Coding system

Objectives:-

At the end of the lecture students may learn:

1- Binary coded decimal (BCD-8421)

2- The 2421-code

3- The Gray code

4- Excess-3 code

1-Binary coded decimal (BCD-8421)

Code have been used for security reasons, so that others will not be able to read the message. There are many types of codes, each of them have 4-bits and different weight.

BCD(8421) code is most commonly used for the decimal digits is the straight binary assignment as listed below. This code gives the 4-bits code for one decimal digit. A number with K decimal digits will require 4K bits in BCD. It is easy of converting it to decimal numbers and vice versa. It is unable to code binary numbers greater than 9.

Note:- the binary combinations 1010 through 1111 are not valid and have no meaning in BCD code.

Decimal	BCD-8421
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

Note: (1010, 1100,1101,1110,1111) are invalid in BCD-8421 code.

Conversion from decimal to BCD or BCD to decimal numbers.

Example-1: $(937.25)_{10} \Rightarrow (\quad\quad)_{BCD}$

Solution: $(1001\ 0011\ 0111.\ 0010\ 0101)_{BCD}$

Example-2: $(1000101.011001)_{BCD} \Rightarrow (\quad\quad)_{10}$

Solution: 0100 0101. 0110 0100

4 5 . 6 4

the result $(45.64)_{10}$

BCD-8421 Addition

- Add two BCD numbers, using rules for binary addition
- If a 4-bits sum ≤ 9 , it is a valid BCD number.
- If a 4-bits sum > 9, or if a carry out of the 4-bits group is generated it is invalid, the result add 6(0110) to the 4-bits sum in order to skip the six invalid state and return the code to BCD-8421
- If a carry result when 6 is added, add the carry to next 4-bits group

Example-3: Add the following numbers using BCD addition.

1- $(45)_{10} + (33)_{10} = (\qquad\qquad)_{BCD}$

$$
\begin{array}{rcc}
 & 0100 & 0101 \\
+ & 0011 & 0011 \\
\hline
 & 0111 & 1000 \\
 & 7 & 8
\end{array}
$$

The result is $(01111000)_{BCD}$

2- $(47)_{10} + (35)_{10} = (\qquad\qquad)_{BCD}$

$$
\begin{array}{rrrl}
 & 0100 & 0111 & \\
+ & 0011 & 0101 & \\
\hline
 & 0111 & 1100 & (12 > 9 \text{ , add } 6 \text{)} \\
 & 1 & 0110 & + \\
\hline
 & 1000 & 0010 & \\
 & 8 & 2 & \text{result}(82)_{10} = (10000010)_{BCD}
\end{array}
$$

3- $(59)_{10} + (38)_{10} = (\qquad\qquad)_{BCD}$

$$
\begin{array}{rrrl}
 & 0101 & 1001 & \\
+ & 0011 & 1000 & \\
 & 1 & & \\
\hline
 & 1001 & 10001 & (\ 17 > 9) \text{ and carry } 1 \\
 & & 0110 \quad + & (\text{add six}) \\
\hline
 & 1001 & 0111 & \\
 & 9 & 7 & \text{result } (97)_{10} = (10010111)_{BCD}
\end{array}
$$

4- $(67)_{10} + (53)_{10} = (\qquad)_{BCD}$

$$
\begin{array}{rrrrl}
 & & 0110 & 0111 & \\
 & + & 0101 & 0011 & \\
\hline
(11>9) & & 1011 & 1010 & (10>9) \\
 & & 0110 & 0110\ + & \\
 & & 1\ + & & \\
\hline
0001 & & 0010 & 0000 & \text{result } (120)_{10}
\end{array}
$$

2- The 2421- code

Decimal	2421
0	0000
1	0001
2	0010
3	0011
4	0100
5	1011
6	1100
7	1101
8	1110
9	1111

Self complementing

Conversion from decimal to 2421 or from 2421 to decimal

Example-4 : $(34.5)_{10} \Rightarrow (\quad)_{2421}$

Solution: $(34.5)_{10} = (0011\ 0100.\ 1011)_{2421}$

Example -5 : $(11011.01)_{2421} \Rightarrow (\quad)_{10}$

Solution: 0001 1011. 0100

$(1 \quad 5 \quad . \quad 4)_{10}$

Note: there are six invalid cases:

(0101,0110,0111,1000,1001,1010)

5 6 7 8 9 10

3- The Gray code

Decimal	Binary	Gray Code
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
11	1011	1110
12	1100	1010
13	1101	1011
14	1110	1001
15	1111	1000

Binary-to- Gray code conversion

- It is not a weighted code.
- It is not an arithmetic code
- It exhibits only a single bit change from one number to the next.
- The most significant bit (MSB) in the Gray code is the same as the corresponding (MSB) in the binary numbers.
- Going from left to right, add each adjacent pair of binary code bits to get the next Gray code bit.

Example-6: 1 → 0 → 1 → 1 → 0 Binary

↓ ↓ ↓ ↓ ↓

1 1 1 0 1 Gary

Gray – to – Binary conversion

- (MSB) in the binary code is the same as the corresponding bit in the Gray code.
- Add each binary code bit generated to the gray code bit in the next adjacent position.

Example-7: 1 1 0 1 1 Gray

↓ ↓

1 0 0 1 0 Binary

4- Excess-3 code

The excess-3 code is an important 4-bits code some times used with binary-coded decimal(BCD) numbers. To convert any decimal number into its excess-3 form, add 3 to each decimal digit, and then convert the sum to a BCD number.

- It is also a self- complementing code
- There are 6-invalid cases(0000,0001,0010,1101,1110,1111)

0 1 2 13 14 15

Decimal	Binary	Excess-3 Code
0	0000	0011
1	0001	0100
2	0010	0101
3	0011	0110
4	0100	0111
5	0101	1000
6	0110	1001
7	0111	1010
8	1000	1011
9	1001	1100

Decimal – to – Excess-3 code conversion

Example-7: $(12)_{10} \Rightarrow (\qquad)_{XS-3}$

Solution:

$$\begin{array}{rr} 1 & 2 \\ +3 & 3 \\ \hline 4 & 5 \end{array}$$

Binary 0100 0101

the result $(0100\ 0101)_{XS-3}$

Example-8: $(29)_{10} \Rightarrow (\quad)_{XS-3}$

Solution:

$$
\begin{array}{rcc}
 & 2 & 9 \\
+ & 3 & 3 \\
\hline
 & 5 & 12
\end{array}
$$

Binary (0101 1100)

The result $(0101\ 1100)_{XS-3}$

Excess-3 code –to – decimal conversion

Example-9: $(1100\ 0110)_{XS\text{-}3} \Rightarrow (\quad\quad)_{10}$

Solution: 1100 0110

1- conversion to decimal 12 6

2-subtract 3 from each number - 3 - 3

$(\ 9 \quad 3\ )_{10}$