Floating Point Binary Exercise 3

**Example 1** - How would -101.00011 be represented in binary floating point representation using 16 bits for the mantissa (including the sign bit) and 8 bits for the exponent?

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Fixed Point** | **Floating Point** | **Sign (1 bit)** | **Mantissa (15 bit)** | **Exponent (8 bit)** |
| -101.00011 | -0.10100011 x 211 | 1 | 101000110000000 | 00000011 |

For the floating point representation, the decimal point has moved 3 places to the **left**. However, as this needs to be represented in binary, you need to convert 3 into:

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

(we can ignore the six 0s at the start of this number and just use 11)

The sign bit indicates whether the number is positive or negative. If the number is positive, then the sign bit must be 0. If the number is negative, then the sign bit must be 1. In this case the sign bit is 1 because the number is negative (-101.00011).

The full number after the decimal point is used as the mantissa. This is 10100011, which uses 8 bits in total. However, as we are required to use 15 bits for the mantissa, you need to add **seven** 0s at the **end** of the number. This is 101000110000000.

Similarly with the exponent, we already know that we are moving 3 (11, which uses 3 bits in total) places. As we are using 8 bits, we need to add **six** 0s at the **start** of the number. This is 00000011.

**Example 2** – How would -1.10111 be represented in binary floating-point representation using 16 bits for the mantissa (including the sign bit) and 8 bits for the exponent?

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Fixed Point** | **Floating Point** | **Sign (1 bit)** | **Mantissa (15 bit)** | **Exponent (8 bit)** |
| -1.10111 | -0.110111 x 21 | 1 | 110111000000000 | 00000001 |

For the floating point representation, the decimal point has moved 1 place to the left. This is represented the same way in binary.

Again, the sign bit is 1 because the number is negative.

The full number after the decimal point is used as the mantissa. This is 110111, which uses 6 bits in total. However, as we are required to use 15 bits for the mantissa, you need to add **nine** 0s at the end of the number. This is 110111000000000.

Similarly with the exponent, we already know that we are moving 1 place. As we are using 8 bits, we need to add **seven** 0s at the **start** of the number. This is 00000001.

**EXERCISE**

Complete the table below:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Fixed Point** | **Floating Point** | **Sign (1 bit)** | **Mantissa (15 bit)** | **Exponent (8 bit)** |
| -100.00011 |  |  |  |  |
| -1001.0001 |  |  |  |  |
| -11100.111 |  |  |  |  |
| -101110.01 |  |  |  |  |
| -1000001.1 |  |  |  |  |