Bits and Bytes

Note: you are required to have the HOMEWORK BOOK with you in EVERY Wednesday lesson of this unit!

Objectives (spec links)

- Bit Patterns in a Computer
  - Explain the different interpretations that may be associated with a pattern of bits.
- Pure Binary Representation of Denary Integers
  - Describe the representation of unsigned denary integers in binary.
  - Perform conversion from denary to binary and vice-versa.
  - Binary arithmetic
- The Concept of Number Bases: Denary, Binary and Hexadecimal
  - Describe the conversion of a denary integer to hexadecimal form and vice versa.
  - Describe the use of hexadecimal as shorthand for binary.
- Integers and numbers with a fractional part
  - Draw a distinction between integers and numbers with a fractional part in a computer context.
  - Describe how an unsigned denary number with a fractional part is represented in fixed-point form in binary.

Data vs Information

- Unprocessed raw material held in a computer
  - 08976 (some meter reading)
- Processed data that has been given structure or meaning
  - Meter reading turned into a gas bill

Direct and indirect sources

- **Direct**: collected for a specific purpose. Addresses collected to send out gas bills
- **Indirect**: data/information collected for a different purpose. Addresses from gas company used for junk mail purposes.

What can be stored in computer?

- Text
- Numbers
- Sound
- Graphics

Binary number system

- All digital computers use the binary system for representing data of all types –numbers, characters, sound, pictures etc.
- **Bit** = **B**inary Dig**it**, 0 or 1
- **Byte** = **B**inar**y te**rm
  - Unit of storage
  - 8 bits (normally one character)
  - Can represent 256 different characters – character coding schemes to be covered later (eg ASCII, Unicode).

Denary Number System

- Commonly used number system
- **10** digits (0 to 9)
- Move from right to left
- Each digit represents a place value (powers of **10**)
- E.g. 782
  - 782 = 7 x 100 + 8 x 10 + 2 x 1

$10^2 = 100$	$10^1 = 10$	$10^0 = 1$
7	8	2

- Known as Base 10 system

Binary Number System

- Uses 2 digits from 0 to 1
- Move from right to left
- Each digit represents a place value (power of 2)
- E.g. 13
  - $13 = 1 \times 8 + 1 \times 4 + 0 \times 2 + 1 \times 1$
  - $(13)_{10} = (1101)_2$

$2^3 = 8$	$2^2 = 4$	$2^1 = 2$	$2^0 = 1$
1	1	0	1

- Also known as **Base 2** system
- Ideal for computers as 0 = off and 1 = on

Hexadecimal Number System

- "Hexa" means 16, so using 16 characters:
  - digits 0-9
  - letters A-F
- Each digit represents a place value (powers of **16**)
- E.g.
  - 179 = B x 16 + 3 x 1
  - $(179)_{10} = (B3)_{16}$
- Advantages:
  - Eases task of examining contents of memory or a computer file
  - Easier to represent large numbers (used to represent colour codes in web design)

$16^1 = 16$	$16^0 = 1$
B	3

Task 1

- Draw a table and write down the numbers 0 to 15 in Denary, Binary, Hexadecimal:

Denary	Binary	Hexadecimal

Denary to Binary Translation

- Translate 177 into binary:

  128 64 32 16 8 4 2 1

- Compare the denary number to the first (left) column of the binary number line (powers of 2)
- If it is greater than or equal to this number then write a 1 under it

128	64	32	16	8	4	2	1
1							

- Then subtract the column value from the denary number (177-128 = 49)

Denary to Binary Translation

- Take the new number and find the first column that the remainder is greater than or equal to and write a 1 under it

128	64	32	16	8	4	2	1
1		1					

- ...subtract the column value from the denary number (49-32=17)

Denary to Binary Translation

- Repeat the process until nothing is left

128	64	32	16	8	4	2	1
1		1	1				1

- …then fill in the blanks with 0's

128	64	32	16	8	4	2	1
1	0	1	1	0	0	0	1

- An odd number will **always** end in a one and an even number will **always** end with a 0

Translate to binary

- 127
- 86
- 124
- 17
- 19

In pairs, check each other's answers.

Translate to binary

127	=	01111111
86	=	01010110
124	=	01111100
17	=	00010001
19	=	00010011

Binary to denary translation

- Write the binary number below the place values (eg 01100011)

128	64	32	16	8	4	2	1
0	1	1	0	0	0	1	1

- look at the numbers with a 1 below and add the column headings :

64+32+2+1=99

- Always use this method to check a denary to binary translation

Translate to denary

- 10101010
- 11111111 (the largest 8 bit pure binary number)
- 00001110
- 01010101
- 11001100

In pairs, check each other's answers.

Translate to denary

- 10101010 = 170
- 11111111 = 255
- 00001110 = 14
- 01010101 = 85
- 11001100 = 204

Binary number range

- The smallest pure binary number that may be written in 8 bits is 00000000_2
- The largest unsigned number that may be written in 8 bits is 11111111_2

Or

- $2^8 - 1 = 255_{10}$

Or

- $2X2X2X2X2X2X2X2-1 = 255_{10}$

Hexadecimal numbers

- A way of writing large binary (or denary) numbers in a shorter number of digits

e.g. colour codes in HTML

Denary	Pure Binary	Hex
1	1	1
2	10	2
3	11	3
4	100	4
5	101	5
6	110	6
7	111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F
16	10000	10

Convert unsigned binary to Hexadecimal

- If the number is in denary then convert to binary first – MUCH easier conversion
- Break the binary number into 4-bit nibbles
- Translate each nibble into the hex equivalent (from previous table)

- **Example**: translate 11111111_2

- This can be split into 1111_2 1111_2
- The hex code for 1111_2 is F_{16}
- Therefore, the number is FF_{16}

Convert Hexadecimal to unsigned binary

- The quickest way to translate back to denary is to repeat the process in reverse.
- Convert each hex character to a 4-bit nibble
- Combine the nibbles into a single binary value

Example Translate AD into denary

A	D
1010	1101
10101101	

This may then be converted to denary if required.

Task

- Translate 124_{10} into:
  - Binary
  - Hexadecimal
- Translate 110010_2 into:
  - Denary
  - Hexadecimal
- Translate 1101111_2 into:
  - Hexadecimal

Homework

Homework booklet

Finish off

page 2-7