Chap-03

Number System

Number System (সংখ্যা পদ্ধতি)

কোন পরিমাপ বা পরিমান কে লিখিতরূপে প্রকাশের জন্য যেসকল চিহ্ন ব্যবহার করা হয়, তা ব্যবহারের নিয়মাবলী কে সংখ্যা পদ্ধতি বলা হয়।

Example : Arabic, Roman, English, Bangla number System

Digit (অংক)

কোন সংখ্যা পদ্ধতি তে ব্যবহৃত চিহ্ন বা প্রতিক কে অংক বলা হয়।

Example : 0, 1, 2, 3, 4, 5, 6, 7, 8, 9

Base (ভিত্তি)

কোন সংখ্যা পদ্ধতি তে ব্যবহৃত মোট চিহ্ন বা প্রতিক বা ডিজিট কে ভিত্তি বলা হয়।

Example : decimal number system base 10,

Binary number system base 2.

Number (সংখ্যা)

ডিজিট এক বা একাধিক পাশাপাশি বসে মান ও অর্থ প্রকাশ করলে, তাকে সংখ্যা বলা হয়।

Example : Page 15, Wings Books

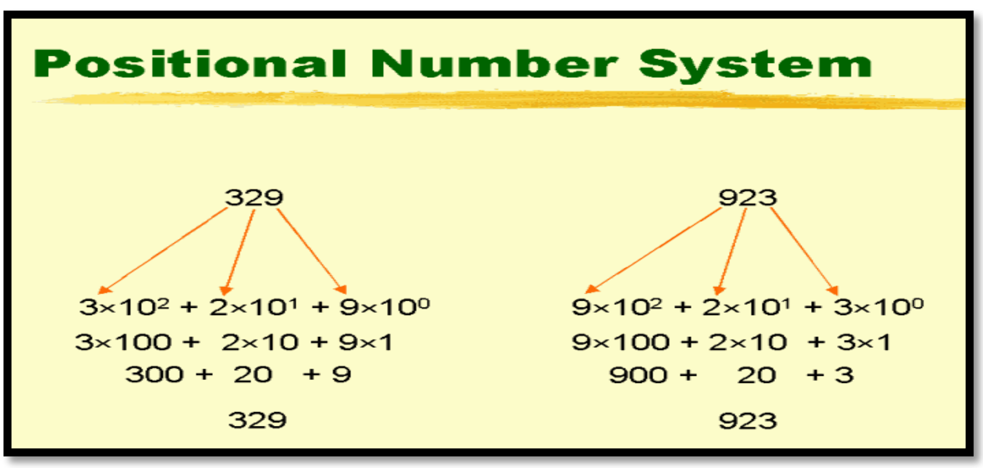
সংখ্যার গঠন

Integer	Radix Point	Fraction
23	.	75

Example : 23.75

Positional NS

- \# অংকের স্থানীয় মান থাকে
- \# অর্থ বা মান প্রকাশ করে
- \# সংখ্যা লিখতে ব্যবহৃত হয়
- \# প্রচলিত অংক ব্যবহৃত হয়



Non Positional NS

- \# অংকের স্থানীয় মান থাকে না
- \# প্রচলিত অর্থ বা মান প্রকাশ করে না
- \# কোড লিখতে ব্যবহৃত হয়
- \# যেমন : আইসিটি বিষয় কোড -- ২৭৫, *৫৬৬#

Type of Number System

There are four categories number system is used in computer system for mathematical operation. These are

1. Binary Number System
2. Octal Number System
3. Decimal Number System
4. Hexadecimal Number System

Number System	**Base**	**Digit**	**Identify**	**Example**
Binary	2	0,1	B_2	101_2
Octal	8	0,1,2,3,4,5,6,7	B_8	257_8
Decimal	10	0,1,2,3,4,5,6,7,8,9	B_{10}	1025_{10}
Hexadecimal	16	0,1,2,3,4,5,6,7,8,9, A,B,C,D,E,F A=10, B=12, C=13, D=14, E=15, F=16	B_{16}	$12AD_{16}$

Conversion

Basic Rules

Method 01 :

B_{10}	to	B_2 /B_8/B_{16} /Any NS

Method 02:

B_2 /B_8/B_{16} /Any NS	to	B_{10}

Method 03 :

B_2	to	B_8 /B_{16}

Method 04:

B_8 /B_{16}	to	B_2

Decimal to Binary

Given That $(156.75)_{10}$

Here Integer Part 156_{10}

```
2)156
 2)78-----0 LSB
 2)39-----0
 2)19-----1
  2)9-----1
  2)4-----1
  2)2-----0
  2)1-----0
    0-----1 MSB
```

$156_{10} = (1001110)_2$

Here Fraction Part 0.75_{10}

$.75 \times 2 = 1.50$ Integer 1 MSB

$.50 \times 2 = 1.00$ Integer 1

$.00 \times 2 = 0.00$ Integer 0 LSB

$0.75_{10} = (0.110)_2$

Result $156.75_{10} = (1001110.110)_2$

Decimal to Octal

Given That $(156.75)_{10}$

Here Integer Part 156_{10}

```
8)156
 8)19-----4 LSB
 8)2------3
   0------2 MSB
```

$156_{10} = (234)_8$

Here Fraction Part $(0.75)_{10}$

$.75 \times 8 = 6.00$ Integer 6 MSB

$.00 \times 8 = 0.00$ Integer 0 LSB

$0.75_{10} = (0.60)_8$

Result $156.75_{10} = (234.6)_8$

Decimal to Hexadecimal

Given That $(156.75)_{10}$

Here Integer Part 156_{10}

```
16)156
 16)9----12(C)  LSB
    0-----9    MSB
```

$156_{10} = (9C)_{16}$

Here Fraction Part 0.75_{10}

$.75 \times 16 = 12.00$ Integer 12 (C) MSB

$.00 \times 16 = 0.00$ Integer 0 LSB

$0.75_{10}=(0.C)_{16}$

Result $156.75_{10}= (9C.C)_{16}$

Binary To Decimal

Given That 10111.11_2

$$\begin{array}{cccccccc} 1 & 0 & 1 & 1 & 1 & . & 1 & 1 \\ \times & \times & \times & \times & \times & & \times & \times \\ 2^4 & 2^3 & 2^2 & 2^1 & 2^0 & & 2^{-1} & 2^{-2} \end{array}$$

$= (1\times16) + (0\times8) + (1\times4) + (1\times2) + (1\times1) + (1\times1/2) + (1\times1/4)$

$= 16 + 0 + 4 + 2 + 1 + 0.50 + 0.25$

$= 23.75_{10}$

Octal To Decimal

Given That 72.6_8

$$\begin{array}{cccc} 7 & 2 & . & 6 \\ \times & \times & & \times \\ 8^1 & 8^0 & & 8^{-1} \end{array}$$

$$\begin{array}{lllllll} = & (7\times8) & + & (2\times1) & + & (6\times1/8) \\ = & 56 & + & 2 & + & 0.75 \\ = & 58.75_{10} \end{array}$$

Hexadecimal to Decimal

Given That $9A.4D_{16}$

9	A	.	4	D
$\times$	$\times$		$\times$	$\times$
16^1	16^0		16^{-1}	16^{-2}

$= (9\times16) + (10\times1) + (4\times1/16) + (13\times1/256)$

$= 144 + 10 + 0.25 + 0.0508$

$= 154.3008_{10}$

Binary to Octal

Given That $(11011101101.11)_2$

$(11 \quad 011 \quad 101 \quad 101 \;.\; 11)_2$

$= (011 \quad 011 \quad 101 \quad 101 \;.\; 110)_2$

$= (3 \quad 3 \quad 5 \quad 5 \;.\; 3)_8$

$= (3355.3)_8$

Binary to Hexadecimal

Given That $(11011101101.11)_2$

	$(110$	1110	1101	.	$11\ )_2$
=	$(0110$	1110	1101	.	$1100\ )_2$
=	$(6$	$14(E)$	$13(D)$	.	$12(C)\ )_{16}$
=	$(6ED.C)_{16}$				

Octal to Binary

Given that $(64.73)_8$

6	4	.	7	3
110	010		111	011

= $(110010.111.011)_2$

Hexadecimal to Binary

Given that $(9A.4F)_{16}$

9	A	.	4	F
1011	1010		0100	1111

= $(10111010.01001111)_2$

Binary Subtraction (Unsigned Number)

[1's Compliment]

Given that 10101010 - 111000

= 10101010 - 111000

= 10101010 - 00111000

=10101010 + (- 00111000)

Now 00111000 = 11000111

[1's Compliment]

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

Carry

Carry = 1, So it will be added with right most bit of result

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

Result : 0 1110010

Binary Subtraction (Unsigned Number)

[2's Compliment]

Given that 10101010 - 111000

= 10101010 - 111000

= 10101010 - 00111000

=10101010 + (- 00111000)

Now ,

00111000	=	1 1 0 0 0 1 1 1	[1's Compliment]
		+ 1	
		1 1 0 0 1 0 0 0	[2's Compliment]

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

Carry

Carry = 1, So we can ignore it

Result : 0 1110010

Binary Subtraction (Unsigned Number)

[1's Compliment]

Given that 10101010.11 – 111000.101

= 10101010.11 – 111000.101

= 10101010.110 – 00111000.101

=10101010.110 + (- 00111000.101)

Now

00111000.101 = 11000111.010 [1's Compliment]

```
            1 0 1 0 1 0 1 0. 1 0 1
   +        1 1 0 0 0 1 1 1. 0 1 0
-------------------------------------
       1    0 1 1 1 0 0 0 1. 1 1 1
Carry
```

Carry = 1, So it will be added with right most bit of integer part of result

```
 0 1 1 1 0 0 0 1    . 1 1 1
           +    1
-----------------------------
 0 1 1 1 0 0 1 0    . 1 1 1
```

Result : 0 1110010. 111

Binary Subtraction (Unsigned Number)

[2’s Compliment]

Given that 10101010.11 – 111000.101

= 10101010.11 – 111000.101

= 10101010.110 – 00111000.101

=10101010.110 + (- 00111000.101)

Now

00111000.101 = 1 1 0 0 0 1 1 1. 0 1 0 [1’s Compliment]

+ 1

1 1 0 0 1 0 0 0. 0 1 0 [2’s Compliment]

		1 0 1 0 1 0 1 0	.	1 0 1
+		1 1 0 0 1 0 0 0	.	0 1 0
	1	0 1 1 1 0 0 1 0	.	1 1 1

Carry

Carry = 1, So we can ignore it

Result : 0 1110010.111

Binary Subtraction (Signed Number)

[1's Compliment]

Given That $(+5)_{10} + (-3)_{10}$

+5 = 0   0 0 0 0 1 0 1
+3 = 0   0 0 0 0 0 1 1
-3 = 1   1 1 1 1 1 0 0 [1's Compliment]

+5 =		0	0 0 0 0 1 0 1
-3 =		1	1 1 1 1 1 0 0
+2 =	1	0	0 0 0 0 0 0 1

Carry

Carry = 1, So it will be added with right most bit of result

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

Result : 0   0 0 0 0 0 1 0

Binary Subtraction
(Signed Number)

[1's Compliment]

Given That $(+5)_{10}$ - $(+3)_{10}$

$(+5)_{10} - (+3)_{10} = (+5)_{10} + (-3)_{10}$

+5 = 0 0 0 0 0 1 0 1
+3 = 0 0 0 0 0 0 1 1
-3 = 1 1 1 1 1 1 0 0 [1's Compliment]

+5 =		0	0 0 0 0 1 0 1
-3 =		1	1 1 1 1 1 0 0
+2 =	1	0	0 0 0 0 0 0 1

Carry

Carry = 1, So it will be added with right most bit of result

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

Result : 0 0 0 0 0 0 1 0

Binary Subtraction (Signed Number)

[1’s Compliment]

Given That $(-5)_{10} + (-3)_{10}$

+5 = 0 0 0 0 0 1 0 1
-5 = 1 1 1 1 1 0 1 0 [1’s Compliment]

+3 = 0 0 0 0 0 0 1 1
-3 = 1 1 1 1 1 1 0 0 [1’s Compliment]

-5 = 1 1 1 1 1 0 1 0
-3 = 1 1 1 1 1 1 0 0

1 1 1 1 1 0 1 1 0

Carry

Carry = 1, So it will be added with right most bit of result

1 1 1 1 0 1 1 0
+ 1

1 1 1 1 0 1 1 1

Result : 1 1 1 1 0 1 1 1= -8

Binary Subtraction (Signed Number)

[2's Compliment]

Given That $(-5)_{10} + (-3)_{10}$

+5	= 0	0 0 0 0 1 0 1		
-5	= 1	1 1 1 1 0 1 0	[1's Compliment]	
		+ 1		
-5	= 1	1 1 1 1 0 1 1	[2's Compliment]	

+3	= 0	0 0 0 0 0 1 1	
-3	= 1	1 1 1 1 1 0 0	[1's Compliment]
		+ 1	
-3	= 1	1 1 1 1 1 0 1	[2's Compliment]

-5 =		1	1 1 1 1 0 1 1
-3 =		1	1 1 1 1 1 0 1
	1	1	1 1 1 1 0 0 0

Carry

Carry = 1, So we can ignore it

Result : 1 1 1 1 1 10 0 0= -8

8 bits (1 Byte Register) for Signed Number

Structure

Sign bit	Value bits						

Sign Bit = 0 Positive
Sign Bit = 1 Negative

Positive Number (Lowest Values)

0	0	0	0	0	0	0	0

= + 0

0	0	0	0	0	0	0	1

= + 1

0	0	0	0	0	0	1	0

= + 2

Positive Number (Highest Values)

0	1	1	1	1	1	1	1

= + 127

Range

+0 to +127 = 128 values

Negetive Number (Highest => Lowest Values)

1	0	0	0	0	0	0	0

= - 0   or   128   or   -128

Not acceptable,   8 cell = value

Negetive Number (Highest Values)

1	0	0	0	0	0	0	1

= - 1

1	0	0	0	0	0	1	0

= - 2

1	1	1	1	1	1	1	1

= - 127

Range

-1   to   -128   =   128 values

Code

বর্ণ, চিহ্ন, অংক কে যন্ত্রেও বোধগম্য ভাষায় অর্থাৎ বাইনারিতে রূপান্তরের জন্য কোড ব্যবহৃত হয়।

BCD Code

- # Binary Coded Decimal
- # 4 bit Code
- # it is used to coding 0-9
- # Pattern 8421, 5421, 7421, 6423

Example

2	1		
0	0	=	0
0	1	=	1
1	0	=	2
1	1	=	3

4	2	1		
0	0	0	=	0
0	0	1	=	1
0	1	0	=	2
0	1	1	=	3
1	0	0	=	4
1	0	1	=	5
0	1	1	=	6
1	1	1	=	7

ASCII Code

American Standard Code for information Interchange
7 bit (ASCII -7) or 8 bit (ASCII -8) Code
it is used to coding 0-9, A-Z, a-z, # % & * (< etc
Unique Code 2^7 or 2^8
Pattern

ASCII -7 :

Zone Bits 03	Value Bits 04	Total 07 bit
100	1101	M

ASCII -8 :

Parity Bit 01	Zone Bits 03	Value Bits 04	Total 08 bit
0	100	1101	M

EBCDIC Code

- # Extended Binary Coded Decimal Information Code
- # 8 bit Code
- # it is mainly for IBM Machine
- # it is used to coding 0-9, A-Z, a-z, # %&*(< etc
- # Unique Code 2^8
- # Pattern

Zone Bits 04	Value Bits 04	Total 08 bit
1101	0100	M

Unicode

- # Universal code
- # 16 bit Code
- # Unique Code 2^{16}