Floating Point Binary Exercise 1

At Higher level, you will be required to represent binary numbers in scientific notation using a mantissa and exponent. Scientific notation used x 10n because denary uses 10 different digits. Binary only uses 2 different digits so you will now see x 2n being used.

**Example 1** - How would 1101.0011 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)** |
| 1101.00111 | 0.110100111 x 2100 | 0 | 110100111000000 | 00000100 |

For the floating point representation, the decimal point has moved 4 places to the left. However, as this needs to be represented in binary, you need to convert 4 into:

|  |  |  |
| --- | --- | --- |
| **4** | **2** | **1** |
| 1 | 0 | 0 |

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 0 because the number is positive.

The full number after the decimal point is used as the mantissa. This is 110100111, which uses 9 bits in total. However, as we are required to use 15 bits for the mantissa, you need to add 6 0s at the end of the number. This is 110100111000000.

Similarly with the exponent, we already know that we are moving 4 (100, which uses 3 bits in total) places. As we are using 8 bits, we need to add 5 0s at the start of the number. This is 00000100.

**Example 2** – How would 111101011.01 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)** |
| 111101011.01 | 0.11110101101 x 21001 | 0 | 111101011010000 | 00001001 |

For the floating point representation, the decimal point has moved 9 places to the left. However, as this needs to be represented in binary, you need to convert 9 into:

|  |  |  |  |
| --- | --- | --- | --- |
| **8** | **4** | **2** | **1** |
| 1 | 0 | 0 | 1 |

Again, the sign bit is 0 because the number is positive.

The full number after the decimal point is used as the mantissa. This is 11110101101, which uses 11 bits in total. However, as we are required to use 15 bits for the mantissa, you need to add 4 0s at the end of the number. This is 111101011010000.

Similarly with the exponent, we already know that we are moving 9 (1001, which uses 4 bits in total) places. As we are using 8 bits, we need to add 4 0s at the start of the number. This is 00001001.

**EXERCISE**

Complete the table below:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Fixed Point** | **Floating Point** | **Sign (1 bit)** | **Mantissa (15 bit)** | **Exponent (8 bit)** |
| 1101.10001 |  |  |  |  |
| 10001.0111 |  |  |  |  |
| 100100.101 |  |  |  |  |
| 1111001.11 |  |  |  |  |
| 11110111.1 |  |  |  |  |