# Bits, bytes and digital information

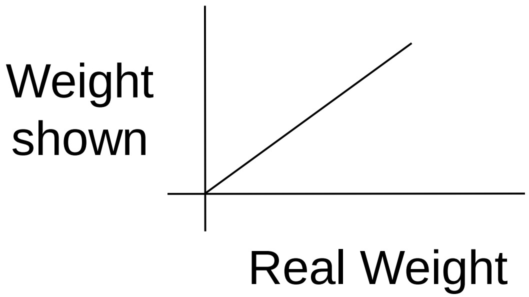
Lecture 2 – COMPSCI111/111G SS 2016

# Today's lecture

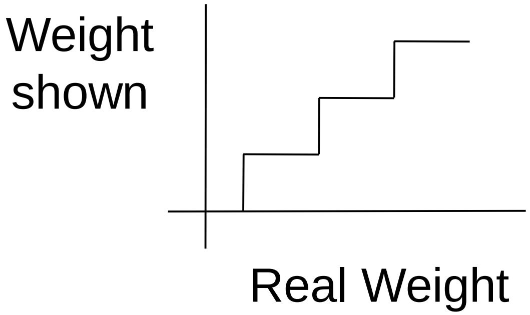
- Understand the difference between analogue and digital information
- Convert between decimal numbers and binary numbers

# Analogue vs digital information

- Information in the real world is continuous
  - Continuous signal

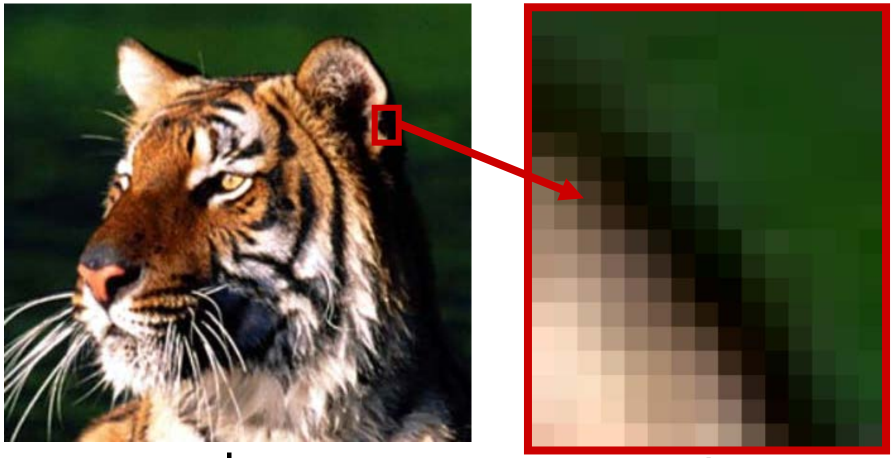



- Information stored by a computer is digital
  - Represented by discrete numbers



# Encoding information

- Real world information is stored by a computer using numbers
- Visual information



Image

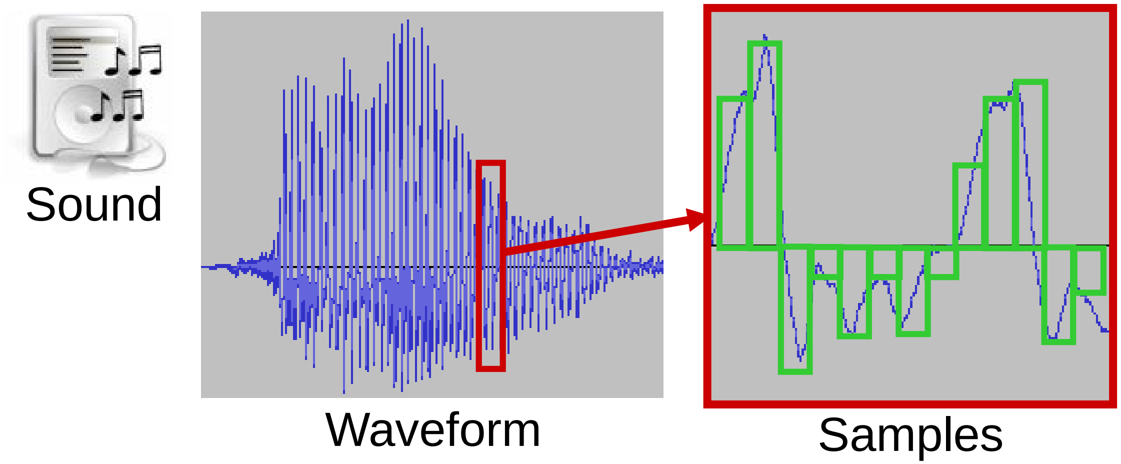
Pixels

```
11111111111111111111111
01111111111111111111111
00001111111111111111111
00000011111111111111111
00000000011111111111111
44444000001111111111111
75444000000011111111111
55554401000000111111111
33367544000000011111111
22283554444000000111111
99928357544000000011111
99999233657504000001111
99999983666554400000011
99999928338674400000001
```

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

# Encoding information

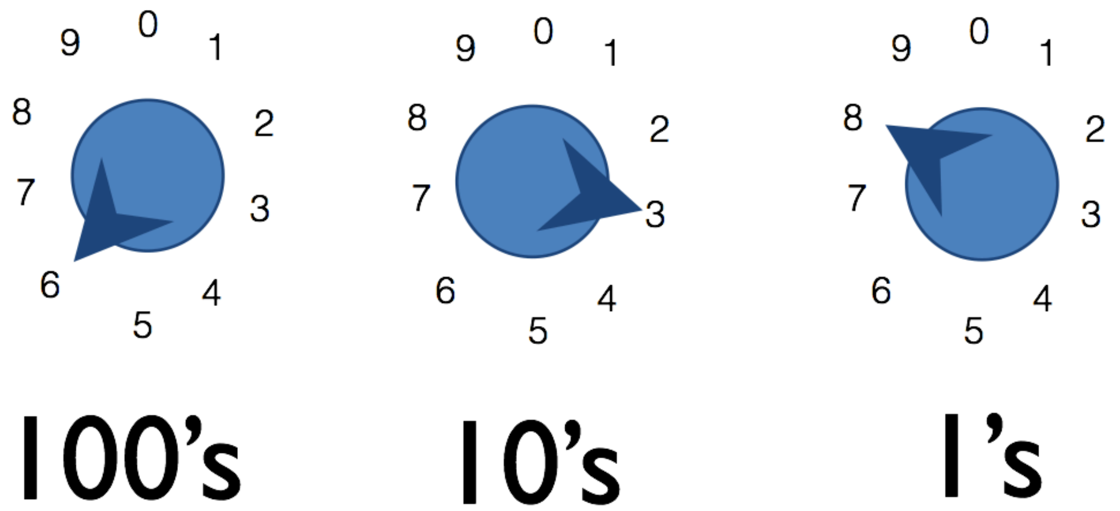
- Sound information




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

# Decimal numbers

- The decimal number system is a **base 10** system
- You can think about it as a dial with 10 positions:




**600 + 30 + 8 = 638**

# Decimal numbers

- The number of dials corresponds to the numbers that can be generated
- So:
  - Possible numbers = $10^n$
  - Range = 0 to $10^n$-1
- For example, if we have four dials…
  - Therefore:
    - $10^4$ = 10,000 possible numbers
      - Note 10 = base 10 and 4 = number of dials
  - Range = 0 to 9999 (ie. 0 to $10^4$-1)

# Binary numbers

- A number whose value is either 0 or 1
- Too complex to create 10 states in electronic circuitry. Much easier if we have two states like a switch, ON and OFF
- This is how binary numbers work; 0 usually means OFF and 1 usually means ON

0

1

# Binary numbers

- Each binary number is called a **bit (binary digit)**
- Using strings of bits, we can represent any whole number
- Using one switch (ie. one bit) we can generate up to two numbers (ie. 0 and 1)

# Binary numbers

- Using two switches (ie. two bits) we can generate up to four numbers

| | Binary | Decimal |
|---|---|---|
|  | 00 | 0 |
|  | 01 | 1 |
|  | 10 | 2 |
|  | 11 | 3 |

# Binary numbers

- So:
  - Possible numbers = $2^n$
  - Range = 0 to $2^n$-1
- For example, if we have four switches...
  - Therefore:
    - $2^4$ = 16 possible numbers
      - Note 2 = base 2 and 4 = number of switches
  - Range = 0 to $2^4$-1:
    - $0000_2$ to $1111_2$
    - $0_{10}$ to $15_{10}$

# Converting binary to decimal

- With decimal numbers, each dial's position has a value:

$1 * 10^3 \quad + \quad 5 * 10^2 \quad + \quad 2 * 10^1 \quad + \quad 1 * 10^0$

$1000 \quad + \quad 500 \quad + \quad 20 \quad + \quad 1$

$= 1521_{10}$

- Similarly with binary numbers, each switch's position has a value. Convert $1101_2$ to decimal:

$1 * 2^3 \quad + \quad 1 * 2^2 \quad + \quad 0 * 2^1 \quad + \quad 1 * 2^0$

$1 * 8 \quad + \quad 1 * 4 \quad + \quad 0 * 2 \quad + \quad 1 * 1$

$= 13_{10}$

# Converting binary to decimal

- Convert $10011_2$ to decimal
- Convert $35_{10}$ to binary

# 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

| $10^n$ | Prefix | Symbol | Decimal |
|---|---|---|---|
| 1 | none | | 1 |
| $10^3$ | kilo | K | 1000 |
| $10^6$ | mega | M | 1,000,000 |
| $10^9$ | giga | G | 1,000,000,000 |
| $10^{12}$ | tera | T | 1,000,000,000,000 |
| $10^{15}$ | peta | P | 1,000,000,000,000,000 |
| $10^{18}$ | exa | E | 1,000,000,000,000,000,000 |
| $10^{21}$ | zetta | Z | 1,000,000,000,000,000,000,000 |
| $10^{24}$ | yotta | Y | 1,000,000,000,000,000,000,000,000 |

# Binary prefixes

| $2^n$ | Prefix | Symbol | Decimal |
|---|---|---|---|
| $2^0$ | none | | 1 |
| $2^{10}$ | kibi | Ki | 1024 |
| $2^{20}$ | mebi | Mi | 1,048,576 |
| $2^{30}$ | gibi | Gi | 1,073,741,824 |
| $2^{40}$ | tebi | Ti | 1,099,511,627,776 |
| $2^{50}$ | pebi | Pi | 1,125,899,906,842,624 |
| $2^{60}$ | exbi | Ei | 1,152,921,504,606,846,976 |
| $2^{70}$ | zebi | Zi | 1,180,591,620,717,411,303,424 |
| $2^{80}$ | yobi | Yi | 1,208,925,819,614,629,174,706,176 |

# 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

| Binary prefix | Decimal prefix | Value (bytes) |
|---|---|---|
| 8 bits | 1 byte | same |
| 1 KiB | 1 KB | 1024 ≠ 1000 |
| 1 MiB | 1 MB | 1,048,576 ≠ 1,000,000 |

# Example – hard disk sizes

- A 160GB hard disk is equivalent to 149.01GiB
  - 160GB = $160 * 10^9$
  - 149.01GiB = $(160 * 10^9) / 2^{30}$


# Examples

- Which has more bytes, 1KB or 1KiB?
- How many bytes are in 128MB?
- What is the decimal prefix for $10^{12}$ bytes?

# 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