BAPUJI POLYTECHNIC SHABANUR 335 PUSHPALATHA B

unit 2 Unit –II (8 Hrs) **Digital principles** **25 marks**

Digital Principles: Definitions- bit, nibble, byte, word, and parity bit. Number system definition, Types, radix, decimal, BCD, binary and hexadecimal.BCD addition.Binary addition, Subtraction, Multiplication, Division, 1's and 2's complement. Hexadecimaladdition, Subtraction, advantages. Conversion- decimal to binary and hexadecimal and viceversa. ASCII, Gray codes, and list applications.

BIT: it is smallest unit of data in a computer .bit has a single binary value '0' or '1'.

Nibble :A group of four bits is called nibble.

Byte and word : A group of eight bits is called byte .A memory unit stores binary information in group of bits called word .

Most computer memories use words that are multiples of bytes in length .

Parity bit : A parity bit or check bit is abit added to the end od string of binary code that indicates whether the number of bits in the string with the value is even or odd .parity bits are used as the simplest form of error detecting code

Number system :A number system is a code that use symbol to refer a number of items or Numbering system

Is a basic form of counting various items

various number system with their radices

→ 1. **Decimal number system → 10 digits → 0-9 base is 10.**

2. **Binary number system → 2 digits → 0 & 1 base is 2.**

3. **Octal number system → 8 digits → 0-8. Base is 8.**

4. **Hexadecimal number system → 16 digits → 0-9 & A-F base is 16**

Binary number system: **binary number system is a base 2 system containing only 2 digits..**

The 2 binary digits are 1& 0 .in binary system; weight is expressed as power of 2.

Decimal number system: **The decimal number system is composed of ten numerals or symbols that mean 0 to 9.**

The decimal system called base 10 system because it has 10 digits here each digit value expressed as a power of 10.

Octal number system :**octal number system has base of eight that means it has eight possible digits 0 to 7**

Here each digit has its own value or weight expressed as power of eight.

Hexadecimal number system::**It has base of 16 having 16 characters. 0 ,1, 2,3,4,5, 6, 7, 8, 9.&A,B,C,D,E,F.here each has its own value or weight expressed as power of 16**

Converting from Decimal to any Radix

To convert a decimal integer number (a decimal number in which any fractional part is ignored) to any other radix, all that is needed is to continually divide the number by its radix, and with each division, write down the remainder. When read from bottom to top, the remainder will be the converted result

DECIMAL TO BINARY

Convert the following numbers from base 10 to base 2-

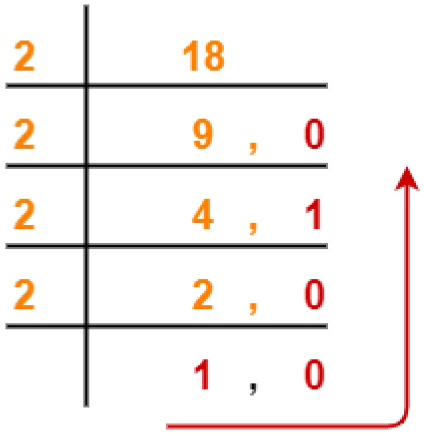
1. $(18)_{10}$
2. $(18.625)_{10}$
3. $(172)_{10}$
4. $(172.878)_{10}$

Solution-

1. $(18)_{10}$

$$(18)_{10} \to (\ ?\ )_2$$

Using division method, we have-

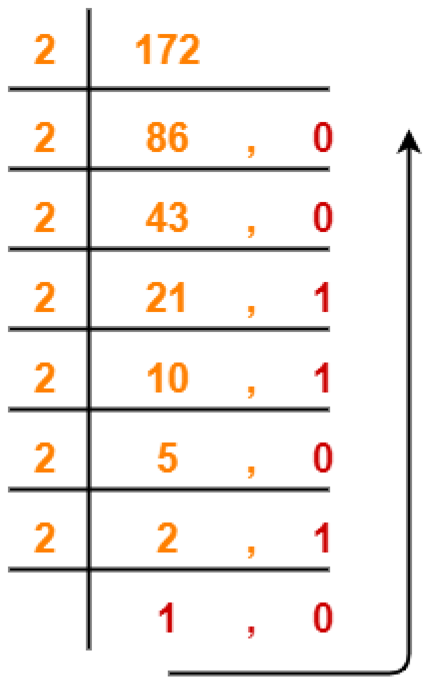


From here, $(18)_{10} = (10010)_2$

2) **$(172)_{10}$**

$(172)_{10} \rightarrow (?)_2$

Using division method, we have-



From here, $(172)_{10} = (10101100)_2$

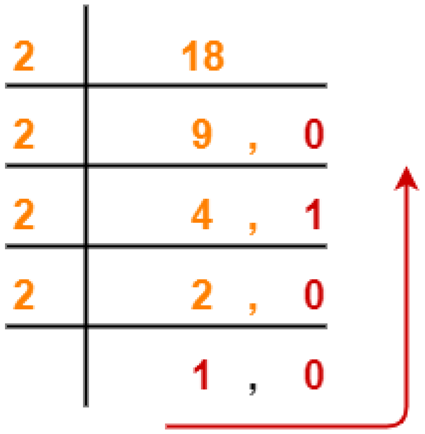
3) $(18.625)_{10}$

$(18.625)_{10} \rightarrow (?)_2$

Here, we treat the real part and fractional part separately-

<u>For Real Part-</u>

- **The real part is $(18)_{10}$**
- **We convert the real part from base 10 to base 2 using division method same as above.**



So, $(18)_{10} = (10010)_2$

For Fractional Part-

- The fractional part is $(0.625)_{10}$
- We convert the fractional part from base 10 to base 2 using multiplication method.

	Real part	Fractional Part
0.625 x 2=1.25→	1	0.25
0.25 x 2=050→	0	0.50
0.50 x 2=1.0→	1	0

- The fractional part terminates to 0 after 3 iterations.
- Traverse the real part column from top to bottom to obtain the required number in base 2.

From here, $(0.625)_{10} = (0.101)_2$

Combining the results of real part and fractional part, we have-

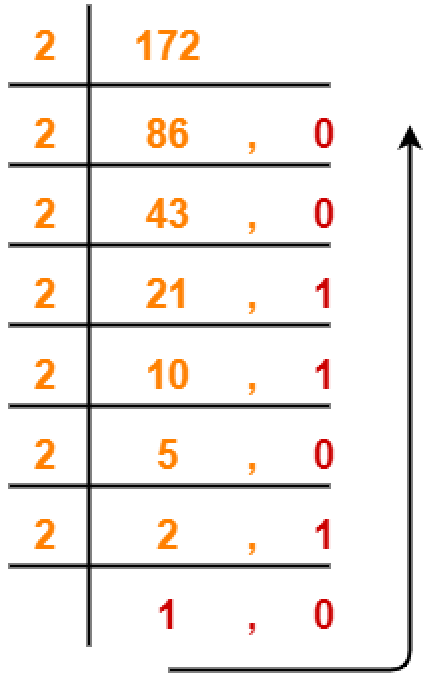
$(18.625)_{10} = (10010.101)_2$

4) $(172.878)_{10}$

$$(172.878)_{10} \rightarrow (\ ?\ )_2$$

Here, we treat the real part and fractional part separately-

- The real part is $(172)_{10}$
- We convert the real part from base 10 to base 2 using division method same as above.



For Fractional Part-

- The fractional part is $(0.878)_{10}$
- We convert the fractional part from base 10 to base 2 using multiplication method.

Using multiplication method, we have-

	Real part	Fractional Part
0.878 x 2=1.756⟶	1	0.756
0.756 x 21.512⟶	1	0.512
0.512 x 2=1.024⟶	1	0.024
0.024 x 2=0.048⟶	0	0.048

- The fractional part does not terminates to 0 after several iterations.
- So, let us find the value up to 4 decimal places.
- Traverse the real part column from top to bottom to obtain the required number in base 2.

From here, $(0.878)_{10} = (0.1110)_2$

Combining the results of real part and fractional part, we have-

$(172.878)_{10} = (10101100.1110)_2$

Solve the below problems

a)93 b)78 c)45 d)25 e) .655 f).98

CONVERTIONS FROM BINARY TO DECIMAL

1. $(10010)_2$

$(10010)_2 \rightarrow (?)_{10}$

Using expansion method, we have-

$(10010)_2$

$= (1 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0)_{10}$

$= (16 + 0 + 0 + 2 + 0)_{10}$

$= (18)_{10}$

2) $10010.101)_2$

$(10010.101)_2 \rightarrow (?)_{10}$

Using expansion method, we have-

$(10010.101)_2$

$= (1 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 + 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3})_{10}$

$= (16 + 0 + 0 + 2 + 0 + 0.5 + 0.125)_{10}$

$= (18.625)_{10}$

CONVERTIONS FROM HEXADECIMAL TO DECIMAL

1) $(AC)_{16}$

$(AC)_{16} \rightarrow (?)_{10}$

Using expansion method, we have-

$(AC)_{16}$

$= (A \times 16^1 + C \times 16^0)_{10}$

$= (10 \times 16 + 12 \times 1)_{10}$

$= (160 + 12)_{10}$

$= (172)_{10}$

.2) $(AC.FBA5)_{16}$

$(AC.FBA5)_{16} \rightarrow (?)_{10}$

Using expansion method, we have-

$(AC.FBA5)_{16}$

$= (A \times 16^1 + C \times 16^0 + F \times 16^{-1} + B \times 16^{-2} + A \times 16^{-3} + 5 \times 16^{-4})_{10}$

$= (10 \times 16 + 12 \times 1 + 15 \times 16^{-1} + 11 \times 16^{-2} + 10 \times 16^{-3} + 5 \times 16^{-4})_{10}$

$= (160 + 12 + 0.9375 + 0.0429 + 0.0024 + 0.0001)_{10}$

$= (172.9829)_{10}$. $(0.ABDF)_{16}$

-

3) $(0.ABDF)_{16} \rightarrow (?)_{10}$

Using expansion method, we have-

$(0.ABDF)_{16}$

$= (0 \times 16^0 + A \times 16^{-1} + B \times 16^{-2} + D \times 16^{-3} + F \times 16^{-4})_{10}$

$= (0 \times 1 + 10 \times 16^{-1} + 11 \times 16^{-2} + 13 \times 16^{-3} + 15 \times 16^{-4})_1$

$= (0 + 0.625 + 0.0429 + 0.0032 + 0.0002)_{10}$

$= (0.6713)_{10}$

- CONVERSIONS FROM OCTAL TO DECIMAL

EXAMPLE:

1. $(254)_8$

$(254)_8 \rightarrow (\ ?\ )_{10}$

Using expansion method, we have-

$(254)_8$

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

$= (128 + 40 + 4\ )_{10}$

$= (172)_{10}$

.2) $(254.7014)_8$

$(254.7014)_8 \rightarrow (?)_{10}$

Using expansion method, we have-

$(254.7014)_8$

$= (2 \times 8^2 + 5 \times 8^1 + 4 \times 8^0 + 7 \times 8^{-1} + 0 \times 8^{-2} + 1 \times 8^{-3} + 4 \times 8^{-4})_{10}$

$= (128 + 40 + 4 + 0.875 + 0.0019 + 0.0009)_{10}$

$= (172.8778)_{10}$

Convert from binary to hexadecimal.

Conversions from binary to hexadecimal is done by grouping the binary Number in to groups of 4 bits starting from LSB and moving towards the MSB for integer part then each group of four bits is replaced by its Hex representation zeros are added as required completing group

Decimal (Base 10)	Binary (Base 2)	Hexadecimal (Base 16)
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

1) 0011010001011010

- 0011 0100 0101 1010

↓ ↓ ↓ ↓

3 4 5 A

∴ $(345A)_{16}$

2) 101011.11011

0010 1011. 1101 1000

↓ ↓ ↓ ↓

2 B D 8

∴(2B.D8)16

3) 1010100011110101.1001

1010 1000 1111 0101. 1001

↓ ↓ ↓ ↓ ↓

A 8 F 5 9

∴ $(A8F5.9)_{16}$

Converting from binary to octal

Using Grouping

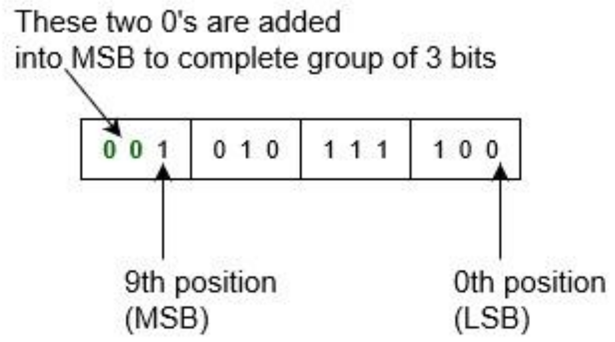
Since, there are only 8 digits (from 0 to 7) in octal number system, so we can represent any digit of octal number system using only 3 bit as following below.

Octal Digit Value	Binary Equivalent
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

So, if you make each group of 3 bit of binary input number, then replace each group of binary number from its equivalent octal digits. That will be octal number of given number number. Note that you can add any number of 0’s in leftmost bit (or in most significant bit) for integer part and add any number of 0’s in rightmost bit (or in least significant bit) for fraction part for completing the group of 3 bit, this does not change value of input binary number

So, these are following steps to convert a binary number into octal number.

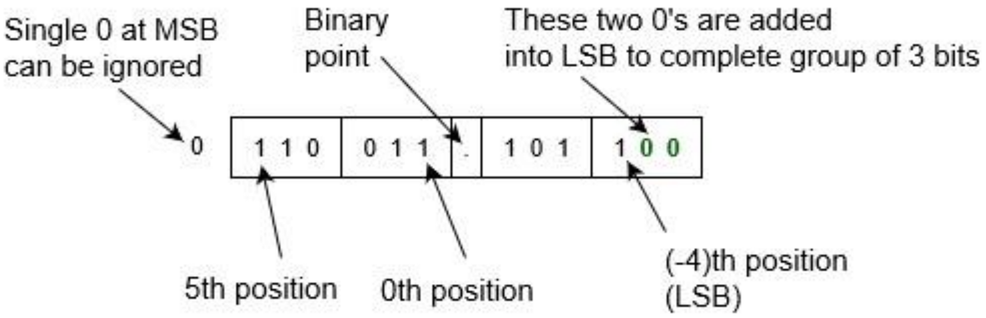
- Take binary number
- Divide the binary digits into groups of three (starting from right) for integer part and start from left for fraction part.
- Convert each group of three binary digits to one octal digit.
- Convert binary number 1010111100 into octal number. Since there is no binary point here and no fractional part. So,



- Therefore, Binary to octal is.

```
= (1010111100)₂
= (001 010 111 100)₂
= (1 2 7 4)₈
= (1274)₈
```

Example-2 Convert binary number 0110 011.1011 into octal number. Since there is binary point here and fractional part. So,



Therefore, Binary to octal is.

```
= (0110 011.1011)₂
= (0 110 011 . 101 1)₂
= (110 011 . 101 100)₂
= (6 3 . 5 4)₈
= (63.54)₈
```

Hexadecimal to binary conversion

To convert the hexadecimal number into binary, we need to represent every hexadecimal digit into 4 binary bits.

Finally, combine the binary bits.

Example

1) Let's convert $(FD)_{16}$ into binary numbers

$(FD)_{16}$

F= (1111)

D= (1101)

$(FD)_{16} = (11111101)_2$

3. Explain

(i) ASCII code (ii) BCD code (iii) Gray code

→ (i) ASCII code.

→ American standard code for information interchange (Abrivated as ASCII) pronounced askey it is widely used for printer key boards and video terminals that interface with small computer system ASCII as 128 characters symbol represented by 7 bit binary code its format is X0 X1 X2 X3 X4 X5 X6 X_7.

(ii) BCD code [binary coded decimal)

BCD numeric code in which each digit of decimal no is represented by a seperate group bits the most common BCD code is 8421 it is a 4 bit code in which each decimal digit 0 to 9 is represented by a 4 bit binary number from 0000 to 1001 each bit is assigned a weight the weights associated with 4 bits are 8421 from left to right. The table shows 4 bits 8421 BCD code used to represent decimal digit

Decimal Digit	BCD			
	8	4	2	1
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

(iii) Gray code:

Gray code is an unweighted code that bit position does not have specific weight assign to them. Gray code is not suitable for arithmetic operations but useful for input and output devices. Analog to digital converts and other peripheral devices. The most important future of gray code is that only a single bit change, occur when going from one code number to next a single bit change property is most important many applications.

Decimal	Binary Code	Gray Code
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
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

.

5. **Explain parity bit.**

   A **parity bit**, or check **bit**, is a **bit** added to a string of binary code. **Parity bits** are used as the simplest form of error detecting code. ... The **parity bit** ensures that the total number of 1-**bits** in the string is even or odd. Accordingly, there are two variants of **parity bits**: even **parity bit** and odd **parity bit**.

Binary arithmetic operations

Binary arithmetic is essential part of all the digital computers and many other digital system.

Binary Addition

It is a key for binary subtraction, multiplication, division. There are four rules of binary addition.

Case	A	+	B	Sum	Carry
1	0	+	0	0	0
2	0	+	1	1	0
3	1	+	0	1	0
4	1	+	1	0	1

In fourth case, a binary addition is creating a sum of (1 + 1 = 10) i.e. 0 is written in the given column and a carry of 1 over to the next column.

Example − Addition

0011010 + 001100 = 00100110

$$
\begin{array}{rl}
1\,1 & \text{carry} \\
0\,0\,1\,1\,0\,1\,0 & = 26_{10} \\
+\,0\,0\,0\,1\,1\,0\,0 & = 12_{10} \\
\hline
0\,1\,0\,0\,1\,1\,0 & = 38_{10}
\end{array}
$$

Binary Subtraction

Subtraction and Borrow, these two words will be used very frequently for the binary subtraction. There are four rules of binary subtraction.

Case	A	-	B	Subtract	Borrow
1	0	-	0	0	0
2	1	-	0	1	0
3	1	-	1	0	0
4	0	-	1	0	1

Example − Subtraction

0011010 - 001100 = 00001110

$$
\begin{array}{rl}
1\;\;1 & \text{borrow} \\
0\,0\,\cancel{1}\,\cancel{1}\,0\,1\,0 & = 26_{10} \\
-\,0\,0\,0\,1\,1\,0\,0 & = 12_{10} \\
\hline
0\,0\,0\,1\,1\,1\,0 & = 14_{10}
\end{array}
$$

Binary Multiplication

Binary multiplication is similar to decimal multiplication. It is simpler than decimal multiplication because only 0s and 1s are involved. There are four rules of binary multiplication.

Case	A	x	B	Multiplication
1	0	x	0	0
2	0	x	1	0
3	1	x	0	0
4	1	x	1	1

Example − Multiplication

Example:

0011010 x 001100 = 100111000

```
        0 0 1 1 0 1 0   = 26(10)
      x 0 0 0 1 1 0 0   = 12(10)
  -----------------
        0 0 0 0 0 0 0
      0 0 0 0 0 0 0
    0 0 1 1 0 1 0
  0 0 1 1 0 1 0
  -----------------
  0 1 0 0 1 1 1 0 0 0   = 312(10)
```

Binary Division

Binary division is similar to decimal division. It is called as the long division procedure.

Example − Division

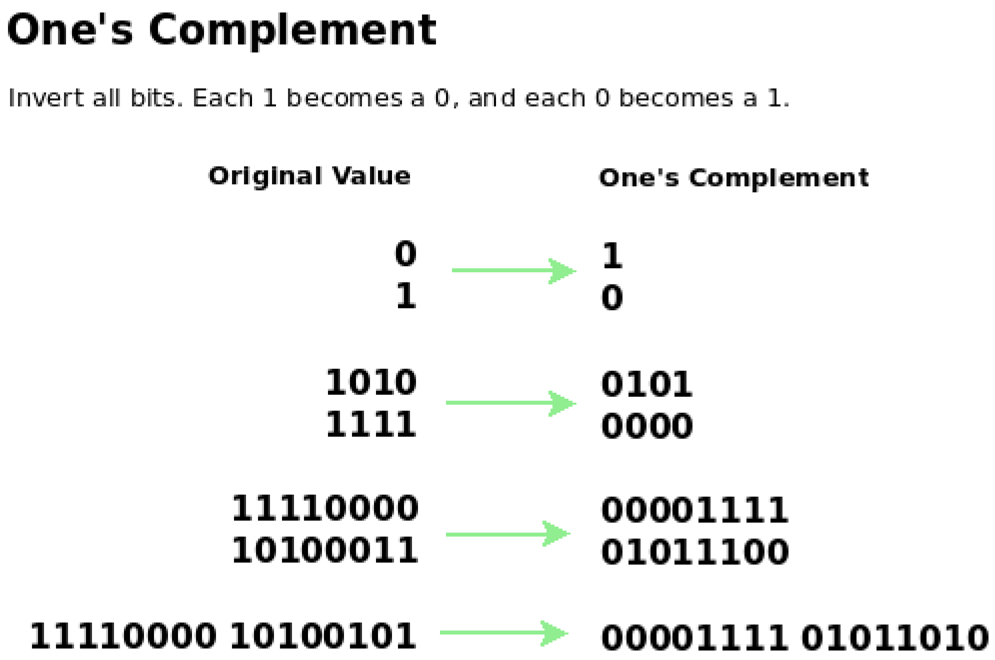
101010 / 000110 = 000111

```
               1 1 1      = 7(10)
          -------------
  000110 ) 1 0 1 0 1 0    = 42(10)
          -1 1 0          = 6(10)
           ------
             1 0 0 1
            -1 1 0
            -------
               1 1 0
              -1 1 0
              ------
                   0
```

Complements are used in digital computers in order to simply the subtraction operation and for the logical manipulations. For the Binary number (base-2) system, there are two types of complements: 1’s complement and 2’s complement.

1’s Complement of a Binary Number

There is a simple algorithm to convert a binary number into 1’s complement. To get 1’s complement of a binary number, simply invert the given number.



Example-1: Find 1’s complement of binary number 10101110.

Simply invert each bit of given binary number, so 1’s complement of given number will be 01010001.

Example-2: Find 1’s complement of binary number 10001.001.

Simply invert each bit of given binary number, so 1’s complement of given number will be 01110.110.

2’s Complement of a Binary Number

There is a simple algorithm to convert a binary number into 2’s complement. To get 2’s complement of a binary number, simply invert the given number and add 1 to the least significant bit (LSB) of given result.

Example-1 − Find 2’s complement of binary number 10101110.

Simply invert each bit of given binary number, which will be 01010001. Then add 1 to the LSB of this result, i.e., 01010001+1=01010010 which is answer.

01. Subtract 10100 from 11001 using 2's compliment.

$\rightarrow$ I's complimented of 10100

$\downarrow$

01011

$$\begin{array}{r} 01011 \\ + \quad 1 \\ \hline 01100 \end{array}$$

$$\begin{array}{r} 01100 \\ +\ 11001 \\ \hline 1]00101 \end{array}$$

Ignore the carry

02. Subtract 10100 from 11001 using 2's compliment.

$\rightarrow$ I's complimented of 10100

$\downarrow$

01011

$$\begin{array}{r} 01011 \\ + \quad 1 \\ \hline 01100 \end{array}$$

$$\begin{array}{r} 01100 \\ +\ 11001 \\ \hline 1]00101 \end{array}$$

Ignore the carry

3) Subtract $(1010)_2$ from $(10110)_2$ using 2s compliment.

→ 1010

↓

0101

+ 1

0110

10110

0110 +

11100

Use the following steps to perform hexadecimal addition:

1. Add one column at a time.
2. Convert to decimal and add the numbers.
3. (a) If the result of **step** two is 16 or larger subtract the result from 16 and carry 1 to the next column.

Oct 22, 2012

Use the following steps to perform hexadecimal addition:

1. Add one column at a time.
2. Convert to decimal and add the numbers.
3. (a) If the result of **step** two is 16 or larger subtract the result from 16 and carry 1 to the next column.

Oct 22, 2012