Number system

Decimal number system:

In this system, the base is equal to 10, because there are altogether ten symbols or digits (0, 1, 2, 3, 4, 5, 6, 7, 8, 9).

The weights for whole numbers are positive powers of 10 that increase from R to L, beginning with $10^0 = 1$.

$L \longleftarrow R$

$\ldots \quad 10^3 \quad 10^2 \quad 10^1 \quad 10^0$

For fractional numbers, the weights are negative powers of 10 that decrease from L to R beginning with 10^{-1}.

$L \longleftarrow R \qquad L \longrightarrow R$

$10^3 \; 10^2 \; 10^1 \; 10^0 \; . \; 10^{-1} \; 10^{-2} \; 10^{-3} \; \ldots$

↑ decimal point.

The value of a decimal number is the sum of the digits after each digit has been multiplied by its weight.

eg1: Express the decimal number ... as a sum of the values of each digit ?.

$47 = 4 \times 10^1 + 7 \times 10^0$

$= 4 \times 10 + 7 \times 1$

$= 40 + 7$

$= \underline{\underline{47}}$

eg2: Express the decimal number 568.23 as a sum of the values of each digit ?.

$568.23 = 5 \times 10^2 + 6 \times 10^1 + 8 \times 10^0 + 2 \times 10^{-1} + 3 \times 10^{-2}$

$= 5 \times 100 + 6 \times 10 + 8 \times 1 + 2 \times \frac{1}{10} + 3 \times \frac{1}{100}$

$= \underline{\underline{500 + 60 + 8 + 0.2 + 0.03}}$

eg3. what weight does the digit 7 have in each of the following number : 1370 ?.

$1370 = 1 \times 10^3 + 3 \times 10^2 + 7 \times 10^1 + 0 \times 10^0$

weight of 7 = $\underline{\underline{10}}$

Q4. Express the decimal number 51 as a sum of the products obtained by multiplying each digit by its appropriate weight.

$$51 = \underline{\underline{5 \times 10^1 + 1 \times 10^0}}$$

Binary number system:

In this system, the base is equal to 2, because there are altogether 2 digits (0 and 1).

the weights for whole numbers are positive powers of 2 that increase from R to L, beginning with $2^0 = 1$.

LSB - Least significant bit

MSB = Most significant bit

$L \longleftarrow R$

$\ldots \quad 2^3 \quad 2^2 \quad 2^1 \quad 2^0$

MSB $\longleftarrow$ LSB

For fractional numbers, the weights are negative powers of 2 that decrease from L to R beginning with 2^{-1}.

$L \longleftarrow R \qquad L \longrightarrow R$

$2^3 \; 2^2 \; 2^1 \; 2^0 \,.\, 2^{-1} \; 2^{-2} \; 2^{-3}$

↑ binary point

MSB $\longleftarrow$ LSB $\qquad$ MSB $\longrightarrow$ LSB eg: Determine the weight of each bit in binary number 1101101.

$\overline{1101101} = 1\times 2^6 + 1\times 2^5 + 0\times 2^4 + 1\times 2^3 + 1\times 2^2 + 0\times 2^1 + 1\times 2^0$

weight:

$1 \Rightarrow 2^0 = 1$

$0 \Rightarrow 2^1 = 2$

$1 \Rightarrow 2^2 = 4$

$1 \Rightarrow 2^3 = 8$

$0 \Rightarrow 2^4 = 16$

$1 \Rightarrow 2^5 = 32$

$1 \Rightarrow 2^6 = 64$

Note:

1) Bits : only two digits 0 and 1.

2) the value of bit is determined by its position in the number.

3) In general, with n bits you can count upto a number equal to $2^n - 1$.

4) Largest decimal number $= 2^n - 1$.

eg: $n = 5$

$2^5 - 1 = 32 - 1 = \underline{\underline{31}}$

Scanned By Camera Scanner

Octal Numbers:

In this system, the base is equal to 8, because there are altogether 8 digits (0, 1, 2, 3, 4, 5, 6, 7).

The weights for whole numbers are positive powers of 8 that increase from R to L, beginning with $8^0 = 1$.

L ⟵ R

$8^3 \quad 8^2 \quad 8^1 \quad 8^0$

For fractional numbers, the weights are negative powers of 8 that decrease from L to R beginning with 8^{-1}.

L ⟵ R   L ⟶ R

$8^3 . 8^2 . 8^1 . 8^0 \bullet 8^{-1} \; 8^{-2} \; 8^{-3}$

↑ octal point

Note:

1) 3 bits are suffient to represent any octal number in binary.
2) Each position in an octal number represent a power of the base 8.

Hexadecimal number:

In this system, the base is equal to 16, because there are altogether 10 digits and 6 alphabets (0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F).

The weights for whole numbers are positive powers of 16 that increase from R to L, beginning with $16^0 = 1$.

$$L \longleftarrow R$$

$$16^2 \quad 16^1 \quad 16^0$$

For fractional numbers, the weights are negative powers of 16, that decrease from L to R beginning with 16^{-1}.

$$L \longleftarrow R \qquad L \longrightarrow R$$

$$16^2 \quad 16^1 \quad 16^0 \;\; . \;\; 16^{-1} \quad 16^{-2} \quad 16^{-3}$$

↑ hexadecimal point

Note:

1) 4 bits ($2^4 = 16$) are sufficient to represent any hexadecimal number in binary.

Inter Conversions of Number System:

1) decimal to binary:

→ using division-by-2 method.

1. dividing decimal no by 2.
2. Record the remainder from step1 as the LSB.
3. Divide the quotient by 2.
4. Record the remainder from step3 as the next digit.
5. Repeat step 3 and 4, recording remainders from R to L, until the quotient becomes zero in step 3. The last remainder to be produced is the MSB.

MSB → LSB

eg!. $(12)_{10} \longrightarrow (\ )_2$?

Remainder

2 | 12 — 0 → LSB
2 | 6 — 0
2 | 3 — 1
2 | 1 — 1
0 ↑ MSB

quotient

$(12)_{10} \rightarrow (1100)_2$

Converting decimal fractions to binary:

1. multiply fraction no. by 2.
2. Record the carry from step 1 as the MSB
3. multiply each resulting fraction part by 2.
4. Record the carry from step3 as the next bit.
5. Repeat step 3 to 4 until the fractional product is zero or until the desired number of decimal places is reached.

   The last carry is the LSB.

eg: $(.3125)_{10} \longrightarrow (\quad)_2$.

MSB $\longrightarrow$ LSB

0.3125×2

MSB — $\boxed{0}.625 \times 2$

$\boxed{1}.25 \times 2$

$\boxed{0}.5 \times 2$

LSB — $\boxed{1}.00$

0101

$(.3125)_{10} \to (.0101)_2$

9. $(45.5)_{10} \longrightarrow (.)_2$

LSB

2	45	— 1
2	22	— 0
2	11	— 1
2	5	— 1
2	2	— 0
2	1	— 1
	0	MSB

$$\begin{array}{r} 0.5 \times \\ 2 \\ \hline \text{MSB } \boxed{1}.00 \end{array}$$

$(45.5)_{10} \longrightarrow (101101.1)_2$

Decimal to octal:

1. Dividing decimal no by 8.
2. Record the remainder from step 1 as the LSB.
3. Divide the quotient by 8.
4. Record the remainder from step 3 as the next digit.
5. Repeat step 3 and 4, recording remainders from R to L, until the quotient becomes zero in step 3. The last remainder to be produced is the MSB.

Scanned By Camera Scanner

eg: $(12)_{10} \longrightarrow (\ )_8$

remainder

$$\begin{array}{r|l} 8 & 12 - 4 \quad \text{LSB} \\ 8 & 1 - 1 \quad \text{MSB} \\ & 0 \end{array}$$

$\therefore (12)_{10} \longrightarrow (14)_8$

converting decimal fractions to octal:

1. multiply fractionno by 8.
2. Record the carry from step 1 as the MSB.
3. multiply each resulting fractionpart by 8.
4. Record the carry from step 3 as the next bit.
5. Repeat step 3 to 4 until the fractional product is zero or until the desired number of decimal places is reached.

The last carry is the LSB.

eg: $.875 \times 8$

MSB $\boxed{7}.00$

$(.875)_{10} \longrightarrow (.7)_8$

eg2: $(45.5)_{10} \longrightarrow (\ \ )_8$

$$\begin{array}{l|l} 8 & 45 - 5 \text{ LSB} \\ 8 & 5 - 5 \\ & 0 \text{ MSB} \end{array} \qquad \begin{array}{r} .5 \times \\ 8 \\ \hline \text{MSB } 4.00 \end{array}$$

$(45.5)_{10} \longrightarrow (55.4)_8$

3) decimal to hexadecimal:

1) Dividing decimal no by 16.
2) Record the remainder from step 1 as the LSB.
3) Divide the quoitent by 16.
4) Record the remainder from step 3 as the next digit.
5) Repeat step 3 and 4, recording remainders from R to L, until the quotient becomes zero in step 3. the last remainder to be produced is the MSB.

eg1: $(12)_{10} \longrightarrow (\ )_{16}$?.

$$16 \underline{|\,12} - 12^{LSB}$$
$$\quad 0$$

10 - A
11 - B
12 - C
13 - D
14 - E
15 - F

$(12)_{10} \longrightarrow (C)_{16}$

converting decimal fractions to hexadecimal:

1. multiply fractions by 16.
2. Record the carry from step 1 as the MSB
3. multiply each resulting fractionalpart by 16.
4. Record the carry from step 3 as the next bit.
5. Repeat step 3 to 4 until the fractional part is zero or until the desired number of decimal places is reached.

   The last carry is the LSB.

eg:
$$\begin{array}{r} .875 \times \\ 16 \\ \hline 14.00 \end{array}$$

$(.875)_{10} \longrightarrow (0.E)_{16}$

eg: $(45.5)_{10} \rightarrow (\ \ )_{16}$

$(45.5)_{10} \longrightarrow (\ \ )_{16}$

$16 \,|\, 45 - 13$ LSB ↑

$16 \,|\, 2 - 2$ MSB

0

$.5 \times 16 = 8.00$

$\therefore (45.5)_{10} \longrightarrow (2D.8)_{16}$

4) binary to decimal:

1. multiply bit into position in binary number.

2. sum the product in step 1.

eg: $(1101101)_2 \longrightarrow (\ \ )_{10}$

$= 1\times2^6 + 1\times2^5 + 0\times2^4 + 1\times2^3 + 1\times2^2 + 0\times2^1 + 1\times2^0$

$= 64 + 32 + 0 + 8 + 4 + 1$

$= (109)_{10}$

$\therefore (1101101)_2 \longrightarrow (109)_{10}$

converting binary fractions to decimal:

eg: $(1101.110)_2 \longrightarrow (\ )_{10}$

$\Rightarrow 1\times2^3+1\times2^2+0\times2^1+1\times2^0 . 1\times2^{-1}+1\times2^{-2}+0\times2^{-3}$

$\Rightarrow 8+4+1 . (0.5)+0.25+0$

$\Rightarrow \cancel{(13.075)_{10}}$

$\Rightarrow (13.75)_{10}$

5) binary to octal:

octal table

octal	Binary		
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

eg: $(10101)_2 \longrightarrow (\ )_8$

010/101

$(10101)_2 \rightarrow (25)_8$

6. binary to hexadecimal:

Hexadecim.	Binary			
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
A 10	1	0	1	0
B 11	1	0	1	1
C 12	1	1	0	0
D 13	1	1	0	1
E 14	1	1	1	0
F 15	1	1	1	1

eg: $(10101)_2 \longrightarrow (\ )_{16}$

0001/0101

$(10101)_2 \longrightarrow (15)_{16}$

7. octal to binary:

[Refer octal table in binary to octal]

eg: $(15)_8 \longrightarrow (\ )_2$

$\longrightarrow (001|101)_2$

eg: $(201)_8 \longrightarrow (\ )_2$?.

$\longrightarrow (010000001)_2$

8. octal to decimal:

eg: $(15)_8 \longrightarrow (\ )_{10}$

$\longrightarrow 1 \times 8^1 + 5 \times 8^0$

$\longrightarrow 8 + 5$

$\longrightarrow (13)_{10}$

eg: $(201)_8 \longrightarrow (\ )_{10}$?.

$\longrightarrow 1 \times 8^2 + 0 \times 8^1 + 2 \times 8^0$

$\longrightarrow 64 + 0 + 2 \Longrightarrow 66$

4) octal to hexadecimal

1. convert octal to binary.

2. binary to hexadecimal.

eg: $(15)_8 \longrightarrow (\,)_{16}$?

$\rightarrow (15)_8 \longrightarrow (\,)_2$

$\longrightarrow (001101)_2$

$\rightarrow (001101)_2 \longrightarrow (\,)_{16}$ 00|1101

$\longrightarrow (0D)_{16}$

eg2: $(13.75)_8 \rightarrow (\,)_{16}$

$\rightarrow (13.75)_8 \longrightarrow (\,)_2$

$\longrightarrow (001011.111101)_2$

$\rightarrow (001011.11|11|01)_2 \longrightarrow (\,)_{16}$ 00|1011.1111|0100

$\longrightarrow (0B.F4)_{16}$

10. hexadecimal to binary:

[refer binary to hexadecimal]

eg: $(15)_{16} \rightarrow (\ )_2$

$\rightarrow (00010101)_2$

eg2: $(12.34)_{16} \rightarrow (00010010.00110100)_2$

$\rightarrow (10010.001101)_2$

11. hexadecimal to decimal

eg: $(15)_{16} \rightarrow (\ )_{10}$

$\rightarrow 1 \times 16^1 + 5 \times 16^0$

$\rightarrow 16 + 5$

$\rightarrow (21)_{10}$

eg: $(12.34)_{16} \rightarrow (\ )_{10}$

$\rightarrow 1 \times 16^1 + 2 \times 16^0 . 3 \times 16^{-1} + 4 \times 16^{-2}$

$\rightarrow 16 + 2 . 3 \times \frac{1}{16} + 4 \times \frac{1}{256}$

Scanned By Camera Scanner

$\rightarrow 18\cdot(0.1875+0.015625)$

$\rightarrow 18\cdot(0.23125)$

$\rightarrow \underline{\underline{(18.23125)_{10}}}$

12. hexadecimal to octal:

1. convert hexadecimal to binary.
2. convert binary to octal.

eg: $(D)_{16} \rightarrow (\ )_8$

$\rightarrow (D)_{16} \rightarrow (\ )_2$

$\rightarrow (1101)_2$

$\rightarrow (1101)_2 \rightarrow (\ )_8$

$\rightarrow \underline{\underline{(15)_8}}$

Binary arithmetic:

1) Binary addition:

A	B	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

eg:
```
  11 +      3 +
  11        3
 ----      ---
 110        6
```

$1\times 2 + 1\times 2^0$

$2 + 1 = 3$

110

$$\begin{array}{r|l} 2 & 6 - 0 \\ 2 & 3 - 1 \\ & 1 \end{array}$$

eg2:
```
  110 +      6
  100        4
 -----      ---
 1010       10
```

$1\times 2^2 + 1\times 2^1 + 0\times 2^0$

$4 + 2 = 6$

$$\begin{array}{r|l} 2 & 10 - 0 \\ 2 & 5 - 1 \\ 2 & 2 - 0 \\ & 1 \end{array}$$

eg3:
```
 (110.101 +
  100.000
 ---------
 1010.101
```

2) Binary subtraction:

A	B	diff.	borrow
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

Scanned By Camera Scanner

eg. 11 − 01 = 10   3 − 1 = 2

eg2: 110 − 100 = 010   6 − 4 = 2

eg3: 110 . 101 − 100 . 000 = 010 . 101   6.5 − 4.0 = 2.5

011 ×
100
111

3) Binary multiplication:

A	B	multipli
0	0	0
0	1	0
1	0	0
1	1	1

0
1
2
3

000

000
001
000

eg: 101 × 111   5 × 7 = 35

```
   101 ×
   111
   101
  1010
 10100
100011
```

carry

3
4
7

4. Binary division:

$\frac{0}{1} = 0$ $\frac{1}{1} = 1$

eg:

```
         0101
    110 |100,001
         000
        --------
         1000
          110
        --------
         00100
           000
        --------
           1001
            110
        --------
           0011
```

eg2:

```
        11
    10 |110
        10
       ----
        010
         10
       ----
          0
```

1's and 2's complements:

1's complement of binary number:

A	compli A
1	0
0	1

eg1: $10110010 \xrightarrow{\text{1's complement}} 01001101$

1's complement subtraction:

a) Larger → smaller:

1. 1's complement of smaller binary number

2. larger +
   complement of smaller
   result

3. Remove the carry and add it to the result
   result +
   1
   final answer

eg1: subtract $(10011)_2$ from $(11001)_2$ using first complement?

1. Largest no = 11001
   smallest no = 10011
   complement of smallest no = 01100

2.
```
  11001 +     31 +
  01100       14
 ------       --
1]00101       45
```

3.
```
 00101 +     5 +
     1       1
 -----       -
 00110       6
```

b) smaller → larger:

1) 1's complement of smaller

2) Add Larger +
1's comple smaller

result

3) 1's complement of result and put opposite sign

eg: subtract 1101 from 1001 using the first complement method.

1. Largest = 1001
smallest = 1101
1's complement of smallest = 0010

2.
$$\begin{array}{r} 1001 + \\ 0010 \\ \hline 1011 \end{array}$$

3. -0100

2's complement of binary number:

$$\boxed{2\text{'s complement} = (1\text{'s complement}) + 1.}$$

eg: Find the 2's complement of 10110010?

1. 1's complement : 01001101

2.
$$\begin{array}{r} 01001101 + \\ 1 \\ \hline \underline{\underline{01001110}} \end{array}$$ 2's complement

2's complement subtraction:

a) Larger → smaller:

1. 2's complement of smaller

2. Larger +
   2's complement of smaller
   ---
   result

3. Discard the carry.

eg: subtract 1011 from 1100 using 2's complement?

1. Larger = 1100
   smaller = 1011
   2's complement of smaller ⇒ 1011
$$\begin{array}{r} 0100 + \\ 1 \\ \hline \underline{\underline{0101}} \end{array}$$

2.
$$\begin{array}{r} 1100 + \\ 0101 \\ \hline 1\,]\,0001 \end{array}$$

3. 0001 // (remove carry)

b) smaller → larger:

1 2's complement of smaller

2 larger +
2's complement of smaller
result

3 2's complement of result and put -ve sign.

eg: subtract 11100 (smaller) from 10011 (larger) using 2's complements

1. Larger = 10011
Smaller = 11100

2's complement of smaller = 00011 + 1 = 00100

$$\begin{array}{r} 00011\ + \\ 1 \\ \hline 00100 \end{array}$$

2.
$$\begin{array}{r} 10011\ + \\ 00100 \\ \hline 10111 \end{array}$$

3.
$$\begin{array}{r} 01000\ + \\ 1 \\ \hline -01001 \end{array}$$