Bits, bytes and digital information

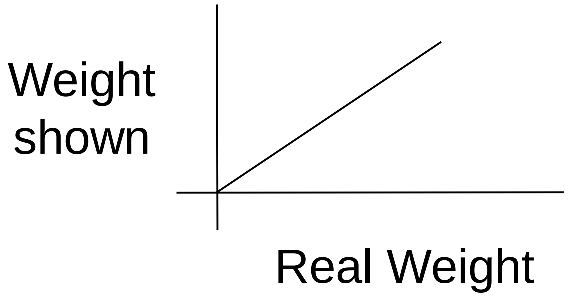
COMPSCI111/111G

Today's lecture

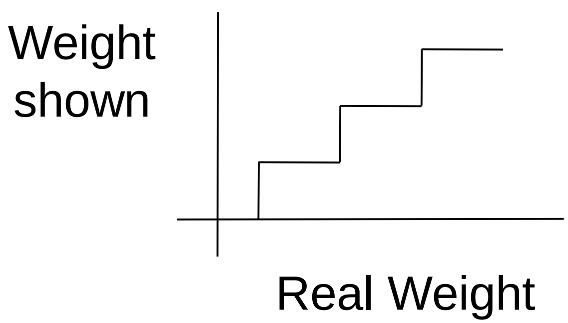
- Understand the difference between analogue and digital information
- Convert between decimal numbers and binary numbers

Analogue vs digital information

- Information in the real world is continuous
  - Continuous signal

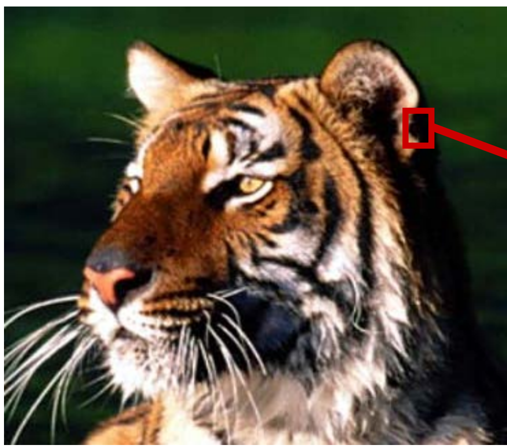


- Information stored by a computer is digital
  - Represented by discrete numbers

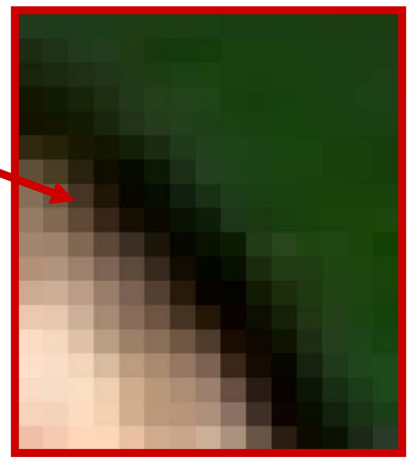


Encoding information

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



Image



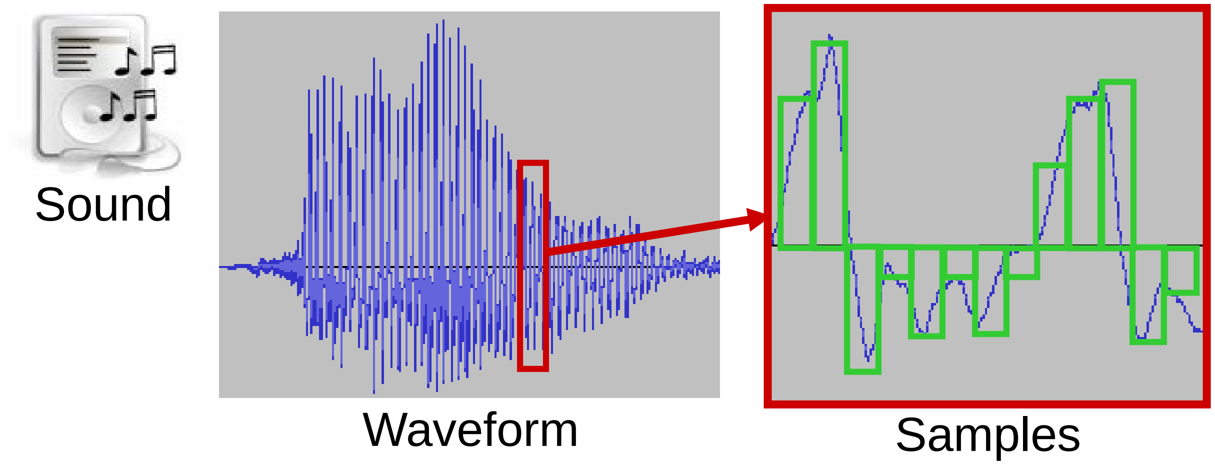
Pixels

```
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.

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.

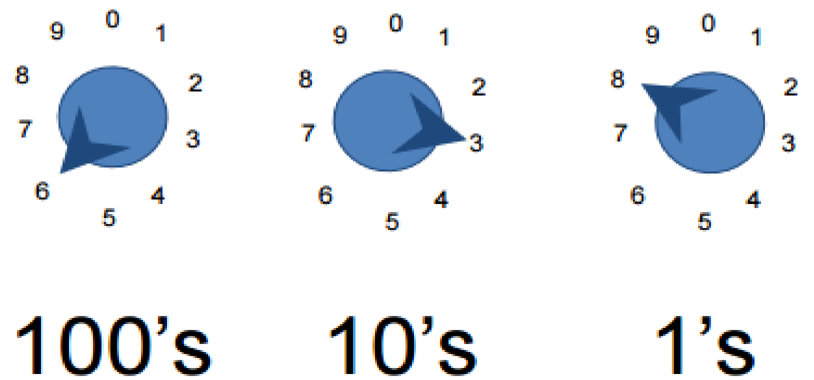
- We use the decimal number system:
  - 0 to 9 are decimal values.

Representing digital data

- At the lowest level, a computer is an electronic machine.
  - works by controlling the flow of electrons
- Easy to recognize two conditions:
  - presence of a voltage – we'll call this state "1"
  - absence of a voltage – we'll call this state "0"
- Could base state on value of voltage, but control and detection circuits much more complex.
  - compare turning on a light switch to measuring or regulating voltage

Representing Decimal Numbers

- We could use a series of dials
  - Each dial goes from 0 to 9.
- Information is stored discretely
  - Finite number of states – 10 per dial.
  - No in-between states.
- Decimal number system
  - 1st dial from right: 10^0
  - 2nd dial from right: 10^1
  - 3rd 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?

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?

  10000

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

  3

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.

Binary Digital System

Digital system:

- finite number of symbols

Binary (base two) system:

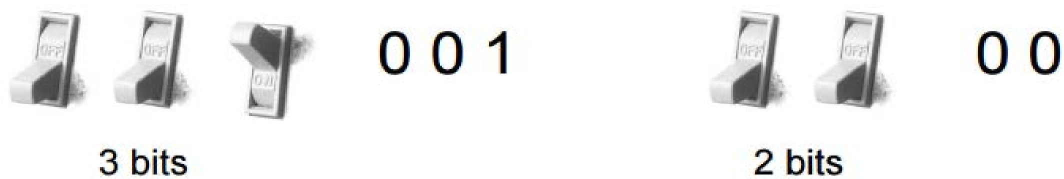
- has two states: 0 and 1

- Basic unit of information is the binary digit, or bit.
- Values with more than two states require multiple bits.
  - A collection of two bits has four possible states:
    00, 01, 10, 11
  - A collection of three bits has eight possible states:
    000, 001, 010, 011, 100, 101, 110, 111
  - A collection of n bits has 2n possible states.

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.

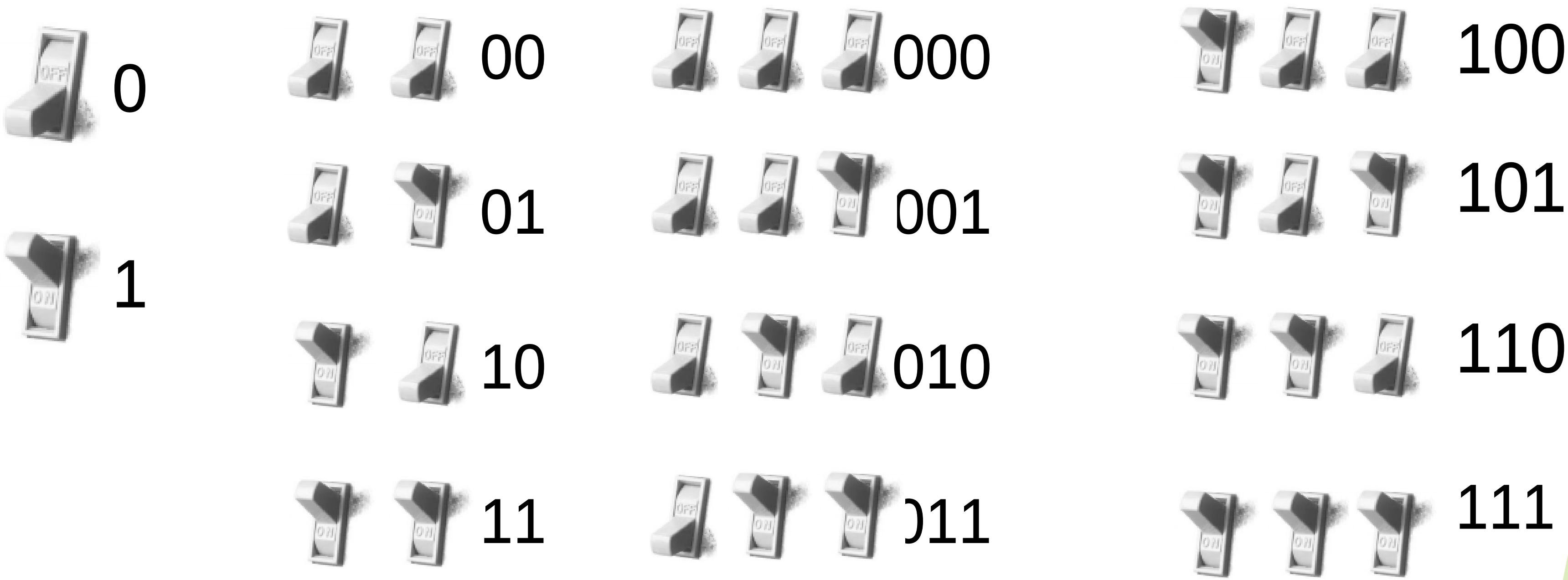


- A group of eight bits is referred to as a byte.

Using Binary Numbers

- How many different values/states can we have with:

1 bit: 2 bits: 3 bits:



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?
-

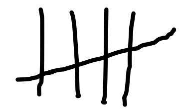
Exercises

- How many different values can we represent with a byte?
- 256

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

Integers

- Non-positional notation
  - could represent a number ("5") with a sequence of marks



- Weighted positional notation
  - like decimal numbers: "329"
  - "3" is worth 300, because of its position, while "9" is only worth 9

329

10^2 10^1 10^0

3x100 + 2x10 + 9x1 = 329

most significant → 101 ← least significant

2^2 2^1 2^0

1x4 + 0x2 + 1x1 = 5

Integers (cont.)

- An n-bit unsigned integer represents any of 2^n (integer) values from 0 to 2^{n-1}.

2^2	2^1	2^0	Value
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7

Converting binary to decimal

Convert the number 110 from binary to decimal

2^5	2^4	2^3	2^2	2^1	2^0	
32	16	8	4	2	1	
			1	1	0	
			1 x 4	1 x 2	0 x 1	
			4	2	0	= 6

Converting binary to decimal

Convert the number 10110 from binary to decimal

2^5	2^4	2^3	2^2	2^1	2^0	
32	16	8	4	2	1	
	1	0	1	1	0	
	1 x 16	0 x 8	1 x 4	1 x 2	0 x 1	
	16	0	4	2	0	= 22

Converting decimal to binary

- Put a 1 in the most significant column less than N
- Calculate remainder = (N – value)
- Repeat with remainder

Example: Convert 29 to binary

2^5	2^4	2^3	2^2	2^1	2^0	
32	16	8	4	2	1	
	1	1	1	0	1	
	1 x 16	1 x 8	1 x 4	0 x 2	1 x 1	
	16	8	4	0	1	= 29

Exercises

- What is the decimal equivalent of 101111?
- What is the binary equivalent of 123?

Exercises

- What is the decimal equivalent of 101111?
- 47
- What is the binary equivalent of 123?
- 1111011

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

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

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 x 2^{10} bytes)	1 KB (1 x 10^3 bytes)	1024 ≠ 1000
1 MiB (1 x 2^{20} bytes)	1 MB (1 x 10^6 bytes)	1,048,576 ≠ 1,000,000

Example – hard disk sizes

- A 160GB hard disk is equivalent to 149.01GiB
  - 160GB = 160 x 109
  - 149.01GiB = (160 x 109) ÷ 230

Exercises

- Which has more bytes, 1KB or 1KiB?
  -
- How many bytes are in 128MB?
  -

Exercises

- Which has more bytes, 1KB or 1KiB?
  - 1KB = 1000 bytes while 1KiB = 1024 bytes

- How many bytes are in 128MB?
  - 128 x 106 = 128,000,000 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