CHAPTER 1: INFORMATION REPRESENTATION

1.1 DATA REPRESENTATION

1.1.1 Fundamental Characteristics of Number Systems

Every number system has two fundamental characteristics:

1. **Base (Radix)**: The number of different digits that a system can use to represent numbers
2. **Place Value**: The specific value of a digit based on its position within a number

1.1.2 Denary (Decimal) System - Base 10

- Uses digits 0-9
- Each position represents powers of 10 (10?, 10^1, 10^2, etc.)
- Example: 3,567 = (3×10^3) + (5×10^2) + (6×10^1) + (7 × 10?)

1.1.3 Binary System - Base 2

Key Points:

- Uses only two digits: 0 and 1
- Each bit (binary digit) represents a power of 2
- All data and characters in computers are represented in binary

Binary Place Values:

<TEXT>

```
128 | 64 | 32 | 16 | 8 | 4 | 2 | 1
 2? 2? 2? 2? 2³ 2² 2¹ 2?
```

Example - Converting Denary to Binary:

- Denary 65 in binary: 01000001
- Calculation: 64 + 1 = 65

Example - Converting Binary to Denary:

- Binary 01000001 = 64 + 1 = 65

1.1.4 Binary Prefixes vs Decimal Prefixes

It is crucial to understand the difference between binary prefixes (based on powers of 2) and decimal prefixes (based on powers of 10):

Denary Prefix	Factor	Value	Binary Prefix	Factor	Value
kilo- (k)	$\times 10^{3}$	1,000	kibi- (Ki)	$\times 2^{1}$?	1,024
mega- (M)	×10?	1,000,000	mebi- (Mi)	$\times 2^{2}$?	1,048,576
giga- (G)	×10?	1,000,000,000	gibi- (Gi)	$\times 2^{3}$?	1,073,741,824
tera- (T)	$\times 10^{12}$	1,000,000,000,000	tebi- (Ti)	×2??	1,099,511,627,776

Important: Always use the correct prefix:

- Computer storage uses binary prefixes (KiB, MiB, GiB, TiB)
- Data transfer rates often use decimal prefixes (kbps, Mbps, Gbps)

1.1.5 Binary Coded Decimal (BCD)

Definition: Binary representation where each individual denary digit is represented by a sequence of 4 bits (nibble).

Characteristics:

- Each nibble can represent denary digits 0-9
- Uses only specific 4-bit patterns (0000 to 1001)
- The patterns 1010 to 1111 are not used in BCD

Example - Converting 429 to BCD:

<TEXT>

```
4 = 0100
2 = 0010
9 = 1001
Therefore, 429 in BCD = 0100 0010 1001
```

Practical Applications:

- Electronic devices displaying numbers (calculators)
- Accurately measuring decimal fractions

- Electronically coding denary numbers

1.1.6 Two's Complement Representation

Two's complement is used to represent negative numbers in binary.

Converting Negative Denary to Binary (Example: -42):

Step 1: Find binary equivalent (ignoring sign)

<TEXT>

```
42 = 00101010 (8-bit representation)
```

Step 2: Convert to one's complement (flip all bits)

<TEXT>

```
00101010 ? 11010101
```

Step 3: Add 1 to get two's complement

<TEXT>

```
11010101 + 1 = 11010110
```

Converting Binary Two's Complement to Denary (Example: 11010110):

Step 1: Flip all bits

<TEXT>

```
11010110 ? 00101001
```

Step 2: Add 1

<TEXT>

```
00101001 + 1 = 00101010
```

Step 3: Convert to denary and apply negative sign

<TEXT>

```
00101010 = 42
Therefore: -42
```

Range in 8-bit Two's Complement:

- Maximum positive: +127 (01111111)
- Maximum negative: -128 (10000000)

Overflow:

- Occurs when the result of an arithmetic operation is too large/small to fit in the allocated bits
- Example: Adding 127 + 1 in 8-bit gives -128 (overflow)

1.1.7 Hexadecimal System - Base 16

Characteristics:

- Uses digits 0-9 and letters A-F
- A=10, B=11, C=12, D=13, E=14, F=15

Converting Denary to Hexadecimal: Example: 165 to Hex

<TEXT>

```
165 ÷ 16 = 10 remainder 5
10 = A
Therefore: 165 = A5 (hex)
```

Converting Hexadecimal to Denary: Example: A5 to Denary

<TEXT>

```
A5 = (10 × 16) + (5 × 1) = 160 + 5 = 165
```

Practical Applications:

- Defining colours in HTML (#FF0000 = red)
- Defining MAC addresses
- Assembly languages and machine code
- Debugging via memory dumps

1.1.8 Character Sets and Encoding

Definition: A character set is a collection of characters that can be represented using binary codes. It typically includes upper and lower case letters, number digits, punctuation marks, and other characters.

Character Encoding Standards:

Standard	Description	Bits per Character	Characters
ASCII	American Standard Code for Information Interchange	7 bits	128
Extended ASCII	Extension of ASCII	8 bits	256
Unicode	Superset of ASCII and extended ASCII	16 or 32 bits	65,536+

ASCII:

- Only supports English alphabet
- 7 bits = 128 possible characters
- Includes control characters (0-31), printable characters (32-126)

Extended ASCII:

- 8 bits = 256 possible characters
- Includes most European languages' alphabets
- Still limited for global languages

Unicode:

- Modern international standard
- Supports all global languages
- UTF-8 uses 1-4 bytes per character
- Backward compatible with ASCII

1.2 MULTIMEDIA - GRAPHICS AND SOUND

1.2.1 Bitmap Images

Definition: Bitmap images are created by assigning a solid colour to each pixel using bit patterns. The image is represented as a grid of pixels, where each pixel's colour is encoded using binary values.

Key Terms:

- **Pixel:** The smallest picture element whose colour can be accurately represented by binary code
- **File Header:** Contains metadata including image size, number of colours, etc.

Image Resolution:

- Definition: The number of pixels that make up an image
- Example: 4096 × 3192 pixels

- Effect: Higher resolution results in sharper, more detailed images

Screen Resolution:

- Definition: The number of pixels that can be viewed horizontally and vertically on a device's screen
- Example: 1680 × 1080 pixels

Colour Depth:

- Definition: The number of bits used to represent the colour of a single pixel
- Formula: If n bits are used, there are 2? colours per pixel
- Example: 16-colour bitmap = 4 bits per pixel (2? = 16)
- Effect: Increasing colour depth improves colour quality but increases file size

File Size Calculation:

<TEXT>

```
File Size = Number of Pixels × Colour Depth
```

Example Calculation:

<TEXT>

```
Image: 1024 × 768 pixels, 24-bit colour
Number of Pixels = 1024 × 768 = 786,432
Colour Depth = 24 bits
File Size = 786,432 × 24 = 18,874,368 bits
 = 18,874,368 ÷ 8 = 2,359,296 bytes
 ? 2.36 MB
```

Applications:

- Scanned images
- Digital photographs
- Computer screen displays
- Small file sizes and easy manipulation when needed

1.2.2 Vector Graphics

Definition: Made up of drawing objects (mathematically defined constructs like rectangles, lines, circles, curves).

Components:

- **Drawing List:** A set of commands defining the vector
- **Properties:** Basic geometric data determining shape and appearance
- **Encoding:** Data is encoded using mathematical formulas

Advantages over Bitmap:

- Objects can be resized without losing quality
- Scalability is the key benefit
- Smaller file sizes for simple images
- Can be enlarged infinitely without pixelation

Disadvantages:

- Cannot represent complex images like photographs
- More complex to create

Applications:

- Company logos
- Architectural drawings
- Icons and symbols
- Fonts (TrueType, PostScript)

1.2.3 Sound Representation

Analogue vs Digital:

Analogue	Digital
Continuous electrical signals	Discrete electrical signals
Infinite detail	Finite representation
Cannot be stored directly	Can be stored in binary

Sound as Analogue Data:

- Sound consists of vibrations through a medium
- Inherently analogue due to infinite detail variation

Conversion Process (Analogue to Digital):

1. **Sampling:** The sound wave's amplitude is measured at set time intervals
2. **Quantization:** Each sample is assigned a binary value
3. **Encoding:** Binary values are stored

Key Terms:

- **Sampling Rate:** Number of samples taken per unit of time (measured in Hz)
    - Effect: Increasing sampling rate improves accuracy but increases file size

- CD quality: 44,100 Hz

- **Sampling Resolution:** Number of bits used to encode each sample
  - Effect: Increasing resolution improves accuracy but increases file size
  - CD quality: 16 bits

- **Bit Rate:** Number of bits used to store 1 second of sound
  - Formula: Bit Rate = Sampling Rate × Sampling Resolution
  - Example: 44,100 × 16 = 705,600 bps (approximately 706 Kbps)

1.3 COMPRESSION

1.3.1 Need for Compression

Definition: Compression is the process of reducing file size without significant loss in quality.

Benefits:

- Reduced storage requirements
- Faster data transfer (uses less bandwidth)
- Reduced time needed to search for data

1.3.2 Lossless Compression

Definition: A type of compression that allows original data to be perfectly reconstructed from the compressed file.

Key Feature:

- Uses some form of replacement (substitution)
- No data is permanently deleted

Examples:

- PNG images (for graphics with sharp edges)
- ZIP files
- Text file compression
- Database records
- Run-Length Encoding (RLE)

Run-Length Encoding (RLE):

Definition: A form of lossless compression used for compressing text files and bitmap images.

Mechanism:

- Reduces file size by encoding sequences of adjacent, identical elements
- Encodes as two values: run count and run value

Example: Original: AAAAAAABBBBBCCCCCC Compressed: 7A5B6C

Example - Bitmap: Original row: White White White White White Black Black Compressed: 5W2B

Applications:

- Simple graphics with large areas of same colour
- Database records with repeated values

1.3.3 Lossy Compression

Definition: A type of compression that irreversibly eliminates unnecessary data.

Characteristics:

- File accuracy/quality is lower than lossless
- File size is significantly reduced (often to about 10% of lossless size)
- Some original data is permanently lost

Examples:

- MP3 (sound files)
- JPEG (images)
- MP4 (video files)

Mechanism in Sound Files (MP3):

- **Perceptual Coding:** Removes parts of the sound that are less audible or discernible to human hearing
- Removes frequencies outside human hearing range
- Removes subtle volume differences

Mechanism in Images (JPEG):

- Removes high-frequency details
- Uses mathematical approximations
- Reduces colour precision in less important areas

When to Use Lossy vs Lossless:

Lossless	Lossy
Text documents	Photography
Database files	Video streaming

Lossless	Lossy
Program files	Music (streaming)
Spreadsheets	Web graphics (where size matters)

Revision #2
Created 2026-03-16 06:50:51 UTC by Samuel Lee
Updated 2026-03-17 01:14:04 UTC by Samuel Lee