Date 20/9/23
Page 2.

Unit-1.

Number System

→ It is basically a technique by which we can represent any number, ~~work~~ word which can be understandable by the machine that system is called number system.
→ Basically four number systems are defined (standard) these are -
1. Binary (Radix 2) 2. Decimal (8)
3. Octel (10) 4. Hexadecimal (16).

Decimal	Binary	Octel	Hexadecimal.
0	0	0	0
1	1	1	1
2	10	2	2
3	11	3	3
4	100	4	4
5	101	5	5
6	110	6	6
7	111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	~~20~~ E
15	1111	17	F
16	10000	20	10

BCD (Binary Coded Decimal)

Interconversion between Various Number Systems-

1. Binary to Octel-

$\underline{1\ 0\ 1}\ \underline{0\ 0\ 1}\ \underline{1\ 0\ 0}$ (Binary form).
4 2 1 4 2 1 4 2 1

5 1 4

$(514)_8$

21/9/22

2. Binary to Hexadecimal.

$(\underline{0011}\ \underline{0011}\ \underline{0101})_2 = (335)_{16}$.
8421 8421 8421

3. $(\underline{0101}\ \underline{1111}\ \underline{0000}\ \underline{1101}\ \underline{1110})_2 = (5F0DE)_{16}$.
8421 8421 8421
5 F 0 D E

3. Binary to Decimal.

$(\cancel{1}1\ 0\ 1\ 1\ \cancel{0}\ 1\ 1\ \cancel{0}\ 0\ 1) =$
128 64 32 16 8 4 2 1

1. $(11001) = 25$
16 8 4 2 1

2. $(101011)_2 = (43)_{10}$.
32 16 8 4 2 1

4. Decimal to Binary.

1. $(105)_{10} = (1101001)_2$.

2	105	
2	52	1
2	26	0
2	13	0
2	6	1
2	3	0
	1	1

2. $(105.75)_{10}$.

$0.75 \times 2 = 1.5 = 1$
$0.5 \times 2 = 1.0 = 1$

$(1101001.11)_2$.

5. Decimal to Octel.

1. $(105.75)_{10} = (151.06)_8$.

8	105.	
8	13	1
	1	5

$(151)_8$.

$0.75 \times 8 = 6.00$

2. $(375)_{10} = (567)_8$.

8	375	
8	46	7
	5	6

6. Decimal to Hexadecimal.

$(375)_{10} = (177)_{16}$.

16	375	
16	23	7
	1	7.

7. Octel to Binary : 3 bit (4 2 1)

1. $(375)_8 = (011\,111\,101)_2$.

8. Octel to Decimal.

$(375)_8 = 3 \times 8^2 + 7 \times 8^1 + 5 \times 8^0$

$= 3 \times 8^2 + 7 \times 8 + 5 \times 1 = (253)_{10}$.

9. Octel to Hexadecimal. (No direct conversion).

$(375)_8 = \underline{0000}\ \underline{11\ 11}\ \underline{1\ 10\ 1} = (0FD)_{16}$.

8421 8421 8421

0 F D

8421

10. Hexadecimal to binary conversion
every single digit of a number in hexadecimal is expressed in 4-bit binary equivalent.

$(ABC014)_{16}$ - 10 11 12 0 1 4

0 1 1 0 0

$(1010101111100\ 00000001\ 0100)_2$.

11. Hexadecimal to decimal.

$(A\ B\ C\ 0\ 1\ 4)_{16}$.

$= 16^5 \times 10 + 16^4 \times 11 + 16^3 \times 12 + 16^2 \times 0 + 16^1 \times 1 + 16^0 \times 4$

$= (11255828)_2$.

12. Hexadecimal to Octal conversion.
Hexadecimal → Binary
↓
will Take 3 bit combination

1. $(ABC014)_{16}$ = 1010 1011 1100 0000 0001 0100
5 2 7 4 0 0 2 4

$(52740024)_8$.

Unsigned = 6 16 8 4 2 1

Representation of a Number in Binary.

+8 = 01000 -8 = 11000

↓ represents sign

If first bit is ~~One~~ 0 then no. is positive else negative

A binary number can be represented ~~by~~ via two ways in a binary number system -

1. Signed magnitude ~~Two~~ numbers.
2. Unsigned magnitude number.

In signed ~~n~~ bit no. system we require additional bit to represent the sign ~~of~~ of the number whether the no. is positive or negative.

Complement

In digital computers to simplify substraction operations and for logical manipulation compliments are used. There are two types of compliments used in each radix system. (2, 8, 16, 10).

1. Radix Complement (R's complement).
2. The Diminished Radix Complement /(R-1)'s Complement.

1. Representation of Signed Numbers

1. Signed Magnitude Form.
2. Unsigned and Complement Form

Complement form further divided into.

(i) 1's Complement Form.

(ii) 2's Complement Form

The advantage of performing ~~of~~ substraction by complement method is reduction in hardware. (Instead of addition and substraction only adding circuit is needed). That is substraction is performed by adders only.

→ If signed bit is zero then the no. is positive and if it is 1 then no. is negative

Representation of signed numbers using 1's and 2's complement.

Finding 1's and 2's Complement.

1. Method 1.

1 0 1 0 1.

0 1 0 1 0 (1's complement)

Just finding out the complement of each single bit

2. Method 2. (Preferred).

$$\begin{array}{r} 1\,1\,1\,1\,1 \\ 1\,0\,1\,0\,1 \\ \hline 0\,1\,0\,1\,0 \end{array}$$

eg.

$$\begin{array}{r} F\;F\;F\,F\,F\,F \\ A\;B\;C\,0\,1\,4 \\ \hline (5\;4\;3\,F\,E\,B.) \end{array}$$

(F = 15).

(R − 1)'s complement

15's Complement

2's Complement:

1. Method 1.

N → Find 1's complement
↓
+1

Eg.

1 0 1 0 1 = 0 1 0 1 0
+1
(0 1 0 1 1) 2's complement.

2. Method 2.

N bit → N+1. rest all zero.

eg.

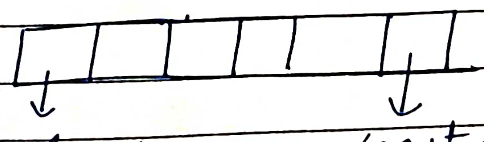
$(10101)_2$ = 5 bit then keep 6^{th} bit 1 and rest all zero.

1 0 0 0 0 0
1 0 1 0 1

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

3. Methode 3. (Not preferred)

1 0 1 0 1
↓ (1's complement).
0 1 0 1 1



Most significant bit (MSB)

Least significant bit (LSB.

* If No. is not ~~negative~~ positive then no need of 2's complement and 1's complement. 421

Q.

No.	Binary equivalent	Sign magnitude	2's complement	1's Complement
~~13~~ ~~1101~~				
13	1101	01101	~~10011~~ 01101	01101
-13	1101	11101	00010	00010

Substraction using complements -

H.W ~~Assignment~~

1. 457.45

$$\begin{array}{r} 457.45 \\ -\ 137.25 \\ \hline 320.20 \end{array}$$

137.25 = ~~00~~1 011 111 :

0.25 × 2 = 0.50

$$\begin{array}{r} 111001001.0111 \\ -\ 100001001.0100 \end{array}$$

8 4 2 1
1 0 1
1 0 0

2.
```
  45
- 32
  13
```

101101
100000

1's complement (-32) : 011111

45 + 1's complement of (-32)

```
  1 0 1 1 0 1
- 0 1 1 1 1 1
 1 0 0 1 1 0 0
 C
          + 1   end around carry
    1 1 0 1
```

If carry is not generated then no. is negative.

2' complement

```
  0 1 1 1 1 1
          + 1
  1 0 0 0 0 0
```

45 + 2' complement of 32.

```
  1 0 1 1 0 1
  1 0 0 0 0 0
 1 0 0 1 1 0 1
```

In 2's complement ~~con~~ substraction method ignore carry and if carry is not generated then no. is negative.

32 16 8 4 2 1 0110100
+1
.1

Page 12

3. 45 101101
−75 1001011.
−13.

1's complement of (−75) = 0110100
2's complement of (−75) = 0110101

Using 1's complement.
45 + 1's complement.

 1 1 1
 1 101101
 0110100
 1100001

1's complement.
0011110.

Using 2's complement.
45 + 2's complement.

 1 1 1 1
 1 0 1 1 0 1
 0 1 1 0 1 0 1
 1. 1 0 0 0 1 0 (Negative no.) (No carry)

2's complement of result = 0011101.
 +1
 0011110

Addition and substraction of 9's complement and 10's complement.

9's complement - 1's complement rule
10's complement - 2's complement rule.

1.

$$\begin{array}{r} 457.45 \\ -137.25 \\ \hline 320.20 \end{array}$$

9's complement -

$$\begin{array}{r} 999.99 \\ -137.25 \\ \hline 862.74 \end{array}$$

~~Add 862.74~~

$$\begin{array}{r} 457.45 \\ 862.74 \\ \hline 1\,320.19 \\ +1 \\ \hline 320.20 \end{array}$$

c

10's complement: (-137.25)

$$\begin{array}{r} 1000.00 \\ -137.25 \\ \hline 862.75 \\ 457.45 \\ \hline \end{array}$$

~~9's complement 320.20~~

1 320.20

c → ignore.

2. -457.45
137.25
-320.20

9's complement
$$\begin{array}{r} 999.99 \\ -\ 457.45 \\ \hline 542.54 \end{array}$$

$$\begin{array}{r} 542.54 \\ 137.25 \\ \hline 679.79 \end{array}$$

Since no carry generated then result is negative. So to find out the result find 9's complement of answer.

$$\begin{array}{r} 999.99 \\ -\ 679.79 \\ \hline 320.20 \end{array}$$

10's complement of (457.45)

$$\begin{array}{r} \cancel{137.} \\ 1\ \ 542.55 \\ 137.25 \\ \hline 0.\ 679.80 \end{array}$$

$$\begin{array}{r} 1000.00 \\ -\ 679.80 \\ \hline 320.20 \end{array}$$

10 have 2 bit so 1 bit is represented in 4 bit so total 8 bit 4221.
choose combination first from right.

BCD (Binary Codes). (Binary Coded Decimal).
Binary course are 4 bit code / representation means 1 no. is represented in 4 bits.
BCD can be weighted and non-weighted.

5	4	2	1
0	0	1	0

BCD 5421

1 Weighted Code.
These can be ⊕ve and ⊖ve. If no sign is defined then it is negative viseversa is positive.

Binary	Decimal	BCD 8421	5421	52-2-1
0000	0	0000	0000	0000
0001	1	0001	0001	0101
0010	2	0010	0010	0100
0011	3	0100	0011	1010
0100	4	0101	0100	1001
0101	5	0110	0101	1000
0110	6	0111	0110	1110
0111	7	1000	0111	1100
1000	8	1001	1011	X
1001	9	1001	1100	X
1010	10	0001 0000		
		1 0		

Arithmatic in BCD Codes.
Standard code BCD 8421.

1. Addition in 8421 codes.

1
$$\begin{array}{r} 456.57 \\ 324.84 \\ \hline 781.41 \end{array}$$

1-1=1 1+1=0 8421
1

0100 0101 0110·0101 0111
0011 0010 0100·1000 0100
0111 0111 1010·1101 1011
+ 0110 0110 0110
~~Te ren~~ 0111 1000 0000·0100 0001
7 8 1 · 4 1

In illegale code is generated after addition then add 6 (0110) to the illegal code.

Substraction in BCD codes.

345.20 0011 0100 0101 001000
- 120.85 0001 0010 0000 100000
224.35

0011 01100 0101 10010 0000
0001 0010 0000 1000 0101
0010 0010 0100·1001 1011
0110
10100001
0110

0010 0010 0101 0000 0001

1 - 1 = 0

wherever is taken borrow the - 0110.

0011	0100	0101 · 00100000	
0001	0010	0000 · 10000101	
0010	00010	0100 · 10011011	
			-0110-0110
0010	0010	0100	00110101

Q. 305.5
- 168.8
136.7

Using 9's complement.

999.9
- 168.8
831.1

305.5
831.1
(1)136.6
C
+1
136.7

1+1=6 8421

0011 0000 0101 . 0101 (305.5)
+ 1000 0011 0001 . 0001 (
1011 0011 0100 . 0100
0110
10001 0011 0110 . 0110

+1

0001 0011 0100 . 01010111

Q 679.6
−885.9
−206.3

9's complement of −888.5

 999.9
− 885.5
 114.0
 679.6
 793.6
No carry generated so no. is negative.

 999.9
 793.6
− 206.3

```
     1 11
  0110  0111 1001 . 0110
+ 0001  0001 0100 . 0000
  0111  1000 1101 . 0110
             +0110
  0111  1001 0011 . 0110
   7     9    3      6
```

793.6 (9's complement)

No carry generated so the result is in 1's complement.

Since answer is (−206.3).

Q. Using 10's complement.

```
  342.7
- 108.9
  233.8
```

10's complement -

```
  1000.00
-  108.90
   891.1
```

```
  1
  342.7
  891.1
(1)233.8
 C
```

In 10's ignore carry.

8421

111

0011 0100 0010 . 0111

1000 1001 0001 . 0001

1011 1101 0011 . 1000

0110 0110

(1) 0010 0011 0011 . 1000

c 2 3 3 . 8

233.8

Q 206.4

−507.6

−301.2

Using 10's complement

1000.00

−507.60

492.40

206.40

+492.40

698.80

No carry is generated so the answer is negative. again 10's complement.

1000.00

−698.80

301.20

8421
0100

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

classmate
Date
Page 21

~~1 0 0 0 0 0~~

~~1 1~~

492.	0010	0000	0110	.	0100
206	+ 0100	1001	0010	.	0100
	0110	1001	1000	.	1000
			~~0110~~		

Excess - 3 / EX - 3 / X - 3 code.

Excess - 3 and grey code are the example of non- weighted code.
whatever binary equivalent is there add 3 to it is excess - 3 code

Binary	Ex - 3.
0000	0011
0001	0100
0010	0101
0011	0110
0100	0111
0101	1000
0110	1001
0111	1010
1000	1011
1001	1100
1010.	0100 0011

Date 8/11/22
Page 22

Binary

→ Octal (pairing (3 bits from R→L)

MSB ← 001 011 101 → LSB
1 3 5

→ Decimal $1 \times 2^6\ 0 \times 2^5\ 1 \times 2^4\ 1 \times 2^3\ 1 \times 2^2\ 0 \times 2^1\ 1 \times 2^0 = 64+16+8+4+2 = (93)_{10}$.

→ Hexadecimal (pairing (4 bits from R→L).

0101 1101 = 5 13
8421 8421
5D.

Octal.

→ Binary: Write each digit in equivalent 3 bit binary no
$(145)_8 = (001\ 100\ 101)_2$.

→ Decimal
$(145)_8 = 1 \times 8^2\ 4 \times 8^1\ 5 \times 8^0 = (101)_{10}$.

→ Hexadecimal.
0000 0110 0101 = $(065)_{16}$.
8421 8421
No direct conversion first convert in binary and then to hexadecimal.

Binary

→ Octal (pairing (3 bits from R→L)

MSB ← 0 0 1 0 1 1 1 0 1 → LSB

$\underline{001}\ \underline{011}\ \underline{101}$

1 3 5

→ Decimal $1\ 0\ 1\ 1\ 1\ 0\ 1 = 64+16+8+4+2 = (93)_{10}$

$2^6\ 2^5\ 2^4\ 2^3\ 2^2\ 2^1\ 2^0$

→ Hexadecimal (pairing (4 bits from R→L).

$\underline{0\,1\,0\,1}\ \underline{1\,1\,0\,1} = 5\ 13$

8 4 2 1 8 4 2 1 5D.

Octal.

→ Binary : Write each digit in equivalent 3 bit binary no

$(145)_8 = (001\ \underline{100}\ \underline{101})_2$.

→ Decimal

$(145)_8 = 1 \times 8^2 + 4 \times 8^1 + 5 \times 8^0 = (101)_{10}$.

→ Hexadecimal.

$\underline{0000}\ \underline{0110}\ \underline{0101} = (065)_{16}$.

8 4 2 1 8 4 2 1

No direct conversion first convert in binary and then to hexadecimal.

Decimal

→ Binary $(145)_{10} = (10010001)_2$.

2	145	
2	72	1
2	36	0
2	18	0
2	9	0
2	4	1
2	2	0
	1	0

→ Octal.

8	145	
8	18	1
	2	2

$(221)_8$.

→ Hexadecimal. $= (91)_{16}$.

16	145	
	9	1

Hexadecimal.

→ Binary (8 4 2 1. (4 bit representation of each digit)

145.

= 0001 0100 0101

→ Octal. (pair of 3)

~~145~~ Binary → Octal

. 000 101 000 101

0 5. 0 5

→ Decimal.

$$1\,4\,5 = (325)_{10}.$$

$\times 16^2\ \ \times 16^1\ \ \times 16^0$

9/11/22

Division process can be eleminated by the use of substraction

11/2 = 5 ½.

$$\begin{array}{r l}
11 & \\
-2 & —(1) \\
\hline
9 & \\
-2 & —(2) \\
\hline
7 & \\
-2 & —(3) \\
\hline
5 & \\
-2 & —(4) \\
\hline
3 & \\
-2 & —(5) \\
\hline
1 & (\text{reminder})
\end{array}$$

5 ½.

Complement:
- r's — 2, 8, 10, 16
- (r-1)'s — 1, 7, 9, 15

Imp.

Method 1.

1's
10110
01001

Method 2.

$$\begin{array}{r} 11111 \\ -10110 \\ \hline 01001 \end{array}$$

7's complement $(145)_8$

$$\begin{array}{r} 7\ 7\ 7 \\ -1\ 4\ 5 \\ \hline 6\ 3\ 2 \end{array}$$

9's complement $(145)_{10}$

$$\begin{array}{r} 9\ 9\ 9 \\ 1\ 4\ 5 \\ \hline 8\ 5\ 4 \end{array}$$

15's Complement

$$\begin{array}{rrr} 15 & 15 & 15 \\ 1 & 4 & 5 \\ \hline 14 & 11 & 10 \end{array}$$

r's complement (r-1)'s +1.

2's complement
Method 2

01001
+1
01010

If no is 5 bit then take 1 and other all 0.

100000
10110
01010

Method 3.

10110
01010

7's complemen.

1 0 0 0 0 (8)
1 0 0 0 0

10000
7543
0235

16's complement.

10000

BCD 4221
2 0010.

Negative Weighted Code.

	8 4 -2 -1	
0	0 0 0 0	
1	0 1 1 1	(4-3) = 1
2	0 1 1 0	
3.	0 1 0 1	
4.	0 1 0 0	
5	1 0 1 1	
6	1 0 1 0	
7.	1 0 0 1	
8.	1 0 0 0	
9	1 1 1 1	
10	1 1 1 0	

EX-3.

Ex -B.

~~0~~
~~.0~~
~~1~~
2
3

Addition in BCD.

146.7
245.8

392.5.

```
      1 1 1 1 1
  0001 0100 0110 . 0111
+ 0010 0100 0101 . 1000
  0011 1000 1011 . 1111
            (IL)   (IL)
          + 0110 . 0110
  0011 1001 0010 . 0101
   3    9    2      5
```

2.
```
  346.1
  211.2
  134.9.
```

```
                     1110
  0011 0100 0110 . 0001
- 0010 0001 0001 . 0010
  0001 0001 0100   1110
                 - 0110
  0001 0011 0100   1001
```

Arithmatic in complement number/system

```
  35
- 24
  11
```

1's comp. of ~~32~~ (-24)

32 16 8 4 2 1

0011 0101
0010 0010

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

1's complement of (-24) : 1 0 0 1 1 1

  1 1 1
 1 0 0 0 1 1
 1 0 0 1 1 1
1 0 0 1 0 1 0
        +1
~~100~~ 1 0 1 1

Now from 2's complement

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

2's complement of 24 . 1's comp 24 + 1
 1 0 0 1 1 1
       +1
 1 0 1 0 0 0 : 2's complement of 24.

   1 0 0 0 1 1
 + 1 0 1 0 0 0
 1 0 0 1 0 1 1

Ignore carry = 0 0 1 0 1 1 (11).

* .01

In decimal form carry 1 is added in LSB.

$$\begin{array}{r} -35 \\ 24 \\ \hline -11. \end{array}$$

Taking 1's comple. of 35.

$$\begin{array}{r} 011100 \\ +\ 011000 \\ \hline 110100 \\ \hline 001011 \end{array} \quad (-11)$$

* ~~Here~~ 1's comp → 001011

Since carry is not generated so answer is −ve

~~1's co~~

In 2's complement.

Taking 2's complemen of 35.

$$\begin{array}{r} 011100 \\ +1 \\ \hline 011101 \end{array}$$

$$\begin{array}{r} 011101 \\ 011000 \\ \hline 110101 \end{array}$$

1's → 001010

Since carry is not generated:

$$\begin{array}{r} 001010 \\ +1 \\ \hline 001011 \end{array} \quad \text{(2's complement).}$$

Using 9's and 10's complement.

$$\begin{array}{r} 554.63 \\ -\ 435.68 \\ \hline 118.95 \end{array}$$

Using 9's complement

$$\begin{array}{r} 999.99 \\ -\ 435.68 \\ \hline 564.31 \end{array}$$

$$\begin{array}{r} 1 \\ 554.63 \\ +\ 564.31 \\ \hline 1\,118.94 \\ +1 \\ \hline 118.95 \end{array}$$

C

~~001~~
~~0101 0101 0100. 01~~

Using 10's complement.

$$\begin{array}{r} 1000.00 \\ -\ 435.68 \\ \hline 564.32 \end{array}$$

$$\begin{array}{r} 1 \\ 554.63 \\ 564.32 \\ \hline 1\,118.95 \end{array}$$

~~C is generage ignore~~

118.95.

$$\begin{array}{r} 554.63 \\ -\ 435.68 \\ \hline 118.95 \end{array}$$

9's complement.

$$\begin{array}{r} 999.99 \\ -\ 554.63 \\ \hline 445.36 \\ +\ 435.68 \\ \hline 881.04 \end{array}$$

Since no carry is generated

$$\begin{array}{r} 999.99 \\ 881.04 \\ \hline 118.95 \end{array}$$

10' Using 10's complement.

$$\begin{array}{r} 1000.00 \\ -\ 554.63 \\ \hline 445.37 \\ 435.68 \\ \hline 881.05 \end{array}$$

Since no carry is generated

$$\begin{array}{r} 1000.00 \\ -\ 881.05 \\ \hline 118.95 \end{array}$$

* Since 12 is illegal code in BCD

Excess-3 / Ex-3 / X-3 -
It is a non-weighted code is also an sequential code
Sequential code is ~~not~~ none but simply succeeding no. greater than preceding no.

Algebra in Ex-3 code
If the carry is generated while performing addition then 3 is added if no carry is generated then 3 is substracted from the 4bit combination
example-
1. 24
+36
60

```
   1111    111
   0101   0111
 + 0110   0100
   1000   0000
 - 0011 + 0011
   1001   0011
     9      3
```

In Ex-3 9 = 6 and 3 = 0

Substraction in Ex-3

1. 267
- 175
92

If the borrow has been taken then -3 from that bit Otherwise +3.


```
   0101      1001  1010
 - 0100      1010  1000
   0000      1111  0010
   0011     -0011 +0011
   0011      1100  0101
    0          9     2.

    3    12    5
   -3    -3   -3
    0     9    2.
```

Arithmatic in Ex-3 codes using 9's and 10's complement-

```
1.  247 · 1
   -136 · 4
    110 · 7
```

Using 9's Complement

0101 0111 1010 · 0100

Taking 9's Complement of (-136·4)

```
   999 · 9
  -136 · 4
   863 · 5
```

```
    0101   0111   1010 · 0100
  + 1011   1001   0110 · 1000
  (1)0001  0001   0000 · 1100
                             +1
    0001   0001   0000 · 1101
  + 0011  +0011  +0011 · -0011
Ex-3 0100  1100   0011 · 1010
      1      1      0      7.
```

10's complement ignore carry.

Using 10's complement

$$
\begin{array}{r}
247.1 \\
-\ 136.4 \\
\hline
110.7
\end{array}
$$

9's complement +1.

8 6 3 . 6

1 1 1 1   1 1 1 1   1 1

$$
\begin{array}{rrrrr}
 & 0101 & 0111 & 1010 & .\ 0100 \\
+ & 1011 & 1001 & 0110 & .\ 1001 \\
\hline
 & 10001 & 0001 & 0000 & .\ 1101 \\
 & +0011 & +0011 & +0011 & -0011 \\
\hline
 & \cancel{001} & & & \cancel{010} \\
 & 0100 & 0100 & 0011 & .\ 1010 \\
 & 1 & 1 & 0 & .\ 7
\end{array}
$$

EX-3

2.
$$
\begin{array}{r}
136.4 \\
-\ 247.1 \\
\hline
-110.7
\end{array}
$$

Using 9's complement

Taking 9's complement of (−247.1)

$$
\begin{array}{r}
999.9 \\
-\ 247.1 \\
\hline
752.8
\end{array}
$$

1011   1001   0110 . 1000

Using 10's Complement

9's Complement + 1.

7 5 2 . 9

```
              1 1
   0100    0110  1000 . 0111
 + 1010    1000  0101 . 1100
   1110    1110  1111   0011
 - 0011  - 0011 - 0011 + 0011
   1011    1011  1100 . 0110
    11      11    12      6
Ex-3  8     8     9   .   3
```

Answer should be in 10's Complement

```
  1000.0
 - 889.3
   110.7
```

Classification of Codes-

1. Sequential Codes -
The succeeding code word is one binary no. greater the next preceeding code.
Eg. 8 4 2 1, Ex-3 codes are sequential codes where as 5211, 2421, 642-1 are not sequential code.

1.

	BCD 8421	D	Ex-3
0	0000	0	3
1	0001	1	4
2	0010	2	5
3	0011	3	6
4	0100	4	7
5	0101	5	8
6	0110	6	9
7	0111	7	10
8	1000	8	11
9	1001.	9	12.

2.

	BCD 5211	Ex.
0	0000	
1	0001.	
2	0011	3
3	0101	5
4	01	
5		
6		
7		
8		
9.		

Self Complimenting Code
If the code word of can be obtained by simply complementing (9-N)
Eg. Ex-3.

N → code word

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

9 - 0 = 9 : Complement of 0 = 1111
9 - 1 = 8 . Complement of 1 = 1110
9 - 2 = 7.

3. Cyclic code.
The succeeding code will differ only by one bit position from the preceeding code - example - gray code

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

$y = AB + AB$
$y = 11 + 0$

eg. 1). 0001 , 2). 0010
0011 0110.

Gray Code -

It is a non-weighted code or it is also not a BCD code (No 4bit representation).

Binary to Gray code Conversion -

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

eg. 10110
There are 2 methods
2 Method

$$B = 1 \oplus 0 \oplus 1 \oplus 1 \quad 0$$
$$\downarrow \quad \downarrow \quad \downarrow \quad \downarrow \quad \downarrow$$
$$1 \quad 1 \quad 1 \quad 0 \quad 1$$

$$B = 0 \oplus 0 \oplus 0 \oplus 1$$
$$\downarrow \quad \downarrow \quad \downarrow \quad \downarrow$$
$$0 \quad 0 \quad 0 \quad 1$$

Odd harmonics generate while transferring the data using wire.

$$B = 0 \oplus 0 \oplus 1 \oplus 0$$
$$\downarrow \quad \downarrow \quad \downarrow \quad \downarrow$$
$$0 \quad 0 \quad 1 \quad 1$$

$$B = 0 \oplus 0 \oplus 1 \oplus 1$$
$$\downarrow \quad \downarrow \quad \downarrow$$
$$0 \quad 0 \quad 1$$

Decimal	Ex-3	Self Complementary Code
0	0011	1100
1	0100	1011
2	0101	1010
3	0110	1001
4	0111	1000
5	1000	0111
6	1001	0110
7	1010	0101
8	1011	0100
9	1100	0011.

Error Detecting Code -

The ~~codes~~ when the data is being transmitted from sender to receiver then there is a possibility of some distortions in the data due to noise or interference. So the exact data will not be ~~rete~~ retrieved / ~~be~~ received by the receiver. So for detecting the error in the sent data we require error detecting codes. Error detecting codes are nothing but they can ~~be~~ are capable of detect and rectify error from the sent data.

No. of 1's in O.P is odd and even in E.P

Types of error Detector

1. Parity method -
In this technique a parity ~~to~~ bit is added and sent with the data.
The parity can be of two types: odd and even parity.

Data	O.P	E.P.
0010	0	1
0110	1	0
1111	1	0
1011	0	1
1110	0	~~0~~1
1001	1	0.

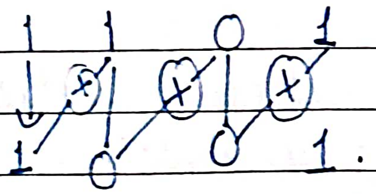
→ No. of data and ~~bo~~ parity bit including ~~n~~ data should be odd in O.P

→ No. of data should be even including ~~paro~~ parity bit in E.P

when two bit changes, even parity bit then we can't identify

16/11/23 2. Check sum method.

★ Grey to Binary.



2 Checksum Method.

~~Whene~~ Whenever there is change in 2 bit it ~~so~~ then we can't identify so we use checksum method.

	O	E
1011	0	1

Sent data

O	E	
0	1	1001.

Received.
identified. (Parity).

Every Time we are sending the data it will add on data. If the sum is same when sent and received is called checksum method.
~~In checks~~
In checksum method the data is sent in a sequential manner once the data is sent / has been completed the sum of all the data sent is matched with the sender and receiver.

Sent		Received
10	→	10
15	→	15
5	→	5
20	→	20
25	→	25
75	→	75

There is demerit in checksum method which ~~ra~~ rarely occurs (doesn't occur in 90% cases).
Suppose - Sent data and received have equal sum but there is error in 2 data.
This method is not applicable in such cases.

Sent		Received
10	→	9
5	→	5
15	→	16
20	→	20
25	→	25
75	→	75

3. Block Parity -

						0
	1	0	0	1	0	1
	1	1	0	1	1	1
	1	0	0	0	1	1
	1	1	1	1	1	0
	1	0	1	0	1	0
0	0	1	0	1	0	X.

Fractional part is represented by - Fixed Point and Floating Point.

When we deal with binary the 2^, decimal 10^, Octal 8^ and hexadecimal 16^

_ _ _ _ (4 bit system).

↑ sign bit   −7 to 7

$2^3 = 8$ (0 − 7) 8 no.

_ _ _ _ _

↑ sign bit   −16 to 15

$2^4 = 16.$

Formula: $-(2^n-1)$ to (2^n-1)

32 bit no., 1 bit sign bit, 5 mantissa,
26 bit exp

__ _ _ _ _ _
↓
sign bit -31 to 31.

$2^5 = 32$

exponent can be

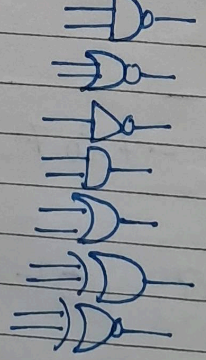
Fixed point - decimal point is fixed
Floating point - decimal point can be floated or shifted

eg. 2.6×10^6
2.4×10^5

Basic Exclusive and Universal Gates.

Signed Rep.

7400 →	NAND	
7402 →	NOR	
7404 →	NOT	;-
7408 →	AND	·
7432 →	OR	+
7486 →	X-OR.	⊕
	X-NOR	⊙.



Formula: $-(2^n-1)$ to (2^n-1)

32 bit no. , 1 bit sign bit, 5 mantissa, 26 bit exp.

__ __ __ __ __ __

↓
sign bit -31 to 31.

$2^5 = 32$

Exponent can be

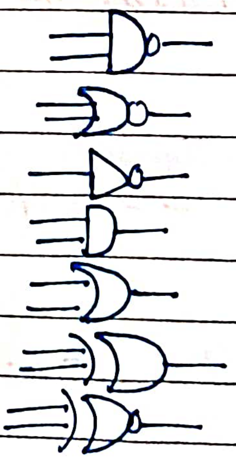
Fixed point – decimal point is fixed
Floating point – decimal point can be floated or shifted

eg. 2.6×10^6
2.4×10^5

Basic Exclusive and Universal Gates.

Signed Rep.

7400 → NAND
7402 → NOR
7404 → NOT   ; —
7408 → AND   .
7432 → OR   +
7486 → X-OR. $\oplus$
X-NOR $\odot$.



All these gates are known as basic gates except NANA and NOR because these are called universal gates.
Universal gates are the gates from which ~~a~~ basic gates can be derived.

Exclusive gate - i) EX-OR $\oplus$
ii) EX-NOR. $\ominus$.

Q. Assume your DOB as 0812 2022
Num 1 Num 2
Perform following operations
1. Num 1 - Num 2 in BCD using 9's complement.
2. Num 2 - Num 1 in BCD using 10's complement.
3. Num 1 - Num 2 in X-3 using 10's complement.
4. Num 2 - Num 1 in X-3 using 9's complement.

DOB = 0401 2003
Num 1 Num 2

1.
$$\begin{array}{r} 0401 \\ -2003 \\ \hline -1602 \end{array}$$

Using 9's complement.
Taking 9's complement of (-2003)

$$\begin{array}{r} 9999 \\ -2003 \\ \hline 7996 \end{array} \qquad \begin{array}{r} \overset{1}{0}401 \\ +7996 \\ \hline 8397 \end{array}$$

$$\begin{array}{r} 9999 \\ -8397 \\ \hline 1602. \end{array}$$

```
  0000   0100   0000   0001
+ 0111   1001   1001   0110
  0111   1101   1001   0111
        +0110
  1000   0011   1001   0111
   8      3      9      7
```

Carry is not generated so answer is negative

2.
```
  2003
- 0401
  1602
```

Using 10's complement:

```
  10000        2003
 -0401         9599
  9599       (1)1602
               C
```

Since carry is generated so ignore carry and answer is positive.

```
   0010   0000   0000   0011
 + 1001   0101   1001   1001
  11011   0101   1001   1100
  +0110                +0110
 (1)0001  0101   1010   0010
 C (Ignore)             +0110
   0001   0110   0000   0010
    1      6      0      2
```

1 - 0 = 1.

3. 0 4 0 1
-2 0 0 3
-1 6 0 2.

Using Ex-3 code.

Using 10's complement.

Taking 10's complement of (-2003)

1 0 0 0 0
- 2 0 0 3
7 9 9 7.

0011	0111	0011	0100
+ 1010	1100	1100	1010
1110	0011	1111	1110
-0011	+0011	-0011	-0011
1011	0110	1100	1011
11	6	12	11

8 1 0 0 0 0
- 8 3 9 8
1 6 0 2.

4. 2 0 0 3
-0 4 0 1
1 6 0 2

By 9's complement.

Taking 9's complement of (-0401)

9 9 9 9
- 0 4 0 1
9 5 9 8

```
 1          111 ← 1111 ← 11
 0101       0011    0011    0110
+1100       1000    1100    1011
1 0001      1100    0000    0001
                               +1

 11                          1
 0001       1100    0000    0010
+0011      -0011   +0011   +0011
 0100       1011    0011    0101
           +0110
 0101       0001    0011    0101

 111        111 ← 1111 ← 11
 0101       0011    0011    0110
+0011       1000    1100    1011
 1000       1100    0000    0001
-0011      -0011   +0011   +0011
 1011       1011    0011    0100
 8          8       0       1
```