In two's complement the left-most bit is changed to a negative value. For instance, for an 8-bit number, the value 128 is now changed to −128, but all the other headings remain the same. This means the new range of possible numbers is: −128 (10000000) to +127 (01111111).

It is important to realise when applying two's complement to a binary number that the left-most bit always determines the sign of the binary number. A 1-value in the left-most bit indicates a negative number and a 0-value in the left-most bit indicates a positive number (for example, 00110011 represents 51 and 11001111 represents −49).

Writing positive binary numbers in two's complement format

Example 1

The following two examples show how we can write the following positive binary numbers in the two's complement format 19 and 4:

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

As you will notice, for positive binary numbers, it is no different to what was done in Section 1.1.2.

Converting positive denary numbers to binary numbers in the two's complement format

If we wish to convert a positive denary number to the two's complement format, we do exactly the same as in Section 1.1.2:

Example 2

Convert **a** 38 **b** 125 to 8-bit binary numbers using the two's complement format.

a Since this number is positive, we must have a zero in the −128 column. It is then a simple case of putting 1-values into their correct positions to make up the value of 38:

−128	64	32	16	8	4	2	1
0	0	1	0	0	1	1	0

b Again, since this is a positive number, we must have a zero in the −128 column. As in part **a**, we then place 1-values in the appropriate columns to make up the value of 125:

−128	64	32	16	8	4	2	1
0	1	1	1	1	1	0	1

Converting positive binary numbers in the two's complement format to positive denary numbers

Example 3

Convert 01101110 in two's complement binary into denary:

−128	64	32	16	8	4	2	1
0	1	1	0	1	1	1	0

As in Section 1.1.2, each time a 1 appears in a column, the column value is added to the total. For example, the binary number (01101110) above has the following denary value: 64 + 32 + 8 + 4 +2 = 110.

Example 4

Convert 00111111 in two's complement binary into denary:

−128	64	32	16	8	4	2	1
0	0	1	1	1	1	1	1

As above, each time a 1 appears in a column, the column value is added to the total. For example, the binary number (00111111) above has the following denary value: 32 + 16 + 8 + 4 +2 + 1 = 63.

Activity 1.12

1 Convert the following positive denary numbers into 8-bit binary numbers in the two's complement format:

a 39 c 88 e 111 g 77 i 49
b 66 d 102 f 125 h 20 j 56

2 Convert the following binary numbers (written in two's complement format) into positive denary numbers:

	−128	64	32	16	8	4	2	1
a	0	1	0	1	0	1	0	1
b	0	0	1	1	0	0	1	1
c	0	1	0	0	1	1	0	0
d	0	1	1	1	1	1	1	0
e	0	0	0	0	1	1	1	1
f	0	1	1	1	1	1	0	1
g	0	1	0	0	0	0	0	1
h	0	0	0	1	1	1	1	0
i	0	1	1	1	0	0	0	1
j	0	1	1	1	1	0	0	0

Writing negative binary numbers in two's complement format and converting to denary

Example 1

The following three examples show how we can write negative binary numbers in the two's complement format:

−128	64	32	16	8	4	2	1
1	0	0	1	0	0	1	1

By following our normal rules, each time a 1 appears in a column, the column value is added to the total. So, we can see that in denary this is: −128 + 16 + 2 + 1 = −109.

−128	64	32	16	8	4	2	1
1	1	1	0	0	1	0	0

Similarly, in denary this number is −128 + 64 + 32 + 4 = −28.

−128	64	32	16	8	4	2	1
1	1	1	1	0	1	0	1

This number is equivalent to −128 + 64 + 32 + 16 + 4 + 1 = −11.

Note that a two's complement number with a 1-value in the -128 column must represent a negative binary number.

Converting negative denary numbers into binary numbers in two's complement format

Consider the number +67 in 8-bit (two's complement) binary format:

−128	64	32	16	8	4	2	1
0	1	0	0	0	0	1	1

Method 1

Now let's consider the number -67. One method of finding the binary equivalent to -67 is to simply put 1s in their correct places:

−128	64	32	16	8	4	2	1
1	0	1	1	1	1	0	1

−128 + 32 + 16 + 8 + 4 + 1 = −67

Method 2

However, looking at the two binary numbers above, there is another possible way to find the binary representation of a negative denary number:

first write the number as a positive binary value – in this case 67:	0 1 0 0 0 0 1 1
we then invert each binary value, which means swap the 1s and 0s around:	1 0 1 1 1 1 0 0
then add 1 to that number:	1
this gives us the binary for −67:	1 0 1 1 1 1 0 1

Example 2

Convert –79 into an 8-bit binary number using two's complement format.

Method 1

As it is a negative number, we need a 1-value in the –128 column.

–79 is the same as –128 + 49

We can make up 49 from 32 + 16 + 1; giving:

–128	64	32	16	8	4	2	1
1	0	1	1	0	0	0	1

Method 2

write 79 in binary:	0 1 0 0 1 1 1 1
invert the binary digits:	1 0 1 1 0 0 0 0
add 1 to the inverted number	1
thus giving –79:	1 0 1 1 0 0 0 1

–128	64	32	16	8	4	2	1
1	0	1	1	0	0	0	1

It is a good idea to practise both methods.

When applying two's complement, it isn't always necessary for a binary number to have 8 bits:

Example 3

The following 4-bit binary number represents denary number 6:

–8	4	2	1
0	1	1	0

Applying two's complement (1 0 0 1 + 1) would give:

–8	4	2	1
1	0	1	0

in other words: –6