Digital Electronics- 2CS303

UNIT-1
Binary Codes

Dr. Sudeep Tanwar

Binary Codes

1. Weighted Codes (8421, 5421)
2. Non Weighted codes (Excess-3 , BCD, Gray/Reflected codes)
3. Error Detection Codes (PARITY Bits)
4. Alphanumeric Codes (ASCII)

Codes

- The **problem** found with computer is how to represent numerals, alphabets, and special characters
- Since data is in the binary form, it must be **converted to a more readable** form known as **Coded form**

1: Weighted Codes:

Weighted binary *codes* are those binary *codes* which obey the positional weight principle. Each position of the number represents a specific weight.

Decimal No.s in Weighted Codes:-

Decimal	8421	7421	5421	*5211	*4221	*3321	*2421	*$84\bar{2}\bar{1}$	$74\bar{2}\bar{1}$
0	0000	0000	0000	0000	0000	0000	0000	0000	0000
1	0001	0001	0001	0001	0001	0001	0001	0111	0111
2	0010	0010	0010	0100	0010	0010	0010	0110	0110
3	0011	0011	0011	0101	0011	0100	0011	0101	0101
4	0100	0100	0100	0111	1000	0101	0100	0100	0100
5	0101	0101	1000	1000	1001	0110	0101	1011	1010
6	0110	0110	1001	1010	1010	1100	0110	1010	1001
7	0111	1000	1010	1100	1011	1101	0111	1001	1000
8	1000	1001	1011	1110	1110	1110	0110	1000	1111
9	1001	1010	1100	1111	1111	1111	1111	1111	1110

2: Non-Weighted Codes:

They do **not have a fixed weight** assigned to each symbol position in the code word. For example, ASCII, BCD, Exess-3, and Gray code.

2.1: Excess-3 is a non-weighted coding method. With excess-3, we add 3 to a decimal number before converting it to binary.

Example:
(0001)2 = (0100)Excess-3
(0010)2 = (0101)Excess-3

2.2: BCD (Binary Coded Decimals) is a non-weighted coding method. Individual decimal digits are converted into equivalent binary bits.

Example:
(321)2 = (0011 0010 0000)BCD
(000)2 = (0000 0000 0000)BCD
(80)2 = (1000 0000)BCD
(00)2 = (0000 0000)BCD
(10)10 = (0001 0000) BCD
(11)10 = (0001 0001) BCD

BCD

Decimal.	8421 BCD.	
0	0000	
1	0001	
2	0010	
3	0011	
4	0100	
5	0101	Unpacked BCD.
6	0110	
7	0111	
8	1000	
9	1001	
10	0001 0000	
11	0001 0001	
12	0001 0010	Packed BCD.
13	0001 0011	
14	0001 0100	
15	0001 0101	

2: Non-Weighted Codes:

2.3: Gray Codes: It is also called as Reflected Binary codes. It is generated via getting mirror image of given data. Only 1 bit will change each time when the decimal number is incremented. Where as the binary system requires all four bits to change when going from 7 to 8

Example : 4 bit Gray codes.

Decimal	0	1	2	3	4	5	6	7	8	9	10	11	12
Gray.	0000	0001	0011	0010	0110	0111	0101	0100	1100	1101	1111	1110	1010
Decimal.	13	14	15										
Gray.	1011	1001	1000										

0	0	0	0
0	0	0	1
0	0	1	1
0	0	1	0
0	1	1	0
0	1	1	1
0	1	0	1
0	1	0	0
1	1	0	0
1	1	0	1
1	1	1	1
1	1	1	0
1	0	1	0
1	0	1	1
1	0	0	1
1	0	0	0

2: Code Conversion:

Converting Binary Codes to Gray codes

Method:

1. Copy MSB of Binary code to MSB of Gray code. **(As it is)**
2. Add MSB of Binary with Next MSB of Binary to get next Gray code.
3. Discard the carry
4. Repeat the same process till we get the LSB

Example: Convert (0001)2 into Gray code.

0	0	0	1
	0+0= 0	0+0= 0	0+1= 1
0	0	0	1

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

2: Code Conversion:

Converting Binary Codes to Gray codes

Method:

1. Copy MSB of Binary code to MSB of Gray code.
2. Add MSB of Binary with Next MSB of Binary to get next Gray code.
3. Discard the carry
4. Repeat the same process till we get the LSB

Example: Convert (1011)2 into Gray code.

1 0 1 1

1+0= 1 0+1= 1 1+1= 0

1 1 1 0

Decimal.	Binary.	Gray.
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

2: Code Conversion:

Converting Gray Codes to Binary codes

Method:

1. Copy MSB of Gray code to MSB of Binary code. **(As it is)**
2. If next bit of Gray code is "1", then invert the present binary bit as next bit.
3. If next bit of Gray code is "0" then copy the present binary bit as next bit.

Example: Convert (1011) gray into Binary code.

1	0	1	1
↓	↓	↓	↓
1	1	0	1

By ex-oring method.

0 1 0 1

$\oplus$ $\oplus$ $\oplus$

0 1 1 0 ⇒ $(0110)_2$ Bin Code.

Decimal	Gray	Binary
0	0000	0000
1	0001	0001
2	0011	0010
3	0010	0011
4	0110	0100
5	0111	0101
6	0101	0110
7	0100	0111
8	1100	1000
9	1101	1001
10	1111	1010
11	1110	1011
12	1010	1100
13	1011	1101
14	1001	1110
15	1000	1111

Binary to BCD and BCD to Binray:

1. Binary to BCD :-

Decimal	Bin	BCD
0	0000	0000
1	0001	0001
2	0010	0010
3	0011	0011
4	0100	0100
5	0101	0101
6	0110	0110
7	0111	0111
8	1000	1000
9	1001	1001
10	1010	00010000
11	1011	00010001
12	1100	00010010
13	1101	00010011
14	1110	00010100
15	1111	00010101

2. BCD to Binary :-

Decimal	BCD	Binary
0	0000	0000
1	0001	0001
2	0010	0010
3	0011	0011
4	0100	0100
5	0101	0101
6	0110	0110
7	0111	0111
8	1000	1000
9	1001	1001
10	0001 0000	1010
11	0001 0001	1011
12	0001 0010	1100
13	0001 0011	1101
14	0001 0100	1110
15	0001 0101	1111

BCD to Excess-3 and Excess-3 to BCD:

3r. BCD to Excess-3:-

Decimal	BCD	Excess-3
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

4r. Excess-3 to BCD:-

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

Binary to Excess-3 and Excess-3 to Binary:

44

5> Binary to Excess-3:-

Decimal	Bin.	Excess-3.
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
10	1010	1101
11	1011	1110
12	1100	1111

6> Excess-3 to Binary:-

Decimal	Excess-3	Bin.
0	0011	0000
1	0100	0001
2	0101	0010
3	0110	0011
4	0111	0100
5	1000	0101
6	1001	0110
7	1010	0111
8	1011	1000
9	1100	1001
10	1101	1010
11	1110	1011
12.	1111	1100

BCD Arithmetic:

Example 1: (83) +(11) =(94)

83	1 0 0 0	0 0 1 1
11	0 0 0 1	0 0 0 1
	1 0 0 1	0 1 0 0 = (94)

Error Detecting Codes

ERROR DETECTING CODES:— Digital System must be accurate to the digit, errors can become a serious problem. Now error detecting codes plays an important role they identify the error & then send the message in the form of acknowledgement.

i. Parity:— This is the simple technique for error detecting. According to this technique, adding an extra bit known as parity bit, to each word being transmitted. For odd parity this bit is set to a 1 or 0 at the tx such that the sum of the 1 bits in the entire word is odd.

e.g.

Parity Bit	Data Bits	Total no. of 1's
1	1 0 1 0 1 1 0	5
0	1 0 1 0 1 1 1	5
1	0 0 1 0 0 1 0	3
1	0 0 0 0 0 0 0	1

In above e.g. for each case total no. of ~~words~~ bits 1's in the word has odd, including the parity parity bit.

At the receiving end, if a word is received that has an even no. of 1's, the receiver will request a retransmission. Since most errors occurs as a single-bit inversion, this system works quite well. But if two errors occurs within the word, they would remain undetected.

Error Detecting Codes

Now it is totally depends upon the user that wheather he or she require odd parity or even parity. In case of even parity the sum of no. of 1's will be equal to even.

e.g.

Parity bit	Data bit	Total no. of 1's
1	1010111	6
1	0000111	4
0	1010000	2.

Problem :- Attach even parity bits to the message 'HELP' in ASCII code.

iii. Check Sum :-

Message HELLO using ASCII code with odd parity.

	Odd parity bits	Data bits
H	1	1001000
E	0	1000101
L	0	1001100
L	0	1001100
O	0	1001111