Logical Design

Lecture 4: BCD and Gray Code

Produced By: Dr. Haleema Essa

2023-2024

Binary Coded Decimal (BCD)

A code used to represent each decimal digit of a number by a **4-Bit** Binary Value, the following table represents a conversion of decimal number to BCD.

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

However, binary coded decimal is not the same as hexadecimal. Whereas a 4-bit hexadecimal number is valid up to $(F)_{16}$ representing binary $(1111)_2$, (decimal 15), binary coded decimal numbers ***stop at 9 binary*** $(1001)_2$.

Example: $(357)_{10}$ = 0011 0101 0111 (BCD).

H.W

Convert each of the following decimal numbers to BCD

A- **45 = ?**

B- **2693 = ?**

Addition of BCD code

Step 1: Add the two BCD numbers, using the rules for binary addition.

Step 2: If a 4-bit sum is equal to or ***less than 9***, it is a ***valid*** BCD number.

Step 3: If a 4-bit sum is ***greater than 9***, or ***if a carry*** out of the 4-bit group is generated, it is an ***invalid*** result. ***Add 6 (0110)*** to the 4-bit sum in order to skip the six invalid BCD code words and ***return the code to 8421***. If a ***carry results*** when 6 is added, simply ***add the carry to the next 4-bit group***.

Example1: Find the sum of the BCD numbers 01000011 + 00110101

```
    0100  0011        43
+   0011  0101      + 35
  ------------      ----
    0111  1000        78
```

Example 2: Find the sum of the BCD numbers 01110101 + 00110101

```
          0111    0101                            75
   +      0011    0101                          + 35
 ----------------------                         -----
          1010    1010                           110
         +0110   +0110
 ----------------------
 0001     0001    0000
  1        1       0
```

Both left and right BCD numbers are invalid. So we would add 6 to both the BCD numbers.

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 number to the next***. Table below is a listing of the four-bit Decimal, Binary, Gray Code for decimal numbers 0 through 15.

Decimal (base 10)	Binary (base 2)	Binary-Reflected (no base)
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	1000	1100
9	1001	1101
10	1010	1111

$g3 = b3$
$g2 = b3 \oplus b2$
$g1 = b2 \oplus b1$
$g0 = b1 \oplus b0$

11 or 00 ⟶ 0 (similar put 0)
01 or 10 ⟶ 1 (different put 1)

Advantages:

1. Can be used to minimize a logic circuit.

2. It minimize error while converting analog to digital signals.

3. It is widely used in digital communications, such as digital terrestrial television and cable TV systems, to correct errors.

Disadvantages:

1. It is not suitable for arithmetic operations.

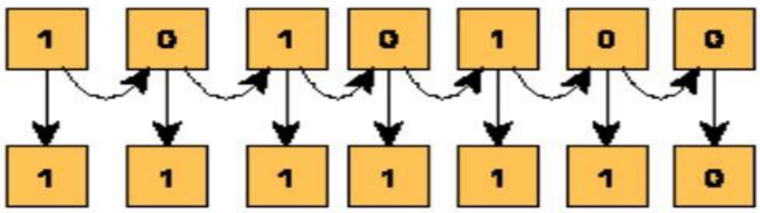
2. It is limited to few practical applications.

Binary-to-Gray Conversion

1. The Most Significant Bit (MSB) in the gray code is the same as the corresponding MSB in binary number.

2. Going from left to right, apply **XOR** on each adjacent pair of binary code bits to get the next gray code bit.

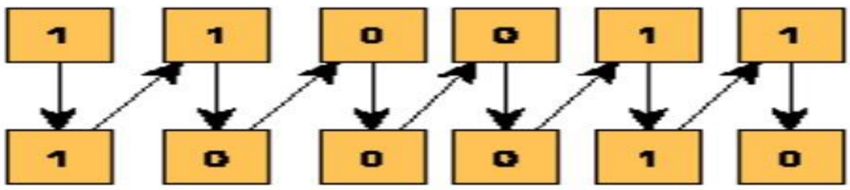
11 or 00 → 0 (similar put 0)
01 or 10 → 1 (different put 1)



Gray-to-Binary Conversion

1− The MSB is the binary code is same as corresponding MSB in the Gray code.

2− **Add** each binary digit generated to the Gray digit in the next adjacent position and discard carry.



<u>H.W 1 :</u> Encode to Gray Code

A- $(1101)_2$ = **(?)**$_{\textbf{gray}}$

B- $(0010)_2$ = **(?)**$_{\textbf{gray}}$

<u>H.W 2 :</u> Covert the following Gray Code to Binary

A- $(1001101)_{\textbf{gray}}$ = **(?)**$_2$

B- $(10110011)_{\textbf{gray}}$ = **(?)**$_2$