1.6 Write the number -0.625 in the following forms (in part (*c*) follow the IEEE-754 standard):
(*a*) Binary form. (*b*) Base 2 floating point representation. (*c*) 32 bit single-precision string.

Solution

(*a*) The largest power of 2 that can be divided into 0.625 is $2^{-1} = 0.5$. Next subtract: $0.625 - 2^{-1} = 0.125$. Now the largest power of 2 that can be divided into 0.125 is 2^{-3}. Thus, the number -0.625 in binary form is -0.101.

(*b*) Using part (*a*), the binary floating point representation of -0.625 is:

$$-\frac{0.625}{2^{-1}} \times 2^{-1} = -\frac{0.625}{0.5} \times 2^{-1} = 1.25 \times 2^{-1}$$

(*c*) According to the IEEE-754 standard, -0.625 in single precision form is as follows:

- Since the number is negative, the first bit is 1
- From part (*b*), the exponent is -1. Adding a bias of 127, the value of the exponent that must be stored is -1+127 = 126. The number 126 in binary form is:

$$\begin{aligned} 126 &= 1 \times 2^6 + 1 \times 2^5 + 1 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 \\ &= 64 + 32 + 16 + 8 + 4 + 2 \end{aligned}$$

  Thus the number 126 in binary form is 1111110. In single precision, 8 bits can be used to store the exponent so that 126 is stored as 01111110 without the need for rounding or chopping.
- Next, the mantissa 0.25 is converted to binary form: 1×2^{-2} or, 0.01
- Since 23 bits are allocated for the mantissa, the binary number stored is 01000000000000000000000

Thus, the number -0.625 in single precision is stored as: |1|01111110|01000000000000000000000|.