Floating Point Representation

Fractional Binary Numbers
IEEE Floating Point Standard
Lessons for Programmers
### Fractional Binary Numbers

<table>
<thead>
<tr>
<th>position</th>
<th>weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>8</td>
</tr>
<tr>
<td>2</td>
<td>$2^2$</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>1</th>
<th>0</th>
<th>1</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>4</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>$2^3$</td>
<td>$2^2$</td>
<td>$2^1$</td>
<td>$2^0$</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
Fixed-Point Representation

Implied binary point.

\[
\begin{array}{cccccc}
\ b_7 & b_6 & b_5 & b_4 & b_3 & . & b_2 & b_1 & b_0 \\
\ b_7 & b_6 & b_5 & b_4 & b_3 & b_2 & b_1 & b_0 & .
\end{array}
\]

- **range**: difference between largest and smallest representable numbers
- **precision**: smallest difference between any two representable numbers

- **fixed point** = fixed range, fixed precision
**IEEE Floating Point** Standard 754

IEEE = Institute of Electrical and Electronics Engineers

**Numerical form:**

\[ V_{10} = (-1)^s \cdot M \cdot 2^E \]

- **Sign bit** `s` determines whether number is negative or positive
- **Significand (mantissa)** `M` usually a fractional value in range [1.0, 2.0)
- **Exponent** `E` weights value by a (-/+ power of two
  - Analogous to scientific notation

**Representation:**

- **MSB** `s` = sign bit `s`
- **exp** field encodes `E` (but is *not equal* to E)
- **frac** field encodes `M` (but is *not equal* to M)

Numerically well-behaved, but hard to make fast in hardware
Precisions

Single precision (float): 32 bits

- 1 bit for sign (s)
- 8 bits for exponent (exp)
- 23 bits for fraction (frac)

Double precision (double): 64 bits

- 1 bit for sign (s)
- 11 bits for exponent (exp)
- 52 bits for fraction (frac)

Finite representation of infinite range...
Three kinds of values

\[ V = (-1)^s \times M \times 2^E \]

1. Normalized: \( M = 1.xxxxx \ldots \)
   As in scientific notation: \( 0.011 \times 2^5 = 1.1 \times 2^3 \)
   Representation advantage?

2. Denormalized, near zero: \( M = 0.xxxxx \ldots \), smallest \( E \)
   - Evenly space near zero.

3. Special values:
   - **0.0:**
     \( s = 0 \quad \text{exp} = 00...0 \quad \text{frac} = 00...0 \)
   - **+\( \text{inf} \), -\( \text{inf} \):**
     \( \text{exp} = 11\ldots1 \quad \text{frac} = 00\ldots0 \)
   - **NaN (“Not a Number”):**
     \( \text{exp} = 11\ldots1 \quad \text{frac} \neq 00\ldots0 \)
   - \( \sqrt{-1}, \infty - \infty, \infty \times 0 \), etc.
Value distribution

-∞  Normalize  -Denormalized  +Denormalized  +Normalize  +∞

NaN  -0.0  NaN  +0.0  NaN
1. Normalized values, with float example

\[ V = (-1)^s \times M \times 2^E \]

Value: float \( f = 12345.0; \)

\[ 12345_{10} = 11000000111001_2 \]
\[ = 1.1000000111001_2 \times 2^{13} \quad \text{(normalized form)} \]

Significand:

\[ M = 1.1000000111001_2 \]
\[ \text{frac} = 1000000111001000000000000_2 \]

Exponent: \( E = \exp - \text{Bias} \rightarrow \exp = E + \text{Bias} \)

\[ E = 13 \]
\[ \text{Bias} = 127 = 2^7 - 1 = 2^{k-1} - 1 \]
\[ \exp = 140 = 10001100_2 \]

Splits exponents roughly -/+ 

Result:

\[ \begin{array}{c|c|c}
  s & \text{exp} & \text{frac} \\
  \hline
  0 & 10001100 & 1000000111001000000000000 \end{array} \]
2. Denormalized Values: near zero

- "Near zero": \( \exp = 000...0 \)

- Exponent:
  \[
  E = 1 + \exp - \text{Bias} = 1 - \text{Bias}
  \]

- Significand: leading zero
  \[
  M = 0.\text{xxx}...\text{x}_2
  \frac{\text{frac}}{\text{frac}} = \text{xxx}...\text{x}
  \]

- Cases:
  
  - \( \exp = 000...0, \frac{\text{frac}}{\text{frac}} = 000...0 \) \( 0.0, -0.0 \)
  
  - \( \exp = 000...0, \frac{\text{frac}}{\text{frac}} \neq 000...0 \)
Try to represent 3.14, 6-bit example

6-bit IEEE-like format

Bias = $2^{3-1} - 1 = 3$

Value: 3.14;

3.14 = 11.0010 0011 1101 0111 0000 1010 000... 
    = 1.1001 0001 1110 1011 1000 0101 000...  $2 \times 2^1$ (normalized form)

Significand:

$M = \overline{1.10010001111010111011100001010000... 2}$

$\frac{M}{2} = 10_2$

Exponent:

$E = 1$  Bias = 3  $exp = 4 = 100_2$

Result:

$0 \ 100 \ 10 = 1.10_2 \times 2^1 = 3$  next highest?
Floating Point Arithmetic*

\[ V = (-1)^{s} \times M \times 2^{E} \]

<table>
<thead>
<tr>
<th>s</th>
<th>exp</th>
<th>frac</th>
</tr>
</thead>
</table>

double x = ..., y = ...;
double z = x + y;

1. **Compute exact result.**
2. **Fix/Round**, roughly:
   - Adjust \( M \) to fit in \([1.0, 2.0)\)...
     - If \( M \geq 2.0 \): shift \( M \) right, increment \( E \)
     - If \( M < 1.0 \): shift \( M \) left by \( k \), decrement \( E \) by \( k \)
   - Overflow to infinity if \( E \) is too wide for \( \text{exp} \)
   - Round* \( M \) if too wide for \( \frac{\text{ac}}{} \).
     Underflow if nearest representable value is 0.

*complicated...
Lessons for programmers

\[ V = (-1)^s \times M \times 2^E \]

- `float` ≠ real number ≠ `double`
- Rounding breaks associativity and other properties.

```java
double a = ..., b = ...;
...
if (a == b) ...
```

X

```java
if (abs(a - b) < epsilon) ...
```