Discussion 7 Richard Guo
Floating Point 02/27/09

2. The Number Line Revisited

ints allocate bits for discrete integers along the number line. A finite number of bits (32) can only hold a finite # of elements. We need to sacrifice precision for greater range, and even then there are trade offs. The decision was made to have variable spacing that becomes bigger the closer to infinity the number is.

3. Reallocating Bits

Floating point uses a sign and magnitude model for representing numbers.

	Sign	Exponent	Significand/Mantissa	Bias
Float (single precision)	1 [31]	8 [30-23]	23 [22-00]	127
Double	1 [63]	11 [62-52]	52 [51-00]	1023

$$\textbf{Floating point \# } = -1^{(\text{sign})} \times (1.\text{significand}) \times 2^{(\text{Exponent - Bias})}$$

Don't forget about the implicit 1 and the bias offset! We use bias so that we don't have to worry about a sign bit in the exponent field for hardware comparators. The bias is calculated as $\frac{2^{MAX\ EXPONENT}}{2} - 1$ and ensures that all exponents can be treated as unsigned. Why don't we use the same method to represent integers?

Operations Table:

Operation	Result
n ÷ ±Infinity	0
±Infinity × ±Infinity	±Infinity
±nonzero ÷ 0	±Infinity
Infinity + Infinity	Infinity
±0 ÷ ±0	*NaN*
Infinity - Infinity	*NaN*
±Infinity ÷ ±Infinity	*NaN*
±Infinity × 0	*NaN*

Special Cases (for floats):

Exponent	Significand	Meaning
0	0	+/- 0
0	Non-zero	+/- Denormalized number
255 (all 1s)	0	+/- Infinity
255 (all 1s)	Non-zero	NaN (Not a Number)

Denorm range: $(+/-)\ (2^{-149} - (1-2^{-23}) \times 2^{-126})$

Norm range: $(+/-)\ (2^{-126} - (2-2^{-23}) \times 2^{127})$

NaNs usually result from nonsense arithmetic operations or represents an irrational number.

Overflow is when we try to represent a number greater than what the max exponent can hold. Underflow is trying to represent a number too small (too close to 0) for the lowest exponent.

5. Rounding

4 ways of rounding when the number to represent doesn't fit in the space allotted:

1. Round up to infinity
2. Round down towards -infinity
3. Truncate – round towards 0
4. Unbiased – round towards even (default used in FP)

6. Solved Problems

Floating Point Exercises

Convert the following decimal numbers into fixed point (not floating point):

1.5	0.25	0.8	-16.5
1.1b	0.01b	0.1100(repet)b	-10000.1b

Give the best hex representation of the following numbers using single precision floats:

1.0	-7.5	(1.0/3.0)	(186.334/0.0)
0x3f800000	0xc0f00000	0x3eaaaaaa	0x7f800000

What is the value of the following single precision floats?

0x0	0xff94beef	0x1
0.0f	NaN	2^-149

Disassembly Exercises

Be a processor! Translate the following hex instructions into MIPS:

```
0x8c880000 lw $t0, 0($a0)
0x2108ffff addi $t0, $t0, -1
0xaca80000 sw $t0, 0($a1)
0x03e00008 jr $ra
```