

Data representation is primarily seen in 4 different forms.

- 1 Binary - A number system with a base of 2.
- 2 Denary - A number system with a base of 10.
- 3 Octal - A number system with a base of 8.
- 4 Hexadecimal - A number system with a base of 16.

Numbers in general are referred to in the denary number system.

1,234,724 is also written as $(1234724)_{10}$

To the base of any number, means that each digit increases in value by an ~~increas~~ increasing power ^{of the base}. Let's visualise this.

$$(121)_{10} \text{ is nothing but } \dots$$

$$\left[\begin{array}{l} \rightarrow 1 * 10^0 = 1 \\ \rightarrow 2 * 10^1 = 20 \\ \rightarrow 1 * 10^2 = 100 \end{array} \right] = 121$$

Given this, it also explains that no digit can contain the value of the base itself. Even base 10 ranges from 0-9, where base 2 ranges from 0-1 and hexadecimal ranges from 0-15.

This is why the binary numbers system, the machine readable language, is in 0s and 1s.

Hexadecimals contain the values 0-9 as 0-9 and the values 10-15 as A-F. In other words;

$$A \rightarrow 10$$

$$B \rightarrow 11$$

$$C \rightarrow 12$$

$$D \rightarrow 13$$

$$E \rightarrow 14$$

$$F \rightarrow 15$$

This might result in us getting numbers in the form of $(FA3)_{16}$ which look much more difficult than they seem.

Taking the above example, consider each value as a digit.

$$(FA3)_{16}$$

$$\left. \begin{array}{l} \rightarrow 3 \times 16^0 = 3 \\ \rightarrow 10 \times 16^1 = 160 \\ \rightarrow 15 \times 16^2 = 3840 \end{array} \right\} = (4003)_{10}$$

It is the exact same with binary. Lets take $(1001100101)_2$ as the example this time.

$$(1001100101)_2$$

$$\left. \begin{array}{l} \rightarrow 1 \times 2^0 = 1 \\ \rightarrow 1 \times 2^1 = 2 \\ \rightarrow 1 \times 2^2 = 4 \\ \rightarrow 1 \times 2^5 = 32 \\ \rightarrow 1 \times 2^6 = 64 \\ \rightarrow 1 \times 2^9 = 512 \end{array} \right\} = 613$$

We need to be able to ~~to~~ convert from 1 number system to another.

Binary $\leftrightarrow$ Denary

Binary $\leftrightarrow$ Hexadecimal

Denary $\leftrightarrow$ Hexadecimal

On the previous page we discussed how to convert to decimal, let's now discuss how to convert from decimal to other forms

Decimal $\longrightarrow$ Binary

$$142 \div 2 = 71 \text{ R } 0$$

$$71 \div 2 = 35 \text{ R } 1$$

$$35 \div 2 = 17 \text{ R } 1$$

$$17 \div 2 = 8 \text{ R } 1$$

$$8 \div 2 = 4 \text{ R } 0$$

$$4 \div 2 = 2 \text{ R } 0$$

$$2 \div 2 = 1 \text{ R } 0$$

$$1 \div 2 = 0 \text{ R } 1$$

$(10001110)_2$

read bottom up

Decimal $\longrightarrow$ Hexadecimal

$$479 \div 16 = 29 \text{ R } 15$$

$$29 \div 16 = 1 \text{ R } 13$$

$$1 \div 16 = 0 \text{ R } 1$$

1 13 15

$(1 \text{ D } \text{F})_{16}$

Hexadecimal $\rightarrow$ Binary

A hexadecimal can hold about 16 different number values. This is the same as 4 digits of binary as they too can hold 16 different numbers. Using this logic, we can convert Hexa decimal to binary.

$$(FA3)_{16} \rightarrow \begin{array}{ccc} 15 & 10 & 3 \\ \downarrow & \downarrow & \\ 1111 & 1010 & 0011 \end{array}$$

$$\rightarrow (FA3)_{16} = (111110100011)_2$$

Binary $\rightarrow$ Hexadecimal

Using the same logic above...

$$(111001011000101)_2$$

111	001	011	000	101	
111	001	011	000	101	
111	001	011	000	101	
111	001	011	000	101	

111

001

011

000

101

$\rightarrow$ 5
 $\rightarrow$ 12
 $\rightarrow$ 2
 $\rightarrow$ 7

~~72C5~~
(72C5)₁₆