Learning Objectives

- Describe how computers represent 2’s complement signed numbers
- Convert between denary and 8 bit binary numbers (-128 to +127)

Two's complement

- Two's complement is a much more useful way of representing binary numbers as it allows you to represent negative numbers
- With 8 bits you can represent −128 to 127
- Because the MSB is used to represent the sign (+ or -)
- If MSB is a 1 = negative number
- If MSB is a 0 = positive number
- This method requires the MSB place value to be negative value

Number of bits	Value of MSB	Range
4	−8	−8 to +7
8	−128	−128 to +127
16	−32,768	−32,768 to +32,767

Place Value	-128	64	32	16	8	4	2	1
	0	0	0	0	0	0	1	1

Representing numbers using 2's complement

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

MSB

IF the MSB is a 0 then it will be a positive number

16 + 2 = 18

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

MSB

IF the MSB is a 1 then it will be a negative number

-128 + 18 = -110

Worked Example

- For example in 8 bits the MSB will be -128

Place Value	-128	64	32	16	8	4	2	1
	1	0	0	0	0	0	0	1

$$-128 + 1 = -127$$

Here are two 8-bit two's complement numbers. Seven of their eight bits are the same, but they have different MSBs.

Positive number	0	1	1	0	1	1	0	1
Negative number	1	1	1	0	1	1	0	1

The one with a 0 is a positive number.
The denary equivalent is: $0 + 64 + 32 + 8 + 4 + 1 = +109$.

The one with a 1 is a negative number.
Its denary equivalent is $(-128) + 64 + 32 + 8 + 4 = -20$.

Worked example

Convert the two's complement binary number 1110 0100 to denary.

Place values	-2^7	$+2^6$	$+2^5$	$+2^4$	$+2^3$	$+2^2$	$+2^1$	$+2^0$
	−128	**+64**	**+32**	**+16**	**+8**	**+4**	**+2**	**+1**
	1	1	1	0	0	1	0	0

The MSB is 1, indicating that it is a negative number.

The denary equivalent is: $(-128) + 64 + 32 + 4 = (-128) + 100 = -28$

Two's complement

- To store -103 we record -128 + 25 or

Column Value	-128	64	32	16	8	4	2	1
Binary number	1	0	0	1	1	0	0	1

- For example, to show −86 in binary, first work out **86**.
- 86 is 01010110 because 64 + 16 +4 + 2 equals 86:

Place Value	-128	64	32	16	8	4	2	1
Binary number	0	1	0	1	0	1	1	0

- Then start at the right-hand side and leave everything alone up to and including the first 1:

- Finally invert everything after this, so all the 1s become 0s and vice versa:

Place Value	-128	64	32	16	8	4	2	1
Binary number	1	0	1	0	1	0	1	0

Adding together two two's complement numbers

Add together these two two's complement numbers:

0010 1010 and 1101 0110

Ignoring the overflow, what result do you get?

The result is 1 0000 0000, which is −256. Ignoring the overflow gives 0000 0000.

Why do you think this is the case?

These numbers represent +42 and −42. They add to zero. It does this with an overflow bit, which is not stored in the result.

Binary Subtraction

64-12

	128	64	32	16	8	4	2	1
Step 1 work out + 12	0	0	0	0	1	1	0	0
Step 2 work out – 12	1	1	1	1	0	1	0	0
Step 3 work out 64	0	1	0	0	0	0	0	0
Step 4 Add together -12 and 64	0	0	1	1	0	1	0	0

Flip

ignore