Floating Point Numbers
Fractional numbers

- Fractional numbers – fixed point
- Floating point numbers – the IEEE 754 floating point standard
- Floating point operations
- Rounding modes
Positional representation of fractional numbers

- In base 10

<table>
<thead>
<tr>
<th>Multiplier</th>
<th>Number</th>
<th>Position</th>
</tr>
</thead>
<tbody>
<tr>
<td>10³</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>10²</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>10¹</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>10⁰</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>10⁻¹</td>
<td>4</td>
<td>-1</td>
</tr>
<tr>
<td>10⁻²</td>
<td>5</td>
<td>-2</td>
</tr>
<tr>
<td>10⁻³</td>
<td>6</td>
<td>-3</td>
</tr>
<tr>
<td>10⁻⁴</td>
<td></td>
<td>-4</td>
</tr>
</tbody>
</table>

Decimal point
### Positional representation of fractional numbers

- **In base 2**

<table>
<thead>
<tr>
<th>$2^3$</th>
<th>$2^2$</th>
<th>$2^1$</th>
<th>$2^0$</th>
<th>$2^{-1}$</th>
<th>$2^{-2}$</th>
<th>$2^{-3}$</th>
<th>$2^{-4}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

- Multiplier: 1101
- Number: 1101
- Position: 3210-1-2-3-4

**Binary point**
Fractional numbers – fixed point

- Fixed-point representation
  - How much information is necessary to store?
  - How do you choose a format for the bits?

- Fixed-point operations
  - Addition
    - Align binary points, and add straight down
  - Multiplication
    - ???
Decimal to binary conversion

- Convert $A = 3.1415_{10}$ to base 2

```
<table>
<thead>
<tr>
<th>i</th>
<th>R</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

```
R = A
i = -1

R == 0

R = R*2

R >= 1

b_i = 0

i = i - 1

b_i = 1

R = R - 1

END
```
Fixed-point number density
Scientific notation

- In base 10
  - Example: $3.0 \times 10^8$

- In base 2
  - Example: $-1.00101 \times 2^4 = -18.5_{10}$

- The general form
  - $r = \text{Sign} \cdot \text{significand} \cdot \text{base}^{\text{Exponent}}$
Single-precision IEEE 754 floating-point numbers

- One-bit sign
- Eight-bit exponent
- 23-bit significand
  - That’s the fractional part
Single-precision IEEE 754 floating-point numbers

- Normalized numbers: only one non-zero bit to the left of the binary point
  - Adjust the exponent as needed
  - \( r = (-2)^S \cdot F \cdot 2^E \)

- Implicit leading 1 in the significand (the “hidden bit”)
  - \( r = (-2)^S \cdot (1 + F) \cdot 2^E \)

- Bias notation to represent the exponent
  - With the bias \( B = 127 \)
  - \( r = (-2)^S \cdot (1 + F) \cdot 2^{E-B} \)
How to convert a base-10 number into IEEE 754 single-precision floating point

- Convert the number to binary
  - The big part
  - And the fractional part
- Normalize
  - Isolate the hidden one
- Remove the significand’s hidden one
- Add bias to the exponent
- Represent the number
Example: 12.625

- Convert to binary
- Normalize
- Remove hidden one
- Add bias exponent
- The end
Double-precision IEEE 754 floating-point numbers

- Sign
- Exponent
- Significand

- One-bit sign
- Eleven-bit exponent
- 52-bit significand
  - That’s the fractional part
Summary of IEEE 754 formats

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Bits for F</td>
<td>24</td>
<td>≥ 32</td>
<td>53</td>
<td>≥ 64</td>
</tr>
<tr>
<td>Bits for E</td>
<td>8</td>
<td>≥ 11</td>
<td>11</td>
<td>≥ 15</td>
</tr>
<tr>
<td>$E_{max}$</td>
<td>+127</td>
<td>≥ +1023</td>
<td>+1023</td>
<td>≥ +16383</td>
</tr>
<tr>
<td>$E_{min}$</td>
<td>−126</td>
<td>≤ −1022</td>
<td>−1022</td>
<td>≤ −16382</td>
</tr>
<tr>
<td>Total bits</td>
<td>32</td>
<td>≥ 43</td>
<td>64</td>
<td>≥ 79</td>
</tr>
</tbody>
</table>

- For every precision, there are reserved exponents, used for special quantities:
  - $E_{min} − 1$ (i.e., $E=0$) is used for zero and denorms
  - $E_{max} + 1$ (i.e., $E=255$ or $E=2047$, with bias) is used for NaN and infinity
Special quantities: Infinity

- This special quantity avoids halt on overflow
  - Much safer than returning the largest possible number

- Representation:
  - $E = E_{max} + 1$
    - $E = 255$ in single precision with bias
    - $E = 2047$ in double precision with bias
  - $F = 0$
  - Sign ($+\infty$ or $-\infty$)
Special quantities: Infinity

- Examples of operations that return \( \text{Inf} \)

<table>
<thead>
<tr>
<th>Operation</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/0</td>
<td></td>
</tr>
<tr>
<td>-1/0</td>
<td></td>
</tr>
<tr>
<td>4 - ( \text{Inf} )</td>
<td></td>
</tr>
<tr>
<td>( \text{Sqrt} (+\text{Inf}) )</td>
<td></td>
</tr>
<tr>
<td>1 / ( \text{Inf} )</td>
<td></td>
</tr>
</tbody>
</table>
Special quantities: NaN (Not a Number)

- This special quantity avoids halt on invalid operations
- Representation:
  - $E = E_{max} + 1$
    - $E = 255$ in single precision with bias
    - $E = 2047$ in double precision with bias
    - $F \neq 0$
Special quantities: NaN (Not a Number)

- Examples of operations that return NaN

<table>
<thead>
<tr>
<th>Operation</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>+0/3</td>
<td></td>
</tr>
<tr>
<td>–0/3</td>
<td></td>
</tr>
<tr>
<td>3/+0</td>
<td></td>
</tr>
<tr>
<td>3/–0</td>
<td></td>
</tr>
<tr>
<td>1/–Inf</td>
<td></td>
</tr>
<tr>
<td>log(+0)</td>
<td></td>
</tr>
<tr>
<td>log(–0)</td>
<td></td>
</tr>
</tbody>
</table>
Special quantities: Zero

- **Representation:**
  - $E = E_{\text{min}} - 1$ (i.e., $E = 0$)
  - $F \neq 0$
  - Sign: $+0$ or $-0$
Special quantities: Zero

- Examples of operations that involve 0

<table>
<thead>
<tr>
<th>Operation</th>
<th>NaN produced by</th>
</tr>
</thead>
<tbody>
<tr>
<td>+</td>
<td></td>
</tr>
<tr>
<td>/</td>
<td></td>
</tr>
<tr>
<td>Sqrt</td>
<td></td>
</tr>
</tbody>
</table>
Floating point numbers range

- What is the largest number we can represent in IEEE 754 single-precision floating point?

- What is the smallest number?
Floating point numbers range

- What is the largest number we can represent in IEEE 754 double-precision floating point?
- What is the smallest number?
Floating point numbers: density

- **Fact 1:** Floats are not reals
  - E.g., 2/3
- **Fact 2:** Floats are not decimals
  - E.g., 0.1 (base 10) = 1.1001100… 2⁻⁴ (base 2)
- **Fact 3:** Not even all the integers in the range are represented
  - E.g., 100,000,001 (base 10) = 1011 1110 1011 1100 0010 0000 0001 (base 2)
Floating point numbers: density

- Close to 0: high density

- Far from 0: high density
Special quantities: Denormals

- These are numbers smaller than $2^{E_{\text{min}}}$
- Fill the gap between $2^{E_{\text{min}}}$ and 0 (gradual underflow)
- Representation:
  - $E = E_{\text{min}} - 1$ (i.e., $E = 0$)
  - $F \neq 0$
- The number represented is $0.f_{-1}f_{-2}\ldots f_{-23} 2^{E_{\text{min}}}$
Special quantities: Denormals

- From 0 to $2^{E_{\text{min}}}$
Summary of IEEE 754 numbers

<table>
<thead>
<tr>
<th>Single precision</th>
<th>Double precision</th>
<th>Object</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Exponent</td>
<td>Significand</td>
<td>Exponent</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>≠ 0</td>
<td>0</td>
</tr>
<tr>
<td>1 – 254</td>
<td>anything</td>
<td>1 – 2046</td>
</tr>
<tr>
<td>255</td>
<td>0</td>
<td>2047</td>
</tr>
<tr>
<td>255</td>
<td>≠ 0</td>
<td>2047</td>
</tr>
</tbody>
</table>
Floating point operations: addition

- Addition: $C = A + B$
- Step 1: Align the significands
  - Are the exponents different? Shift the smaller number to the right and adjust the exponent
- Step 2: Add the significands
  - Don’t forget the sign
- Step 3: Normalize the sum
  - Check for overflow or underflow
Floating point operations: addition

- A = 1.000 \(2^{-1}\) (base 2)
- B = –1.110 \(2^{-2}\) (base 2)

- Step 1: Align the significands
- Step 2: Add the significands
- Step 3: Normalize the sum
Floating point operations: multiplication

- \( C = A \times B \)
- Step 1: Add the exponents
- Step 2: Multiply the significands
  - Without the sign
- Step 3: Normalize the product
- Step 4: Set the sign of the product
  - Both positive? Result is positive
  - Both negative? Result is positive
  - Different signs? Result is negative
Floating point operations: multiplication

- $A = 1.000 \ 2^{-1}$ (base 2)
- $B = -1.110 \ 2^{-2}$ (base 2)

- Step 1: Add the exponents

- Step 2: Multiply the significands

- Step 3: Normalize the product

- Step 4: Set the sign of the product
Rounding

- What is rounding, and when do we need to round?

- Numbers that we can represent, and numbers that we can not represent:

- What is fair?
Rounding modes

- IEEE 754 defines four rounding modes…
- Towards +Inf
- Towards -Inf
- Towards 0
- To nearest
Rounding modes

- We need some extra bits...
- In IEEE-754, the internal representation uses at least 3 more bits in the significand
  - Guard
  - Round
  - Sticky
Rounding modes

2n-bit PRODUCT

case 1)

case 2)
Recommended exercises


- Ex. B: Give the IEEE 754 single-precision representation of the number $34000001_{10}$ (thirty-four million and one). Give its rounded representation using all four rounding modes, and for each one, give the decimal value of the number actually represented.