Decimal Numbers

- ▶ These numbers are represented in digits of 0 to 9.
- ▶ In mathematics, we called it as whole numbers.
- ▶ It is also known as Base 10.
- All the work done by humans are in decimal numbers.
 - ► These are easy to operate a we done daily calculations in this number system.

Note: Decimal numbers are positive integers or unsigned numbers.

Binary Numbers

- These numbers are represented in digits of 0 and 1 only.
- ▶ It is also known as Base 2.
- ► All the work done by computer system is in binary numbers.
 - These are difficult to operate by humans as a single number ranges from 8bits to 64 bits so its difficult to remember and to solve daily calculations.

Hexadecimal Numbers

- ▶ These numbers are represented in digits of 0 to 15.
 - ► Here, first ten are the same as decimal numbers while the other six are alphabets used for numbers from 10 to 15 as A to F respectively.
- ▶ It is also known as Base 16.
- ► It is the direct mapping of binary numbers and hence easy to convert from binary.
 - These are comparatively easy to operate by humans than binary as it shortens the length of numbers from 32 bits to 8 hexadigits.
 - ► Each hexadecimal digit corresponds to 4 binary bits.

Hexadecimal Integers

▶ Binary values are represented in hexadecimal.

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

^{© 2017} learncoal.com

Conversion: Decimal Numbers to Binary Numbers

- ▶ Repeatedly divide the decimal integer by 2.
 - Each remainder is a binary digit in the translated value. i.e. $(37)_{10} = (100101)_2$

Division	Quotient	Remainder
37 / 2	18	1
18 / 2	9	0
9/2	4	1
4/2	2	0
2/2	1	0
1/2	0	1

Conversion: Binary Numbers to Decimal Numbers

Weighted positional notation shows how to calculate the decimal value of each binary bit:

()
$$_{10} = (B_{n-1} \times 2^{n-1}) + (B_{n-2} \times 2^{n-2}) + ... + (B_1 \times 2^1) + (B_0 \times 2^0)$$

B = Binary digit

i.e.
$$(01001001)_2 = (75)_{10}$$

 $(0 \times 2^7) + (1 \times 2^6) + (0 \times 2^5) + (0 \times 2^4) + (1 \times 2^3) + (0 \times 2^2) + (0 \times 2^1) + (1 \times 2^0)$
 $= 0 + 64 + 0 + 0 + 8 + 0 + 0 + 1$
 $= 75$

Conversion: Binary Numbers to Hexadecimal Numbers Vice Versa

- ► Each hexadecimal digit corresponds to 4 binary bits.
- Example: Translate the binary integer 0001011010101011110010100 to hexadecimal:

1	6	A	7	9	4
0001	0110	1010	0111	1001	0100

 $(0001011010100111110010100)_2 = (16A794)_{16}$

Conversion: Decimal Numbers to Hexadecimal Numbers

- ▶ Repeatedly divide the 16.
 - ► Each remainder is a hexa digit in the translated value.
 - \blacktriangleright i.e. $(422)_{10} = (1A6)_{16}$

Division	Quotient	Remainder
422 / 16	26	6
26 / 16	1	A
1 / 16	0	1

Conversion: Hexadecimal Numbers to decimal Numbers

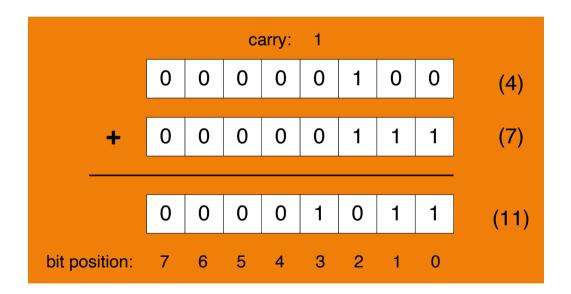
Multiply each digit by its corresponding power of 16:

()
$$_{10} = (D_n \times 16^n) + ...(D_3 \times 16^3) + (D_2 \times 16^2) + (D_1 \times 16^1) + (D_0 \times 16^0)$$

- ► Hex 1234 equals $(1 \times 16^3) + (2 \times 16^2) + (3 \times 16^1) + (4 \times 16^0)$, or decimal 4,660.
- ► Hex 3BA4 equals $(3 \times 16^3) + (11 * 16^2) + (10 \times 16^1) + (4 \times 16^0)$, or decimal 15,268.

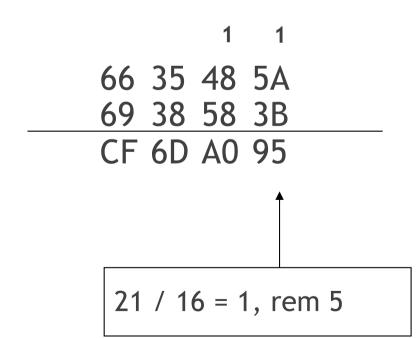
Binary Addition

► Starting with the LSB, add each pair of digits, include the carry if present.



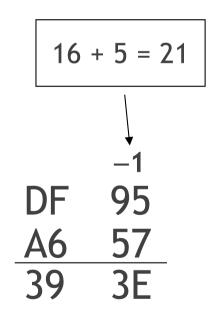
Hexadecimal Addition

- ▶ Divide the sum of two digits by the number base (16).
- ► The quotient becomes the carry value, and the remainder is the sum digit.



Hexadecimal Subtraction

When a borrow is required from the digit to the left, add 16 (decimal) to the current digit's value:



Integer Numbers

- In computer system, programmer should know that the number usage is dependent on the programmers the way they want to use.
- The range of the number varies for signed and unsigned numbers having same size.
- ▶ Integer range is considered to be $-\infty$ to $+\infty$.
 - ▶ While here the range depends on the number of bits used by the data type.
 - ▶ If the data type is BYTE than it uses 8 bits while if it is WORD than it uses 16 bits and so on.
- Note: The smallest memory unit of computer system is BYTE and every BYTE has an address.

Unsigned Integer Numbers

- Unsigned integers are same as Decimal numbers.
- ► There conversion and working is similar to decimal numbers.
- ▶ The range of unsigned integer is from 0 to $+\infty$.
 - ▶ While here the range depends on the number of bits used by the data type.
 - If the data type is BYTE than it uses 8 bits (0 to 2^8 -1) while if it is WORD than it uses 16 bits (0 to 2^{16} -1) and so on.

Signed Integer Numbers

- In computer system, it is known that the most significant bit shows whether the number is positive or negative.
- ► Hence the range of positive number hold by signed number reduces.
 - Here, if the data type is BYTE than it uses 8 bits but range is $(-2^7 \text{ to } 2^7 1)$ while if it is WORD than it uses 16 bits but range is $(-2^{15} \text{ to } 2^{15} 1)$ and so on.
- Consider this number: 11001100. If the programmer uses that value in a unsigned integer than it is a positive number (204) while if the programmer uses it in a signed integer than it is a negative number (-56).

Test your understanding! Solve it.

1. Conversion

- A. $(110100111101)_2 = ()_{10} = ()_{16}$
- B. $(3B0BD1EF)_{16} = ()_{10} = ()_{2}$

2. Operation

- A. Add $(11010110)_2$ and $(10110011)_2$
- B. Subtract (10110011)₂ From (11010110)₂
- c. Add (AC9F2B8E)₁₆ and (7D9B3AC)₁₆
- D. Subtract (7D9B3AC) $_{16}$ From (AC9F2B8E) $_{16}$