Floating Point Representation

CS301
Prof. Szajda
Administrative

- Lab #4 due Fri., 9/27, at 1:30pm
- HW #4 due Wednesday at 5pm
- Read Chapter 3.1–3.2, 3.5
Floating Point Representation
Floating Point

- Need to represent very small values and very large values
- Normalized numbers:
  - Number in scientific notation in which no leading 0s
  - $1.0 \times 10^{-9}$ is normalized but $10 \times 10^{-10}$ is not
- MIPS uses IEEE 754 floating point standard
  - 32b (single precision)
  - 64b (double precision)
Floating Point Representation

- Floating point
  - Binary point is not fixed
  - \(1.xxxxxx_{\text{2}} \times 2^{yyyy}\)
- \((-1)^{S} \times F \times 2^{E}\)

- S: Sign
- E: Exponent
- F: Significand
IEEE Standard assumes leading 1 bit of normalized binary numbers

- Significand, consequently, 24b of precision
- \((-1)^S \times (1+F) \times 2^E\)
  - \(F\) represents fraction between 0 and 1
- \((-1)^S \times (1 + (s1 \times 2^{-1}) + (s2 \times 2^{-2}) + (s3 \times 2^{-2}) + \ldots) \times 2^E\)

- 0 has special reserved exponent field of 0 since no leading 1
Floating Point Representation

- Compare two numbers by comparing exponents first
- If we stuck with notation where negative exponents started with leading 1 (2s complement), they’ll look like large numbers
  - $1.0 \times 2^{-1}$ exponent field? 1111 1111
  - $1.0 \times 2$ exponent field? 0000 0001
- Biased notation
  - Most negative exponent is 00..00$_2$ and most positive 11..11$_2$
  - Bias: number subtracted from normal, unsigned representation to determine real value
  - IEEE 754 uses bias of 127
    - $-1$ represented as $-1+127=126$ in binary 0111 1110
    - $+1$ represented as $1+127=128$ in binary 1000 0000
    - $(-1)^S \times (1+F) \times 2^{(\text{Exponent} - \text{Bias})}$
while (x > 0){
    int bit = (int) 2 × x;
    //append bit to bitstring
    x = 2 × x - bit;
}

What is the binary representation of \(-0.75\)?
What is the value in binary?
\(-3/4\) is equal to \(-11_2/2^2\) or \(-0.11_2\)
How about normalized?
\(-1.1_2 \times 2^{-1}\)
The general representation formula is
\((-1)^S \times (1+F) \times 2^{(\text{Exponent} - 127)}\)
What is the representation for our number?
\((-1)^1 \times (1+.1000\ 0000\ 0000\ 0000\ 0000\ 0000_2) \times 2^{(126 - 127)}\)
What is the Encoding?

- $( -1 )^1 \times (1 + .1000 \ 0000 \ 0000 \ 0000 \ 0000 \ 0000 \ 0000 \ 0002 ) \times 2^{(126 - 127)}$

\[
\begin{array}{cccccccc}
31 & 30 & 29 & 28 & 27 & 26 & 25 & 22 \\
\hline
\text{sign bit} & \text{exponent} & \text{significand} & \text{significand} & \text{significand} & \text{significand} & \text{significand} & \text{significand}
\end{array}
\]

- Sign bit
  - 1
- Significand
- Exponent
What is the Encoding?

- \((-1)^1 \times (1 + 10000000000000000000000002) \times 2^{(126 - 127)}\)
- Sign bit
  - 1
- Significand
  - 1000000000000000000000000
- Exponent
What is the Encoding?

- \((-1)^1 \times (1+0.1000 0000 0000 0000 0000 0000 0000_2) \times 2^{(126 - 127)}\)

```
31 30 22 0
  exponent  significand
```

- **Sign bit**
  - 1
- **Significand**
  - 1000 0000 0000 0000 0000 0000 000
- **Exponent**
  - 0111 1110
What is the Decimal Value?

- The general representation formula is $(-1)^S \times (1+F) \times 2^{(\text{Exponent} - 127)}$
What is the Decimal Value?

• The general representation formula is

\[ (-1)^S \times (1+F) \times 2^{(Exponent - 127)} \]

S: \(-1\)
F: \(1 \times 2^{-2}\)
E: 129
What is the Decimal Value?

- The general representation formula is 
  \[ (-1)^S \times (1+F) \times 2^{(Exponent - 127)} \]

<table>
<thead>
<tr>
<th>31</th>
<th>30</th>
<th>22</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10000001</td>
<td>010000000000000000000000000000000</td>
<td></td>
</tr>
</tbody>
</table>

sign bit

S: \(-1\)
F: \(1 \times 2^{-2}\)
E: 129

\[ (-1)^1 \times (1+0.25) \times 2^{(129 - 127)} \]

\(-5.0\)
Exceptions

- Can represent $2.0 \times 10^{-38}$ to $2.0 \times 10^{38}$
- **Overflow**
  - Too large a number
  - Exponent is too large to be represented in exponent field
- **Underflow**
  - Too small a number
  - Negative exponent is too large to be represented in exponent field
- Note that with 64 bits, there are $2^{64}$ (approx. $1.85 \times 10^{19}$) numbers that can be represented.
  - Vast majority of numbers cannot be represented!
What is the MIPS floating point
Exercises: What is the MIPS floating point encoding for $6.375_{10}$?

- $6 = 110$
- $0.375 = .011$
  - $0.375 \times 2 = 0.75 = 0$
  - $0.75 \times 2 = 1.5 = 1$
  - $0.5 \times 2 = 1$
- $110.011 = 1.10011 \times 2^2$
- $S = 0$
- $F = 1001100000000000000000000$
- $E: 2 + 127 = 129$
  - $10000001$

HEX: $0x40CC0000$
Given 0xC1A88000, what is the decimal value if we interpret these 32b as a IEEE single-precision floating point number?
Exercises:
Given 0xC1A88000, what is the decimal value if we interpret these 32b as a IEEE single-precision floating point number?

- 1100 0001 1010 1000 1000 0000 0000 0000
- S: 1
- E: 10000011 = 128 + 2 + 1 = 131 - 127 = 4
- F: 010 1000 1000 0000 0000 0000
  - 1.01010001
- \(-1 \times 10101.0001 = 16 + 4 + 1 = -21.0625\)
Floating Point Addition (Decimal)

9.998\times10^1 + 7.771\times10^{-2}

- Assume 4 digits significand, 2 for exponent
- Algorithm:
  - Align decimal point (move smaller number)
  
    \[
    9.998 \times 10^1 \\
    0.008 \times 10^1 \text{ (rounded version of } 7.771 \times 10^{-2} = 0.007771 \times 10^1)\\
    \]
  - Add significands
  
    \[
    10.006 \times 10^1\\
    \]
  - Renormalize (check for overflow/underflow)
  
    \[
    1.0006 \times 10^2\\
    \]
  - Round if renormalizing gave more than 4 digits
  
    \[
    1.001 \times 10^2
    \]
Floating Point Addition (Cont)

- In single-precision, we have 24b for significand
  - $2^{24} - 1 = 16,777,215$
  - In decimal, we can represent 8 significant digits
- What happens when you add very small value to a very large value?
  - Smaller value’s bits truncated in normalize process
Floating Point Multiplication

\[(1.110 \times 10^{10}) \times (9.200 \times 10^{-5})\]

- Algorithm:
  - Calculate exponent by adding exponents together (subtract bias if exponents in bias form)
    \[10 + (-5) = 5\]
  - Multiply significands
    \[10.212000\]
  - Renormalize (check for over/underflow)
    \[1.0212 \times 10^6\]
  - Round
    \[1.021 \times 10^6\]
  - Determine sign
    Signs are the same, so positive