DiS Notes (divB):- ODD 2020-2021

(Author: Prof Ravindra Divekar)

Number Systems

Decimal / Binary / Octal / Hexadecimal

Decimal = base 10
Binary = base 2
Octal = base 8
Hexadecimal = base 16

e.g.1 Convert number $(30)_d$ to binary, octal, hexadecimal

2 | 30 0

2 | 15 1

2 | 07 1

2 | 03 1

2 | 01 1

0 ← STOP

↑**Collect remainder bits in reverse order i.e. bottom to top**

$(30)_{10} = (111110)_2$

How can we verify?

$(11110)_2 =$ 1x2^^{4} + 1x2^^{3} + 1x2^^{2} + 1x2^1 + 0x2^0 = 16 + 8 + 4 + 2 + 0 = 30

Now,

4| 30 2

4| 07 3

4| 01 1

0 ← STOP

Or [Shortcut]: $(11110)_2 =$ (01 11 10)$_2 =$ $(132)_4$

How can we verify?

$1 \times 4^2 + 3 \times 4^1 + 2 \times 4^0 = 16 + 12 + 2 = 30$

Similarly,

$8 \overline{| \ 30} \quad 6$

$8 \overline{| \ 03} \quad 3$

0 ← STOP

Or [Shortcut]: $\mathbf{(11110)_{2} = (\ 011\ 110)_{2} = (36)_{8}}$

How can we verify?

$3 \times 8^1 + 6 \times 8^0 = 24 + 6 = 30$

$16 \overline{| \ 30} \quad 14$

$16 \overline{| \ 01} \quad 01$

0 ← STOP

$\mathbf{(11110)_{2} = (\ 0001\ 1110)_{2} = (1E)_{16}}$

How can we verify?

$1 \times 16^1 + 14 \times 16^0 = 16 + 14 = 30$

e.g 3: Convert $(30)_{10}$ to $(30)_3$

[You can use the same principle for any radix – it does not have to be a power of 2!]

3 | 30   0

3 | 10   1

3 | 03   0

3 | 01   1

0 ← STOP

(30)10 = (1010)3 = $1x3^3 + 1x3^1$ = 27 + 3 = 30

e.g.4: Convert $(612)_7$ to base 10

$6x7^2+1x7^1+2x7^0 = 294 + 7 + 2 = (303)_{10}$

Practice Problems:-

P1) Convert $(100)_{10}$ to binary, octal and hexadecimal

P2) $(34)_5 = (?)_9$

P3) Convert $(A09)_{16}$ to decimal.

Ones Complement:

Just reverse the bits:
+5 = 0101
-5 = 1010
But 1010 is also +10
So to avoid confusion we add a sign bit

+5 = 0 101
-5 = 1 010

Addition & Subtraction using One's complement:

<u>Add 04 + 07 (no complement involved)</u>

```
+07 = 111
+04 = 100
      -------
     1011
```

<u>Add -04 + 07</u>

```
-04 = ~(0100) = 1011
+07 =  0111    = 0111
                 ------
                10010
```

The carry-over or overflow bit is removed and added to the remaining number:-

```
   0010
 +    1
 ---------
   0011 ➔ answer = +3
```

Add +04 -07

+04 = 0100   = 00100
-07 = ~(00111) = 11000 ← important
(+07 is 0111. -7 is 1000 which is also +08. To avoid confusion add extra bit).

```
  00100
+11000
---------
  11100
```
← Note when bigger number is negative there is no carry, so we reverse the number using one’s complement
~(11100) = -(00011) = -3 ← answer

Add -04 -07

-04 = ~(00100) = 11011
-07 = ~(00111) = 11000

```
   11011
+ 11000
----------
  110011
```
Remove carry bit and add it to the number, and then reverse it.

```
10011
    + 1
---------
10100
```
~(10100) = -(01011) -11 ← answer

Two's complement:

Ex1: 4 + 7 = 00100 + 00111 = 01011

Ex2: 4 – 7:

```
  00100
+11001
----------
  11101
```

2's comp of 11101 = -(00010+1) = -3

Ex3: -4 – 7:

```
  11100
+11001
-----------
110101
```

110101 Answer is in 2's complement form, so reverse and attach minus sign:

- (001011)

Ex4: -09 + -06:

```
+09 = 01001
        10110
 -09=  10111

+06 = 00110
        11001
-06 =   11010
```

```
   10111
+ 11010
   1 1  0  0  0 1
   001110
```

1

-(001111) = -15

Weighted and non-weighted codes

1. A sequence of binary bits which represents a decimal digit is called a “code word”.
2. Thus x4x3x2x1 is a code word of N.
3. Example of **these codes is: BCD, 8421, 6421, 4221, 5211, 3321 etc.**
4. **Weighted** codes are used in:
   a) Data manipulation during arithmetic operation.
   b) For input/output operations in digital circuits.
   c) To represent the decimal digits in calculators, volt meters etc.

Weighted Codes:

Here each position has a fixed weight, not necessarily a power of 2 or in increasing order:

06 (decimal) = 0110 (binary) = 0x2^3 + 1x2^2 + 1x2^1 + 0x2^0

8 4 2 1 code: (normal binary)
0 1 1 0 = 06

6 4 1 1 code:
1 0 0 0 = 06

2 3 2 1 code:
0 1 1 1 = 06
1 1 0 1 (This combination also possible, but always go for lowest bit combination).

Non-Weighted Codes:

1. Non-weighted or un-weighted codes are those codes in which the digit value does not depend upon their position i.e., each digit position within the number is not assigned fixed value.
2. Examples of non-weighted codes are: Un-weighted BCD code, Excess-3 code and Gray code.
3. Non weighted codes are used in:
   a) To perform certain arithmetic operations.
   b) Shift position encodes.
   c) Used for error detecting purpose.

GRAY CODE

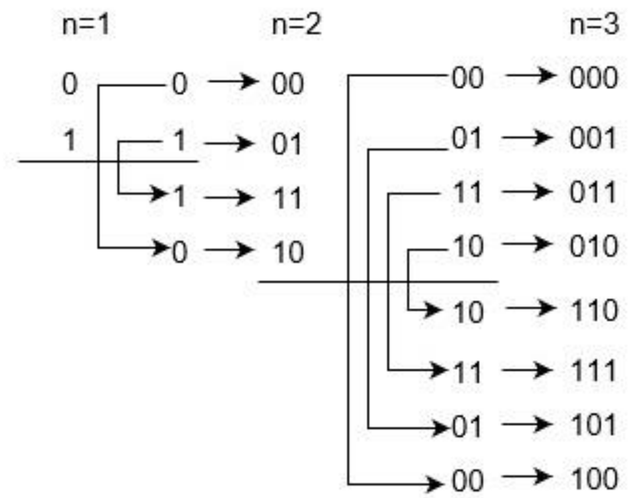
Gray code also known as reflected binary code, because the first (n/2) values compare with those of the last (n/2) values, but in reverse order.

Constructing an *n*-bit Gray code

n-bit Gray code can be generated recursively using reflect and prefix method which is explained as following below.

- Generate code for n=1: 0 and 1 code.
- Take previous code in sequence: 0 and 1.
- Add reversed codes in the following list: 0, 1, 1 and 0.
- Now add prefix 0 for original previous code and prefix 1 for new generated code: 00, 01, 11, and 10.

Therefore, Gray code 0 and 1 are for Binary number 0 and 1 respectively. Gray codes: 00. 01, 11, and 10 are for Binary numbers: 00, 01, 10, and 11 respectively. Similarly you can construct Gray code for 3 bit binary numbers:



Therefore, Gray codes are as following below,

For n = 1 bit		**For n = 2 bit**		**For n = 3 bit**	
Binary	Gray	Binary	Gray	Binary	Gray
0	0	00	00	000	000
1	1	01	01	001	001
		10	11	010	011
		11	10	011	010
				100	110

For n = 1 bit	For n = 2 bit	For n = 3 bit	
		101	111
		110	101
		111	100

Binary to Gray code

Binary number: 10101 – Gray code 11111 – Binary 10101

1 0 1 0 1
↓ xor --xor-- --xor-- --xor--

1 1 1 1 1

↓ xor ↗ ↓ xor ↗ ↓ xor ↗ ↓ xor ↗ ↓

1 0 1 0 1

Example 2:
Binary number = 1101110
1
1 xor 1 = 0
1 xor 0 = 1
0 xor 1 = 1
1 xor 1 = 0
1 xor 1 = 0
1 xor 0 = 1
Gray code = 1011001
1--------→ 1
1 xor 0 = 1
1 xor 1 = 0
0 xor 1 = 1
1 xor 0 = 1
1 xor 0 = 1
1 xor 1 = 0
Binary = 1101110

Example2: decimal 69 → binary? Gray?

Example3: decimal 278 → binary? Gray?

278 = 256 + 16 + 4 + 2 = 100010110

100010110
Gray: 110011101
110011101
Binary:
100010110

Example : 24 =

Binary: 11000
1
11 → 0
10 → 1
00 → 0
00 → 0
Gray: 10100
1
10 → 1
11 → 0
00 → 0
00 → 0
Binary: 11000

0	0000	0000	
1	0001	0001	
2	0010	0011	
3	0011	0010	
4	0100	0110	
5	0101	0111	= x'yzw x'yzw + x'yz'w = x'yw
6	0110	0101	= x'yz'w
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

BCD numbers

BCD = binary coded decimal

e.g. number (11000)base2

= 2^4+2^3 = 16+8 = 24(decimal)

To write a decimal number as a BCD number, write each digit separately in binary form:
24 decimal = 0010 0100 = (00100100) BCD

Note:
(1) Original binary form is 11000 and
BCD form of the same number is 00100100.
(2) The first is proper or real binary;
The second is a “binary-like” number).

Example2:
Binary number is 11001000 = 2^7 + 2^6 + 2^3 = 128+64+8 = 200
Decimal: 200 BCD: 0010 0000 0000

Example3:
Binary number is 11111111 Decimal = 255 BCD: 001001010101

Q: How to convert (00100100) BCD from example1 back to binary?
0010 0100
24
11000

BCD addition

Case1: 5BCD+10BCD

```
  0101
+ 1010
---------
  1111
+ 0110
---------
1 0001
```

Case2:Now let 0001 0011 is added to 0010 0110.

```
  0001 0001
+ 0010 0110
-----------
  0011 0111  ⟶ Valid BCD number
```

$$(0001\ 0001)_{BCD} \rightarrow (11)_{10},\ (0010\ 0110)_{BCD} \rightarrow (26)_{10}\ and\ (0011\ 0111)_{BCD} \rightarrow (37)_{10} (11)_{10} + (26)_{10} = (37)_{10}$$

So no need to add 6 as because both $(0011)_2 = (3)_{10}\ and\ (0111)_2 = (7)_{10}$ are less than $(9)_{10}$. This is the process of BCD Addition.

Case3: add 99BCD+ 72BCD

```
  1001 1001
+ 0111 0010
------------------
 10000  1011
       +0110
-------------------
1 1000 0001
```

Case4: add 90BCD + 26BCD

```
   1001 0000
+  0010 0110
-------------------
   1011 0110
   0110
-------------------
1 0001 0110   116BCD
```

Excess-3 code (XS3):

- Non weighted code
- Self-complementary BCD code

An Excess-3 equivalent of a given binary number is obtained using the following steps:

- Find the decimal equivalent of the given binary number.
- Add +3 to each digit of decimal number.
- Convert the newly obtained decimal number back to binary number to get required excess-3 equivalent

Example-1 −Convert decimal number 23 to Excess-3 code.

So, according to excess-3 code we need to add 3 to both digit in the decimal number then convert into 4-bit binary number for result of each digit. Therefore,

= 23+33=56 =0101 0110 which is required excess-3 code for given decimal number 23.

Example-2 −Convert decimal number 15.46 into Excess-3 code.

According to excess-3 code we need to add 3 to both digit in the decimal number then convert into 4-bit binary number for result of each digit. Therefore,

= 15.46+33.33=48.79 =0100 1000.0111 1001 which is required excess-3 code for given decimal number 15.46.

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

$(0110)_{BCD} = 6_{10}$

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

$(10011010)_{XS-3} = (01100111)_{BCD}$

BCD Addition and Subtraction

Example1: Add 6 + 7

6 = 0110
7 = 0111

+ = 1101 ← valid result, but not a valid BCD format (only 0-9 allowed)
Result was greater than 9 so add 6

  1101
+ 0110

0001 0011

Example2: Add 59 + 38

59 =   0101 1001
38 =   0011 1000

+   = 1001 0001 ← 91 not a valid result; add 6 where
(a) sum was greater than 6
-----OR-----
(b) sum was less than 6 but carry was generated

1001 0001
+   0110

1001 0111 ← 97 correct answer

Example3: Add 95 + 83

95 = 1001 0101
83 = 1000 0011

+ = 10001 1000 ←   118 not a valid result; add 6 where
(a) sum was greater than 6
-----OR-----
(b) sum was less than 6 but carry was generated

1 0001 1000
+   0110
----
1 0111 1000 ← 178 right answer

Practice Problems:-

1)   6+3   2)   6 + 8   3) 22 + 59

4) 22 + 95   5) 99 + 99

22 = 0010 0010

95 = 1001 0101

+ ---------------------

10 1 1   0111

0110

----

1 000 1   0111 = 117

Nine's complement

https://www.youtube.com/watch?v=iMQyPYQ4YOw

9's complement of 6 = 9-6 = 3

9's complement of 28:

  99
- 28

  71
(subtract each digit from 9)

10's complement of 28? = (9's complement +1)

= 71+1 = 72 (10's complement of 28)

How can we verify?

  72
+ 28
----
 100 = 10^2 (because 72 and 28 are 2 digit numbers).

BCD Subtraction using nine's complement
https://www.youtube.com/watch?v=4qTPRwjM_Zc

Some more examples:

Example1:

8-5
  1000 + 0100 = 1100 → 1100 + 0110 = 1 0010 = 1+0010 = 0011 = 3 ← answer

Example2:

5 – 11

11 = 1011

-11 = 99-11 = 88

05 = 0000 0101

88 = 1000 1000

    1000 1101

              0110

    1001 0011 = 93   99-93 = -6

Example3:

98.3 – 81.2

98.3 = 1001 1000 . 0011

(9s complement of 81.2 = 99.9 – 81.2 = 18.7 = 0001 1000 . 0111)

```
   1001   1000 .    0011
+  0001   1000.     0111
--------------------------
          1         1
   1010     0000  1010
   0110     0110  0110
--------------------------
 1 0001     0111  0000
```

0001 0111 0000

1

0001 0111.0001 → 17.1

```
   1001   1000 .     0011
+  0001   1000.      0111
-----------------------------
   1011    0000   1010
   0110    0110   0110
-----------------------------
 1 0001    0111   0000
```

```
0001 0111 0000
             1
-----------------------
0001 0111.0001 ➔  17.1
```