3.3.2 Converting between number bases

Teacher notes

Converting between number bases

A computer uses volatile memory to hold data before processing – Random Access Memory (RAM). You may be familiar with the memory modules that constitute RAM by for example having fitted them to a computer at home in order to enhance its performance.

Image of typical RAM memory module

There are other parts of the computer that will also make use of temporary data storage areas eg the Central Processing Unit (CPU).

Key point

It is important to remember, however, that whatever the physical means of storing data, the **data** is **ordered** in a **logical fashion** on the computer.

Why can a 32-bit processor only access 4GB RAM?
Every byte of RAM requires its own address and the processor limits the length of those addresses. A 32-bit processor uses addresses that are 32 bits long. There are only 4,294,967,296 (or 4GB) possible 32-bit addresses.

Students will look at a simple example of storing data using one byte of storage.

A byte is a collection of eight bits and may be represented as follows:

Value as power of 2	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Decimal value	128	64	32	16	8	4	2	1
Sample 8-bit pattern	0	1	0	1	1	0	0	1
Individual values	0	64	0	16	8	0	0	1
Total decimal value	64 + 16 + 8 + 1 = 89							

Key point

When one byte of storage is used the maximum number in base10 that can be stored is 255 (11111111 in binary) and the minimum is 0.

Converting decimal to binary

Example: Convert the number 115_{10} into 8-bit binary.

Reading from left to right, look at each column in turn.

	128	64	32	16	8	4	2	1
Step 1	0	115 – **128** = negative, write 0 in that column.						
Step 2	0	1	115 – **64** = 51, write 1 in that column.					
Step 3	0	1	1	51 – **32** = 19, write 1 in that column.				
Step 4	0	1	1	1	19 – **16** = 3, write 1 in that column.			
Step 5	0	1	1	1	0	3 – **8** = negative, write 0 in that column.		
Step 6	0	1	1	1	0	0	3 – **4** = negative, write 0.	
Step 7	0	1	1	1	0	0	1	3 - **2** = 1, write 1.
Final answer	**0**	**1**	**1**	**1**	**0**	**0**	**1**	**1***

*Remainder 1 goes in final column

Converting binary to decimal

For the reverse process, look at the binary pattern (splitting it into four-bit sections usually helps readability and is very useful for hexadecimal later) and work from right to left identifying whether each successive bit is ‘on’ or ‘off’ ie is 1 or 0 in value. Then add up the totals where the bit is set.

Example: Convert the number 00101100_2 to decimal

0	**0**	**1**	**0**	**1**	**1**	**0**	**0**
128	64	32	16	8	4	2	1
Decimal total value					32 + 8 + 4 = 44		

Making binary easier to represent

Using the hexadecimal number system

Binary numbers very quickly become very long so it can be beneficial to convert from binary to hexadecimal.

Hexadecimal is a base 16 number representation system. Decimal uses the digits 0-9, whereas hexadecimal uses 0-9 with the addition of A-F. Note that capital letters are always used.

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

Key point

Hexadecimal in itself means nothing to a CPU. However, it is useful to humans as they can abbreviate binary notation.

Converting from binary to hexadecimal

Conversion involves looking at four bits at a time and grouping them as a single hexadecimal ('hex') character. Four bits are used as 1111 in binary, is equivalent to 15 in decimal, which in turn is equivalent to F in hexadecimal.

Example 1	Value			
Binary	1	1	0	0
Decimal	8	4	2	1
Sum(base 10)	12			
Hex	C			

Notes

The binary value 1100_2 converts to a total of 12_{10} which is equivalent to the hexadecimal value of C_{16}.

Example 2	Value			
Binary	0	1	1	0
Decimal	8	4	2	1
Sum(base 10)	6			
Hex	6			

Notes

The binary value 0110_2 converts to a total of 6_{10} which is equivalent to the hexadecimal value of 6_{16}.

Example 2	Value							
Binary	1	1	0	1	0	0	0	1
Decimal	8	4	2	1	8	4	2	1
Sum(base 10)	13				1			
Hex	D				1			

Notes

In this case, the binary value has been split into two groups of four. In each group the highest power of 2 is 2^3.

The first group of four converts to D in hex and the second group converts to 1. Combining these two values gives $D1_{16}$.

Hexadecimal notation used in web pages

Hexadecimal notation is used to define the appearance of text in web pages.

Typically, '#FFFFFF' represents the colour white in the RGB (red, green, blue) notation used in web pages.

All computer input and output devices use the RGB colour model.

In decimal, FFFFFF = 16^6 = 16,777,216. This equates to '24-bit' colour depth where each pixel is represented by three coloured components using 'RGB'.

If each of these components has 256 variations then …

- To represent all combinations you need 256 x 256 x 256 = 16,777,216
- 256 in binary is represented by eights bits and 3 x 8 = 24 bits.

Before the lesson, you should familiarise yourself with Hex colours! (Extension task) as it requires students to have access to computers with the index.html and style.css files on.

Instructions for using these files are contained in the **Lesson plan and printable activities**.