Introduction

Hexadecimal (or hex) is a system with base 16. The 16 digits used in the hexadecimal system and their equivalent value in binary and in denary systems are shown in the following table.

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

It can be noted from the above table that each nibble (a group of 4 bits) can be represented with 1 digit in the hexadecimal system. It is very convenient to write in numbers in hex compared to the binary system. It is helpful for programmers coding in low-level languages. Instead of 4 bits, it is enough if they type a single character. The memory location in a computer is also stated in hexadecimal form. This makes the address more readable. Consider a memory address A3F581. Is represented in binary, it would be:

1010| 0011| 1111| 0101| 0100| 0001

Converting binary to hexadecimal

Binary numbers can be easily converted into a hexadecimal number. Starting from right to left, the binary numbers are split into groups of 4 bits. If the group has less than 4 bits, zeros are added to the left. Each nibble is then converted to its equivalent hexadecimal number.

Let us convert 10111011001010 to hexadecimal.

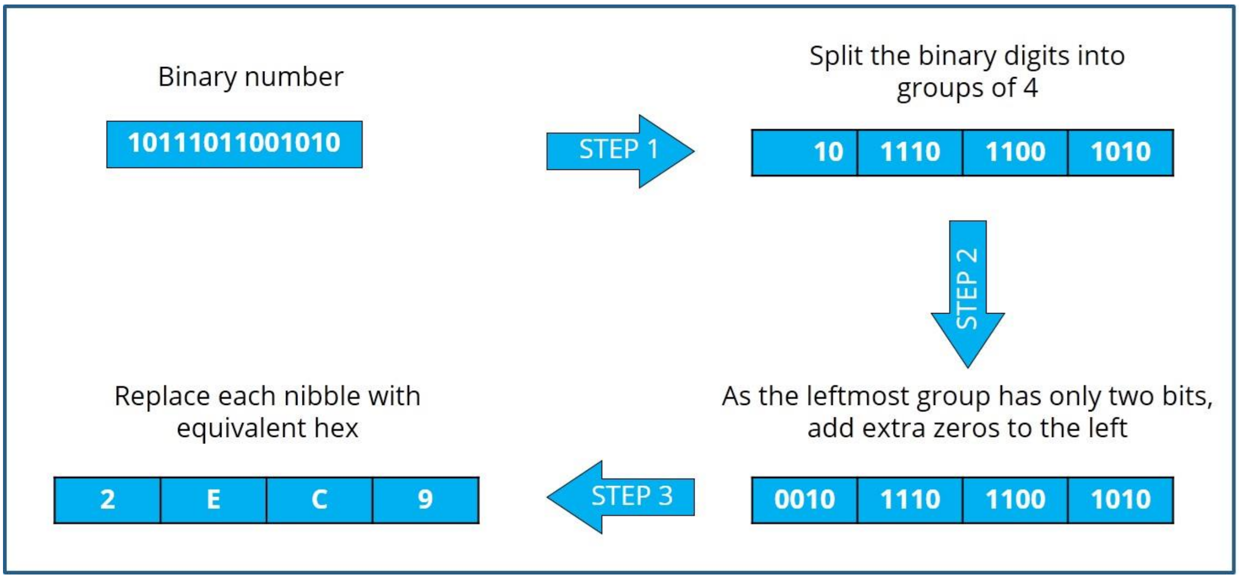

Figure 1: Converting binary numbers to hexadecimal

Converting hexadecimal to binary

Hexadecimal numbers are converted to binary by finding the 4-bit code and writing the nibbles together.

Let us convert 9AF into binary

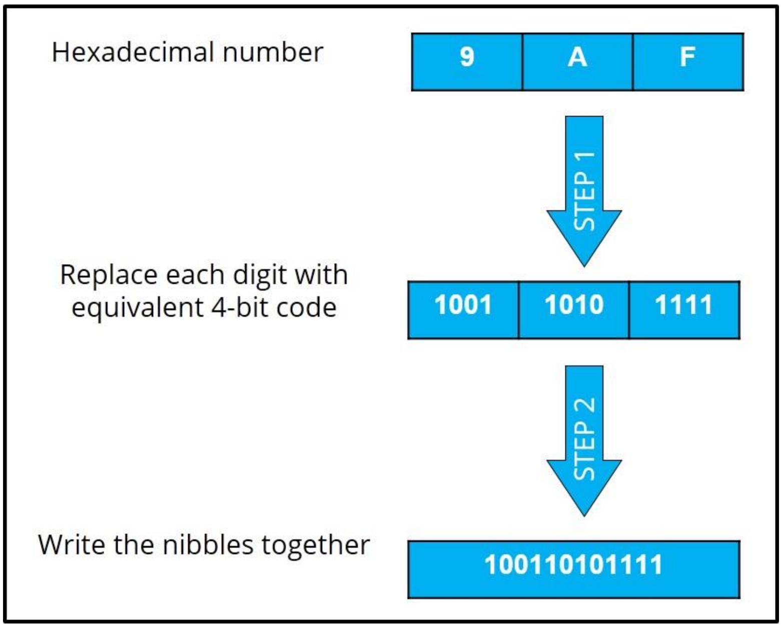

Figure 2: Converting hexadecimal numbers to binary

Converting denary to hexadecimal

To convert a denary number to hexadecimal, the number is converted to binary first. Then, the binary number is converted to the hexadecimal system.

Let us consider converting the number 14. The binary form is 1110, which is E in the hexadecimal system.

Let us consider the denary number 131.

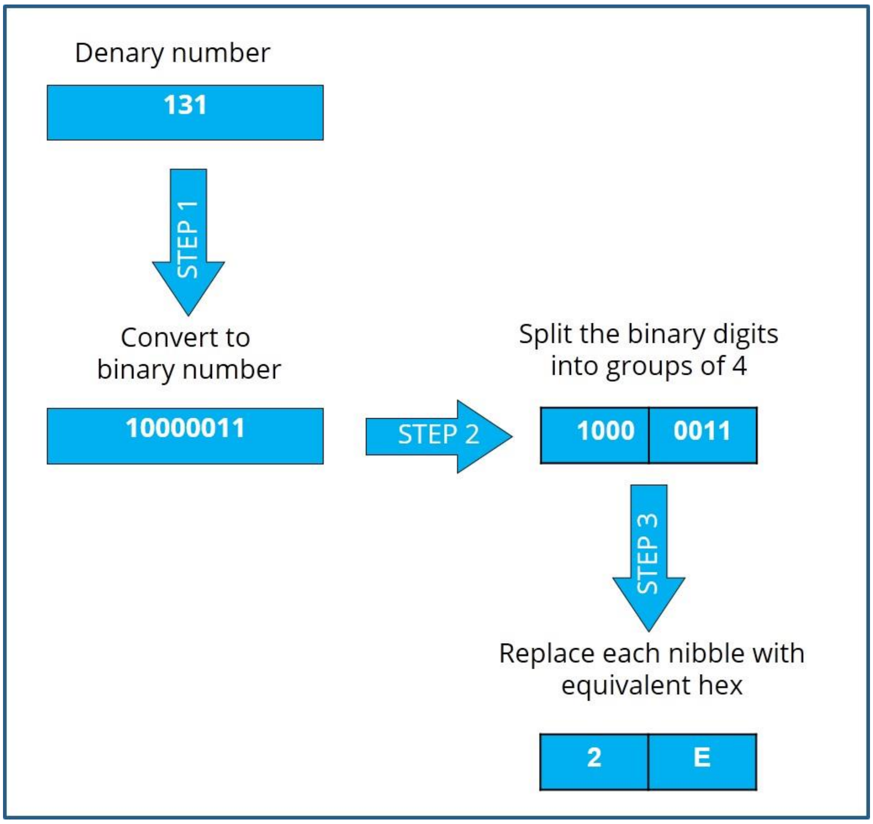

Figure 3: Converting denary numbers to hexadecimal

Converting hexadecimal to denary

Hexadecimal numbers can be converted to the denary system by using the place values. Let us consider converting hexadecimal number 7AF into the denary system. The place value is in the powers of 16. The value of each digit is multiplied with the place value and the values are added to find the denary equivalent.

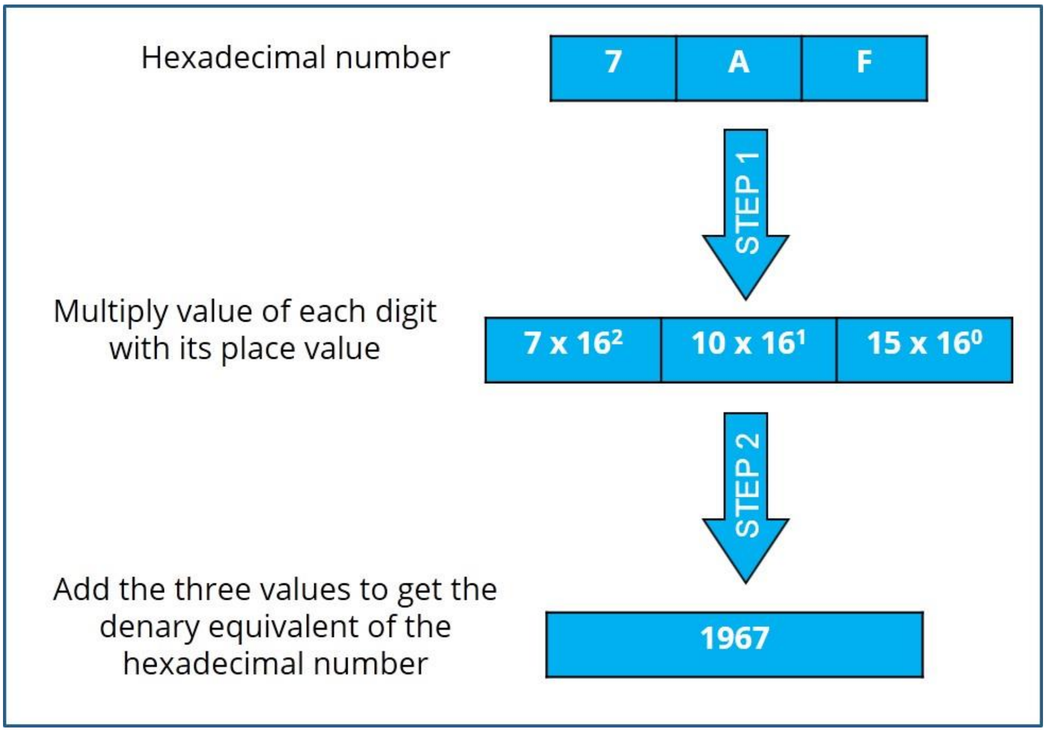

Figure 4: Converting denary numbers to hexadecimal

Applications of the hexadecimal system

The hexadecimal system is a convenient form to represent information compared to binary. A nibble is represented in one digit of the hex system. A few applications of the hex system are explained below:

a) Colours

Hexadecimal numbers are used to represent colours in various software tools. A colour is represented in #RRGGBB format. RR, GG and BB represent the hex number of red, green and blue colours. A pure blue colour is #0000FF, white is #FFFFFFF and black is #0000000. Using this method, 256 variants of each colour can be made, making a total of 256 reds ×256 greens ×256 blues. A shade of yell

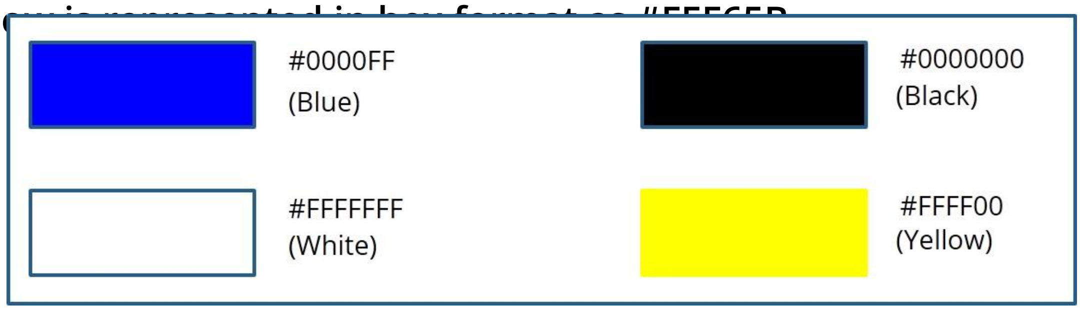

Figure 5: Hex number of colours

b) RGB Colour Model

Colours can also be represented using the RGB colour model. This system is similar to the hex system but each colour has a value between 0 to 255. Hence, a shade of pink represented in hex format as # 54, G value of 246 and B value of 91.

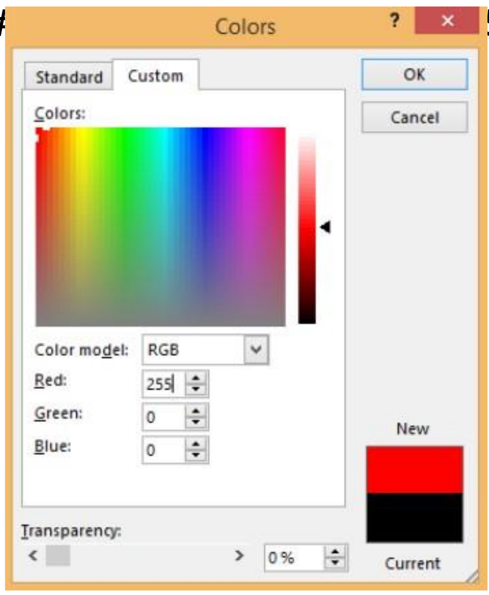

Figure 6: RGB colour model

c) Errors

Error messages contain a hex number that represents the memory location of the error. This information is very useful to programmers. The error can be easily rectified by verifying the code in that particular location. The contents of part of a memory helps the programmer to solve many problems. The process in which the contents of memory is displayed or stored in a storage drive in case of a system crash is called a memory dump. An example of a memory dump is shown in the figure below.

Location	Contents					
A0B8F10	89	20	A0	B1	C2	F3
A0B8F11	1A	3D	F4	56	F7	9A
A0B8F12	AB	C2	D1	9D	4F	5D

Using this information, a programmer can identify the exact location where the error lies. Compared to binary, this is easier to understand. The binary form of DEF6B2C3 is: 11011110111101101011001011000011.

Even though using the hexadecimal system improves readability of errors, the programmer must have knowledge about the computer architecture to interpret the results.

d) MAC addresses

A media access control address refers to the number that uniquely identifies a device on the Internet. This is the address of the network interface card (NIC). A MAC address is made up of 48 bits, which are shown as a six-group of hexadecimal digits.

NN-NN-NN-DD-DD-DD or NN: NN: NN: DD: DD: DD

The first half of the address represents the identification number of the manufacturer and the second half of the number represents the serial number of the device. For example: 00: 14: 22: 34: AC: 4F refers to a device made by Dell with a serial number of 34AC4F.

There are two types of MAC address: Universally Administered MAC Address (UAA) and Locally Administered MAC Address (LAA).

The UAA is set by manufacturer and is most commonly used. This address is not changed most of the times. In case the UAA of a device is changed, it is important to make sure that the UAA is unique.

A user or organisation may change their UAA due to:

- Some software applications used on mainframe systems requires the systems to use a MAC address that follows a strict format. To ensure that all devices have MAC address that obey this format, it may be changed.
- A router or firewall may allow MAC addresses with certain format only. In order to bypass them, MAC address of device may be changed.
- Some networks may restrict certain MAC addresses. In order to use the network, certain devices may have their MAC addresses changed.

e) URL

Hexadecimal values are used to represent web addresses or URL (Uniform Resource Locator). The ASCII codes are used to represent the web address. For example:

www.google.com becomes: (using the ASCII codes)

w	w	w	.	g	o	o	g	l	e	.	c	o	m
%77	%77	%77	%2E	%67	%6F	%6F	%67	%6C	%65	%2E	%63	%6F	%6D

% denotes that hexadecimal values are used. Hexadecimal values are used to represent addresses of files and web pages to improve security. A user is protected from accessing a fake website when he is asked to use a hexadecimal URL rather than a URL with letters that could be misleading.

f) Assembly code and machine code

Machine code and assembly code can be used to refer the computer memory directly. Use of hexadecimal numbers makes low-level language coding easier, faster and reduces errors compared to the binary system. For example: An assembly code instruction ADD is translated to machine code as 0100, which is equivalent to 4 in hexadecimal. It is enough for the programmer to type a single digit '4' compared to four digits '0100'.

g) Hypertext Markup language (HTML)

HTML is a markup language, widely used in developing web pages. It is used to define the attributes of text such as colour. Tags are used to define objects in a web-page. For example: The tag <p> is used to define a paragraph. The tag </p> represents its end. The contents between the two are codes.

Earlier in this section, we have learnt how hexadecimal codes are used to represent colours. These codes are also used in html to define objects.

For example:

```
<p style="color:#FF8000;">Welcome to my website.</p>
```

The colour code FF8000 represents orange colour. The paragraph looks like this:

Welcome to my website.

Error-checking methods

Data is transmitted through a channel, could get corrupted or changed. Hence, it's very important to implement error-checking methods in computer technologies. These methods are used to detect and correct errors.

Parity checking

Parity checking uses a parity bit in each byte of data. This bit is allocated before transmission. An even parity has an even number of bit 1s in a byte. An odd parity has an odd number of bit 1s in a byte. Systems may use even or odd parity. The sender and receiver make an agreement prior to the transmission about the type of parity.

<table>
<tr><td>Even parity</td><td>1</td><td rowspan="2">1</td><td rowspan="2">0</td><td rowspan="2">1</td><td rowspan="2">0</td><td rowspan="2">1</td><td rowspan="2">1</td><td rowspan="2">1</td></tr>
<tr><td>Odd parity</td><td>0</td></tr>
<tr><td></td><td>Parity bit</td><td colspan="7">Data</td></tr>
</table>

In the above data, there are 5 bits of 1. Hence, to make the number of 1 bits even, a parity bit is assigned bit 1. The number of bits is odd and, hence, the parity bit in odd parity is 0.

Let us consider the following byte of data, which uses even parity, received by a receiver.

<table>
<tr><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td><td>1</td><td>1</td><td>0</td></tr>
<tr><td>Parity bit</td><td colspan="7">Data</td></tr>
</table>

The number of bit 1s is 5 and this indicates transmission error. Some bits have changed during transmission. This is explained further in the article "Data transmission technologies".

Check digits

Check digits is another method that is used to detect errors. In this method, an additional digit is included for every seven digits. This bit is calculated using a standard algorithm and is called a checksum digit. In barcodes, this checksum digit is calculated using the following steps.

a) Starting from the left side of the number, multiply each digit alternatively by 3 and 1.

b) Find the sum of the values obtained from step a. Let us name it "x".

c) Round the number x to the nearest 10. Let us name this "y".

d) Subtract x from y to find the check digit.

e) Include the digit obtained in step c to the right-most side of the number in the barcode. This number is called EAN 8. EAN is the European Article Numbers and is used for finding the checksum for various products.

Let us find the EAN 8 form of the number 7325931.

Multiplying digits by 3 and 1	7×3=21	3×1=3	2×3=6	5×1=5	9×3=27	3×1=3	1×3=3	
Sum of the numbers	x= 21 + 3 + 6 + 5 + 27 + 3 + 3 = 68							
Round x to nearest 10	y = 70							
Finding check digit	y – x = 70 – 68 = 2							
EAN 8 number	7	3	2	5	9	3	1	2