

# Data Representation

Keyword	Definition
<b>Bit</b>	A single 0 or 1
<b>Byte</b>	8 bits
<b>Decimal</b>	Base 10 number system
<b>Binary</b>	Base 2 number system
<b>Hexadecimal</b>	Base 16 number system (Hex)
<b>ASCII &amp; Unicode</b>	Character sets that can be represented by a computer
<b>Metadata</b>	Data about data
<b>Lossy compression</b>	Permanently loses some data when compressed
<b>Lossless compression</b>	Will not lose any of the original data when compressed

### Bits and Bytes

- 1 bit = a single 0 or 1 (**b** represents bit)
- 1 byte = 8 bits = 1 character of text (**B** represents byte)
- (A nibble = 4 bits or half a byte)
- 1 kB (kilobyte) = 1000 bytes
- 1 MB (megabyte) = 1000 kB or 1000x1000 bytes
- 1 GB (gigabyte) = 1000 MB
- 1 TB (terabyte) = 1000 GB

### Hexadecimal

Decimal	Hex	Decimal	Hex
1	1	9	9
2	2	10	A
3	3	11	B
4	4	12	C
5	5	13	D
6	6	14	E
7	7	15	F
8	8	16	10

### Binary representation

Number of Bits	Possible states
1	2
2	4
3	8
4	16
5	32
6	64
7	128
8	256

### Most and least significant bit

- The most significant bit (MSB) is the bit with the largest value
  - This is the bit that is furthest to the left
- The least significant bit (LSB) is the bit with the smallest value
  - This is the bit that is furthest to the right

128 64 32 16 8 4 2 1

**1** 0 0 1 1 1 0 **1**

### Decimal to Binary

- How do you convert 28 to binary?
- Method
  - Working right to left, write out the numbers 1, 2, 4, 8 and so on, doubling each time to 128

128 64 32 16 8 4 2 1

**0 0 0 1 1 1 0 0**

- 128, 64 and 32 are all greater than 28, so put a zero for these
- Put a 1 in the 16 column, 28-16=12
- Put a 1 in the 8 column, 12-8 = 4
- Put a 1 in the 4 column, 4-4=0 so put zero in other columns

### A sense of scale

File	Size
One character of text	1 byte
A full page of text	30 kB
One small digital colour photograph	3 MB
Music CD capacity	650 MB
DVD capacity	4.5 GB
Hard disk capacity	1 TB

### Hex to Decimal conversion

16s      Units

**2**      **A**

2x16 + 10 = 42 in decimal

Multiply the left-hand digit by 16, then add the units

# Data Representation

## Decimal to Hex conversion

1 Divide the denary number by 16 and record the remainder.

$$164 \div 16 = 10 \text{ remainder } 4$$

2 The remainder, 4, is the number of 1s, and the answer, 10, is the number of 16s.

10 in hex is A

16	1
A	4

164 is equivalent to A4.

You can also convert denary to hex by converting denary to binary digits and then converting binary digits to hex.

## Binary to Hex conversion

1 Split 01001100 into nibbles, and place each nibble into a base 2 place value table.

0100				1100			
8	4	2	1	8	4	2	1
0	1	0	0	1	1	0	0

2 For each nibble, add the column headings for the columns that contain a 1.

$$4 = 8 + 4 = 12$$

3 Convert the denary values to hex.

$$4 = 4 \quad 12 = C$$

01001100 is equivalent to 4C.

## Hex to Binary conversion

1 Convert each hex digit to binary.

A				6			
8	4	2	1	8	4	2	1
1	0	1	0	0	1	1	0

$$A = 10 = 8 + 2 \quad 6 = 4 + 2$$

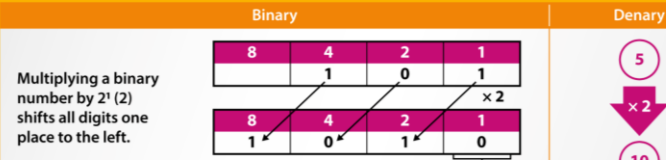
2 Combine the nibbles to make the binary number.

10100110

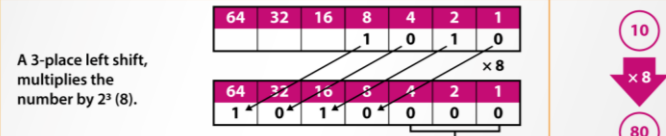
A6 is equivalent to 10100110.

## Binary Shifts - Multiplication

When a binary number is multiplied by a power of 2 there is a 'left shift' of bits.



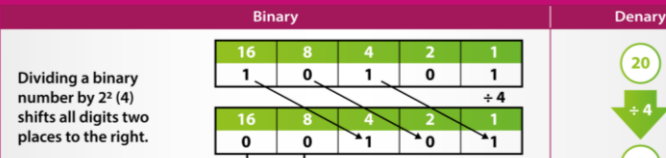
When multiplying, fill each empty place to the right with a 0.



When multiplying, fill each empty place to the right with a 0.

## Binary Shifts - Division

When a binary number is divided by a power of 2 there is a 'right shift' of bits.



When dividing, fill each empty place to the left with a 0.

## Calculating image size

$$\text{Size in bits} = \text{Width} \times \text{Height} \times \text{Colour depth}$$

$$\text{Size in bytes} = \text{Width} \times \text{Height} \times \text{Colour depth} / 8$$

## Calculating sound file size

$$\text{Size in bits} = \text{Sampling Rate} \times \text{Resolution} \times \text{Duration}$$

$$\text{Size in bytes} = \text{Sampling Rate} \times \text{Resolution} \times \text{Duration} / 8$$

## Data Compression

### Lossy Compression

In lossy compression data is permanently removed from a file to reduce its size.

- Reduces file size significantly
- Reduces the quality of original file
- Not suitable for text files or computer programs that must retain all original data
- Removed data cannot be restored



Original Image



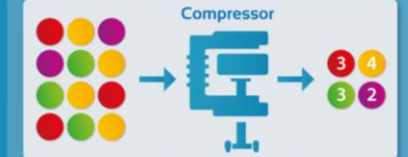
Reduced Quality Image

Example file types: JPEG, MP3, MPEG-4.

### Lossless Compression

In lossless compression no data is removed. Redundant and duplicate data is repurposed to reduce file size.

- Less file-size reduction than with lossy compression
- Decompresses back to original quality
- Can be used on computer programs and text files



Example file types: PNG, TIFF, ZIP.

## RLE – Run Length Encoding

Assuming 1 = white and black = 0, this run could be encoded as:

1111000000000000



Using RLE, the row could be represented as 4 '1's and 12 '0's or 4 1 12 0

## Huffman compression

• Huffman coding of "she sells sea shells" = 49 bits

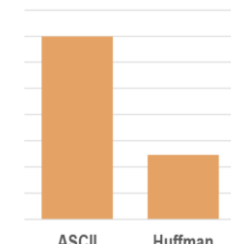
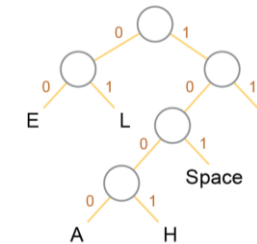
• 7-bit ASCII coding = 20 Characters or 140 bits

• This is a lossless reduction in the file size of 65%

## Huffman Trees

Consider the frequency of each character in the sentence: "SHE SELLS SEA SHELLS"

Character	Frequency
S	6
E	4
L	4
Space	3
H	2
A	1



## Binary Addition

Work right to left and apply these simple rules:

1. 0 + 0 = 0
2. 0 + 1 = 1
3. 1 + 0 = 1
4. 1 + 1 = 0 Carry 1
5. 1 + 1 + 1 = 1 Carry 1

