Single Precision Floating Point

Summary

A single precision floating point number is a 32 bit floating point number. It is composed of a 1 ***sign bit***, an 8 bit ***exponent***, and a 23 bit ***mantissa***.

Definitions

- **Sign Bit** The sign bit is the first bit of a floating point number. It is 0 if the number is positive, and 1 if the number is negative.
- **Exponent** The exponent is the second part of a floating point number. It is an 8 bit number that is used to represent the power of 2 that the mantissa is multiplied by. The exponent is stored in excess 127 notation. This means that the exponent is stored as the exponent plus 127. For example, if the exponent is 10000000, the actual exponent is 0. If the exponent is 10000001, the actual exponent is 1. If the exponent is 01111111, the actual exponent is -127. If the exponent is 11111111, the actual exponent is -0.
- **Mantissa** The mantissa is the third part of a floating point number. It is a 23 bit number that is used to represent the number that is multiplied by 2 to the power of the exponent. The mantissa is stored in normalized form. This means that the first bit of the mantissa is always 1. This allows the mantissa to be stored in 23 bits instead of 24 bits. For example, if the mantissa is 10000000000000000000000, the actual mantissa is 1. If the mantissa is 10000000000000000000001, the actual mantissa is 1.00000000000000000000001. If the mantissa is 01111111111111111111111, the actual mantissa is 0.11111111111111111111111. If the mantissa is 11111111111111111111111, the actual mantissa is 1.11111111111111111111111.

Example

The following is an example of a single precision floating point number:

```
0     10000000 10000000000000000000000
|     |        |
sign exponent mantissa
```

$(-1)^{sign} = (-1)^0 = 1$
$2^{exponent-127} = 2^{128-127} = 2^1$
$1.F = 1.10000000000000000000000 = 1.5$
$\therefore 1 * 2 * 1.5 = 3$

Tricky Things to Remember

- The exponent is stored in excess 127 notation.
- The mantissa is stored in normalized form.
- The first bit of the mantissa is always 1.

Double Precision Floating Point

Summary

A double precision floating point number is a 64 bit floating point number. It is composed of a 1 ***sign bit***, an 11 bit ***exponent***, and a 52 bit ***mantissa***.

Definitions

- **Sign Bit** The sign bit is the first bit of a floating point number. It is 0 if the number is positive, and 1 if the number is negative.
- **Exponent** The exponent is the second part of a floating point number. It is an 11 bit number that is used to represent the power of 2 that the mantissa is multiplied by. The exponent is stored in excess 1023 notation. This means that the exponent is stored as the exponent plus 1023. For example, if the exponent is 10000000000, the actual exponent is 0. If the exponent is 10000000001, the actual exponent is 1. If the exponent is 01111111111, the actual exponent is -1023. If the exponent is 11111111111, the actual exponent is -0.
- **Mantissa** The mantissa is the third part of a floating point number. It is a 52 bit number that is used to represent the number that is multiplied by 2 to the power of the exponent. The mantissa is stored in normalized form. This means that the first bit of the mantissa is always 1. This allows the mantissa to be stored in 52 bits instead of 53 bits. For example, if the mantissa is 1000000000000000000000000000000000000000000000000000, the actual mantissa is 0.

Example

The following is an example of a double precision floating point number:

```
0    10000000000  1000000...0000000000
|    |            |
sign exponent     mantissa
```

$(-1)^{sign} = (-1)^0 = 1$
$2^{exponent-1023} = 2^{1024-1023} = 2^1$
$1.F = 1.100000000000000...000 = 1.5$
$\therefore 1 * 2 * 1.5 = 3$

Half Precision Floating Point

Summary

A half precision floating point number is a 16 bit floating point number. It is composed of a 1 ***sign bit***, a 5 bit ***exponent***, and a 10 bit ***mantissa***.

Definitions

- **Sign Bit** The sign bit is the first bit of a floating point number. It is 0 if the number is positive, and 1 if the number is negative.
- **Exponent** The exponent is the second part of a floating point number. It is a 5 bit number that is used to represent the power of 2 that the mantissa is multiplied by. The exponent is stored in excess 15 notation. This means that the exponent is stored as the exponent plus 15. For example, if the exponent is 10000, the actual exponent is 0. If the exponent is 10001, the actual exponent is 1. If the exponent is 01111, the actual exponent is -15. If the exponent is 11111, the actual exponent is -0.
- **Mantissa** The mantissa is the third part of a floating point number. It is a 10 bit number that is used to represent the number that is multiplied by 2 to the power of the exponent. The mantissa is stored in normalized form. This means that the first bit of the mantissa is always 1. This allows the mantissa to be stored in 10 bits instead of 11 bits. For example, if the mantissa is 1000000000, the actual mantissa is 1. If the mantissa is 1000000001, the actual mantissa is 1.0000000001. If the mantissa is 0111111111, the actual mantissa is 0.1111111111. If the mantissa is 1111111111, the actual mantissa is 1.1111111111.

Example

The following is an example of a half precision floating point number:

```
0    10000    1000000000
|    |        |
sign exponent mantissa
```

$(-1)^{sign} = (-1)^0 = 1$
$2^{exponent-15} = 2^{16-15} = 2^1$
$1.F = 1.1000000000 = 1.5$
$\therefore 1 * 2 * 1.5 = 3$