Bits, bytes and digital information

Lecture 2 - COMPSCI111/111G
Today’s lecture

- Understand the difference between analogue and digital information
- Convert between decimal numbers and binary numbers
Analogue vs digital information

- Information in the real world is continuous
  - Continuous signal

- Information stored by a computer is digital
  - Represented by discrete numbers
Encoding information

- Real world information is stored by a computer using numbers

- Visual information

1. Give each pixel colour a number.
2. Let the computer draw the numbers as coloured pixels (eg. black = 0).
Encoding information

Sound information

1. Give each sample a number (height of green box).
2. Let the computer move the loudspeaker membrane according to the samples.
Numbers and Computing

- Numbers are used to represent all information manipulated by a computer.

- Computers use the binary number system:
  - Binary values are either 0 or 1.

- We use the decimal number system:
  - 0 to 9 are decimal values.
Number Systems

- **Base:**
  - Specifies the number of digits used by the system.
  - Binary is base 2.
  - Decimal is base 10.

- **Positional notation:**
  - Describes how numbers are written.

\[ d_n d_{n-1} \cdots d_1 \]

Most significant digit  Least significant digit
Positional Notation

- Any number can be expressed as:

$$d_n \times b^{n-1} + d_{n-1} \times b^{n-2} + \cdots + d_1 \times b^0$$

where $d_i$ is the digit at position $i$, and $b$ is the base.
Decimal Examples

- 657

\[ 6 \times 10^2 + 5 \times 10^1 + 7 \times 10^0 \]
\[ 600 + 50 + 7 = 657 \]

- 9308

\[ 9 \times 10^3 + 3 \times 10^2 + 0 \times 10^1 + 8 \times 10^0 \]
\[ 9000 + 300 + 0 + 8 = 9308 \]
Storing Decimal Numbers in a Computer

- **Series of dials:**
  - Each dial goes from 0 to 9.

- **Information is stored digitally:**
  - Finite number of states - 10 per dial.
  - No in-between states.

- **Decimal number system:**
  - 1\textsuperscript{st} dial from right: \(10^0\)
  - 2\textsuperscript{nd} dial from right: \(10^1\)
  - 3\textsuperscript{rd} dial from right: \(10^2\)
  - etc.

\[
6 \times 10^2 + 3 \times 10^1 + 8 \times 10^0 = 638
\]
Exercises

The following two questions relate to dials that have 10 different states, as discussed in the previous slide.

- Given a machine that uses 4 dials, how many different numbers can we represent?

- If we want to represent 256 different values, how many dials do we need?
Switches

- A dial is complicated.
  - Each dial has 10 different states (0 - 9).
  - Physically creating circuits that distinguish all states is complicated.
  - Would need to distinguish 10 different strengths of electricity (voltages).

- Switches are simple.
  - Each switch is off or on (0 or 1).
  - Physically creating the circuits is easy.
  - Switch off: electrical current cannot flow.
  - Switch on: electrical current can flow.
Bits and Bytes

- Each binary number is known as a **Binary digit**, or bit.

- A bit can be either a 0 or a 1.

- Bits are used in groups.

- A group of eight bits is referred to as a **byte**.
Using Binary Numbers

How many different values/states can we have with:

1 bit: 2 bits: 3 bits:

0          00          000
0          01          001
1          10          010
1          11          011

100        101        110
111
Exercises

- How many different values can we represent with a byte?

- If we want to represent 30 different values, how many bits would we need?
Converting binary to decimal

110

\[ 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 \]
\[ 4 + 2 + 0 = 6 \]

10110

\[ 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 \]
\[ 16 + 0 + 4 + 2 + 0 = 22 \]
Converting from decimal to binary

35 is 100011 in binary

106 is 1101010 in binary
Exercises

- What is the decimal equivalent of 101111?

- What is the binary equivalent of 123?
Prefixes

- A group of 8 bits is a byte
  - A group of 4 bits is a nibble

- Bytes are the common unit of measurement for memory capacity

- There are two sets of prefixes:
  - Decimal
  - Binary
# Decimal prefixes

<table>
<thead>
<tr>
<th>$10^n$</th>
<th>Prefix</th>
<th>Symbol</th>
<th>Decimal</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>none</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>$10^3$</td>
<td>kilo</td>
<td>K</td>
<td>1000</td>
</tr>
<tr>
<td>$10^6$</td>
<td>mega</td>
<td>M</td>
<td>1,000,000</td>
</tr>
<tr>
<td>$10^9$</td>
<td>giga</td>
<td>G</td>
<td>1,000,000,000</td>
</tr>
<tr>
<td>$10^{12}$</td>
<td>tera</td>
<td>T</td>
<td>1,000,000,000,000</td>
</tr>
<tr>
<td>$10^{15}$</td>
<td>peta</td>
<td>P</td>
<td>1,000,000,000,000,000</td>
</tr>
<tr>
<td>$10^{18}$</td>
<td>exa</td>
<td>E</td>
<td>1,000,000,000,000,000,000</td>
</tr>
<tr>
<td>$10^{21}$</td>
<td>zetta</td>
<td>Z</td>
<td>1,000,000,000,000,000,000,000</td>
</tr>
</tbody>
</table>
## Binary prefixes

<table>
<thead>
<tr>
<th>$2^n$</th>
<th>Prefix</th>
<th>Symbol</th>
<th>Decimal</th>
</tr>
</thead>
<tbody>
<tr>
<td>$2^0$</td>
<td>none</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>$2^{10}$</td>
<td>kibi</td>
<td>Ki</td>
<td>1024</td>
</tr>
<tr>
<td>$2^{20}$</td>
<td>mebi</td>
<td>Mi</td>
<td>1,048,576</td>
</tr>
<tr>
<td>$2^{30}$</td>
<td>gibi</td>
<td>Gi</td>
<td>1,073,741,824</td>
</tr>
<tr>
<td>$2^{40}$</td>
<td>tebi</td>
<td>Ti</td>
<td>1,099,511,627,776</td>
</tr>
<tr>
<td>$2^{50}$</td>
<td>pebi</td>
<td>Pi</td>
<td>1,125,899,906,842,624</td>
</tr>
<tr>
<td>$2^{60}$</td>
<td>exbi</td>
<td>Ei</td>
<td>1,152,921,504,606,846,976</td>
</tr>
<tr>
<td>$2^{70}$</td>
<td>zebi</td>
<td>Zi</td>
<td>1,180,591,620,717,411,303,424</td>
</tr>
</tbody>
</table>
Prefixes in Computer Science

- Both decimal and binary prefixes are used in Computer Science

- Decimal prefixes are preferred because they are easier to calculate, however binary prefixes are more accurate

<table>
<thead>
<tr>
<th>Binary prefix</th>
<th>Decimal prefix</th>
<th>Value (bytes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>8 bits</td>
<td>1 byte</td>
<td>same</td>
</tr>
<tr>
<td>1 KiB</td>
<td>1 KB</td>
<td>$1024 \neq 1000$</td>
</tr>
<tr>
<td>(1 x $2^{10}$ bytes)</td>
<td>(1 x $10^3$ bytes)</td>
<td></td>
</tr>
<tr>
<td>1 MiB</td>
<td>1 MB</td>
<td>$1,048,576 \neq 1,000,000$</td>
</tr>
<tr>
<td>(1 x $2^{20}$ bytes)</td>
<td>(1 x $10^6$ bytes)</td>
<td></td>
</tr>
</tbody>
</table>
Example - hard disk sizes

- A 160GB hard disk is equivalent to 149.01GiB
  - 160GB = 160 \times 10^9
  - 149.01GiB = \frac{(160 \times 10^9)}{2^{30}}
Exercises

- Which has more bytes, 1KB or 1KiB?

- How many bytes are in 128MB?
Summary

- Computers use the binary number system
  - We can convert numbers between decimal and binary
- Decimal prefixes and binary prefixes are used for counting large numbers of bytes