Comp 255: Floating Point Representation Examples

8-bit floating point example

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
+-----+-----+-----+
|     |     |     |
sign  exponent  fraction

sign: 0 = plus / 1 = minus
exponent: excess 3 (011) notation (range is -3 to 4)
fraction: normalized so leading 1 is right of implied binary point (or binary point to left of leading 1)
precision depends on exponent

Example: 5.25 -> 101.01 -> 0.10101×2¹¹ -> 0 | 110 | 1010

Largest value is 0 11111111 -> 0.1111×2¹⁰ -> 1111.0 = 15 (decimal)
Smallest non-zero value is 000001000 -> 0.1×2⁻¹¹ = 0.0001 = 1/16 = 0.0625
Note: 0 0000 0000 is zero

Not all bit patterns can be used

8-bit floating point example – with hidden bit

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
+-----+-----+-----+
|     |     |     |
sign  exponent  fraction

sign: 0 = plus / 1 = minus
exponent: excess 3 (011) notation (range is -3 to 4)
fraction: hidden 1 bit to left of implied binary point (or binary point to right of leading 1)
precision depends on exponent

Example: 5.25 -> 101.01 -> 1.0101×2¹⁰ -> 0 | 101 | 0101

Largest value is 0 11111111 -> 1.1111×2¹⁰ = 11111 = 31 (decimal)
Smallest non-zero value is 0 0000 0001 = 1.0001×2⁻¹¹ = 0.0010001 = 1/8 + 1/128 = 0.1328125
Note: 0 0000 0000 is zero
IEEE 754 Standard for Single Precision Floating Point Numbers

|s|    exponent   |            fraction                         |
|+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+|
| 3 3 2 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0|
| 1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0|

Sign: 0 = plus / 1 = minus
Exponent: excess 127 (01111111) 00000000 and 11111111 are reserved patterns
Therefore exponent range is -126 .. +127
Fraction: hidden 1 bit to left of implied binary point (or binary point to right of leading 1)
precision depends on exponent

5 basic types

Non-zero normalized numbers (exponents between -126 and +127)
Clean zero: exponent and mantissa all 0’s (note +0 and -0)
Infinity: exponent = 11111111, fraction all 0’s (+ and – infinity)
NaN (not a number): exponent = 11111111 and fraction is not all 0’s
Denormalized: exponent = 00000000 (-126) and fraction is actual value w/o/ hidden bit