Contents

[1. Objective 2](#_Toc148310776)

[2. History 2](#_Toc148310777)

[2.1. Fixed floating point 2](#_Toc148310778)

[3. Number to Floating point - IEEE 754 Single precision 32-bit 3](#_Toc148310779)

[3.1. Check sign bit 3](#_Toc148310780)

[3.2. Convert the whole number part 3](#_Toc148310781)

[3.3. Convert the decimal part 3](#_Toc148310782)

[3.4. Combine back the Base 2 4](#_Toc148310783)

[3.5. Convert to Exponent format (shift the decimal point) 4](#_Toc148310784)

[3.6. Round up (round to 23 bit) 4](#_Toc148310785)

[3.7. Find the exponent value 4](#_Toc148310786)

[3.8. Convert exponent to base 2 4](#_Toc148310787)

[4](#_Toc148310788)

[3.9. Slot in the base 2 exponent 4](#_Toc148310789)

[3.10. Slot in the mantissa 4](#_Toc148310790)

[4. Floating point to Number - IEEE 754 Single precision 32-bit 4](#_Toc148310791)

# Objective

* Learn how floating numbers are stored in most modern computers, IEEE754 format

# History

# Fixed floating point

# Number to Floating point - IEEE 754 Single precision 32-bit

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

* 1-bit (sign bit)
* 8-bit (exponent)
* 23-bit (mantissa)

Example: Convert -66.65

### Check sign bit

-66.65 is a negative, therefore the sign bit is 1

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

### Convert the whole number part

2 | 66 | Remainder  
| 33 | 0 (33 >= 2, continue)  
| 16 | 1 (16 >= 2, continue)  
| 8 | 0 (8 >= 2, continue)  
| 4 | 0 (4 >= 2, continue)  
| 2 | 0 (2 >= 2, continue)  
| 1 | 0 (1 < 2, stop)  
| 1

### Convert the decimal part

0.65 \* 2 = 1.30 - 1  
0.30 \* 2 = 0.6 - 0  
0.60 \* 2 = 1.2 - 1  
0.20 \* 2 = 0.4 - 0  
0.40 \* 2 = 0.8 - 0  
0.80 \* 2 = 1.6 - 1  
0.60 \* 2 = 1.2 - 1 (stop, a repeat pattern have appeared)

0.65 = 0.101001 and 1001 … repeating pattern (read from top down)  
0.65 = 0.10100110011001100110011

(For 32-bit floating point, repeat the pattern and fill up for 23-bit)

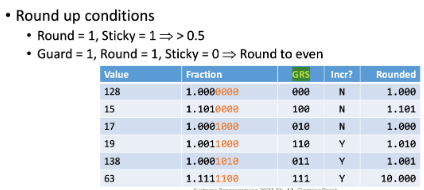
Check:

### Combine back the Base 2

66.65 = 1000010.10100110011001100110011

### Convert to Exponent format (shift the decimal point)

### Round up (round to 23 bit)



1.00001010100110011001100\_110011  
1.00001010100110011001101\_~~110011~~

### Find the exponent value

N + 127 = E  
6 + 127 = 133 (N = 6)

### Convert exponent to base 2 and slot in

### 

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

### Slot in the mantissa

1.0000\_1010\_1001\_1001\_1001\_101

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 |

-66.65 = 1\_10000101\_00001010100110011001101

# Floating point to Number - IEEE 754 Single precision 32-bit

The reverse order of the conversion