ↓       ↓       ↓       ↓       ↓
1       0       1       0       0       1       0       1
```

(b) Gray code to binary:

```
1      0      1      0      1      1      1      1
↓ + ↗  ↓ + ↗  ↓ + ↗  ↓ + ↗  ↓ + ↗  ↓ + ↗  ↓ + ↗  ↓
1      1      0      0      1      0      1      0
```

Related Problem

(a) Convert binary 101101 to Gray code.

(b) Convert Gray code 100111 to binary.

H.W:

Binary	Octal	Hex	Gray	Decimal	BCD
101011.101					
	67.5				
		DA.B			
			11011.1011		
				43.625	
					1001.0101

Arithmetic Operation:

1. Addition:

a. Addition in decimal: Base 10

1 1

368_{10}

$+ \ 287_{10}$

655   15 >= 10 then 15 -10 = 5

Less than 10 then no change

1 1 1

1697_{10}

$+ \quad 888_{10}$

2585

15>= 10 then 15 – 10 =5

18 >= 10 then 18 -10 = 8

15>=10 then 15-10 = 5

b. <u>Addition in Octal</u>: base 8

1 1

365_8

$+ \ 177_8$

564

12 >=8 then 12 – 8 =4

14>= 8 then 14 – 8=6

5<8 then no change

Ex:

1 1 1

377_8

$+ \ 401_8$

1000

8>= 8then 8-8=0

8>=8 then 8-8 =0

8>=8 then 8-8 =0

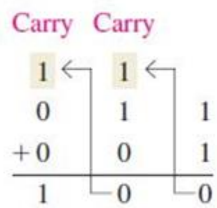
---

c. <u>Addition in binary</u>: base2

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

Notice that the first three rules result in a single bit and in the fourth rule the addition of two 1s yields a binary two (10). When binary numbers are added, the last condition creates a sum of 0 in a given column and a carry of 1 over to the next column to the left, as illustrated in the following addition of 11 + 1:



EXAMPLE 2–7

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)

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

Related Problem

Add 1111 and 1100.

Ex:

$$\begin{array}{r} 1\;1\;1\;1\;\;1\;\;\;\;\;\; \\ 1\;0\;1\;1\;1\;0 \\ +\;1\;1\;1\;0\;1\;1 \\ \hline 1\;1\;0\;1\;0\;0\;1 \end{array}$$

Ex:

$$\begin{array}{r} 11111\;\; \\ 11101 \\ +\;\;11111 \\ \hline 111100 \end{array}$$

d. Addition in Hex: base 16

Ex:

```
  1 1
   CF2
+  EDB
 1BCD
       28>= 16 then 28 – 16 =12   →   C
       27>=16 then 27 – 16=11     →   B
```

Ex:

```
 1 1 1 1
   FE. DA
 + BD. CA
  1BC. A4
       20>= 16 then 20 – 16 =4
       26>= 16 then 26 – 16 =10   →   A
       28>= 16 then 28 – 16 =12   →   C
       27>= 16 then 27 – 16 =11   →   B
```

e. Addition in BCD system:

Note:

If the result of the current digit > 9 or carry = 1 then add 6 the result.

```
  01010111
 +00100110              57
  01111101 > 9          26
 +    0110              83
  10000011
```

Ex:

```
      00110010
    + 10010101
    ----------
      11000111
      └──┘ >9            32
      0110               95
    ----------        --------
  000100100111
  └─┘└─┘└─┘
   1  2  7   ──────►    127
```

Ex:

```
  0111¹1001  ──────►  Carry, add 6

+ 0001  1001
-------------
  1001  0010             79

        0110             19
-------------         --------
  1001  1000  ──────►    98
```