Floating Point (part II)

CSCI 237: Computer Organization
9th Lecture, Wednesday, Sept. 25

Kelly Shaw

1

1

Administrative Details

- Lab #2 due Tuesday at 11pm
  - What questions do you have?
- Read CSAPP 2.4-2.5
- Weekly quiz due Friday 2:30pm
- My help hours this Thursday are on Zoom
- Colloquium talk on Friday at 2:35pm in Wege
  - Sam Thomas, Brown University
  - Towards a Practical Secure Memory for Modern Deployments

2

2

Last Time: Floating Point (part I)

- Memory Abstraction
- Background: Fractional binary numbers
- IEEE FP standard (normalized and denormalized values)
- Examples

3

3

Today: Floating Point (part II)

- IEEE FP standard (normalized and denormalized values)
- Tiny Floating Point Example
- Floating point in C
- Summary

4

4

Floating Point Representation

Example:
$15213_{10} = (-1)^0 \times 1.1101101101101_2 \times 2^{13}$

- Numerical Form:

$$(-1)^s * M * 2^E$$

  - Sign bit s determines whether number is negative or positive
  - Significand (mantissa) M normally a fractional value in range [1.0,2.0).
  - Exponent E weights value by power of two

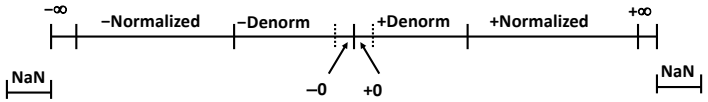
- Encoding
  - MSB is sign bit s
  - exp field *encodes* E (but is not equal to E)
  - frac field *encodes* M (but is not equal to M)

5

5

3 "Cases" in Floating Point Format

- Special values: infinity, negative infinity, and NaN
- So-called "normalized" form
- So-called "denormalized" form

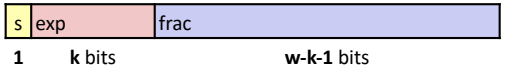


- We will go over each case individually, and revisit this number line at the end

6

6

Special Values



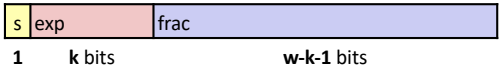
- Condition: **exp** = **111…1**
- Case: **exp** = **111…1**, **frac** = **000…0**
  - **Represents value ∞ (infinity)**
    - Operation that overflows
    - Both positive and negative
    - E.g., 1.0/0.0 = −1.0/−0.0 = +∞, 1.0/−0.0 = −∞
- Case: **exp** = **111…1**, **frac** ≠ **000…0**
  - **Not-a-Number (NaN)**
  - Represents case when no numeric value can be determined
  - E.g., sqrt(–1), ∞ − ∞, ∞ × 0

7

7

"Normalized" Values

$v = (-1)^s M 2^E$



- Condition: exp ≠ 000…0 and exp ≠ 111…1
- Exponent coded as a biased value: E = *exp* – bias
  - *exp*: unsigned value of *exp* field
  - *bias* = $2^{k-1} - 1$, where k is number of exponent bits
    - Single precision: 127 (exp: 1…254, E: -126…127)
    - Double precision: 1023 (exp: 1…2046, E: -1022…1023)
- Significand coded with implied leading 1: M = $1.xxx…x_2$
  - xxx…x: bits of frac field
  - Minimum when frac=000…0 (M = 1.0)
  - Maximum when frac=111…1 (M = 2.0 – ε)
  - Get extra leading bit for "free" with implicit leading 1

8

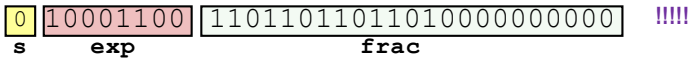
8

Normalized Encoding Example

$v = (-1)^s M 2^E$
$E = exp - bias$

- Value: `float F = 15213.0;`
  - $15213_{10} = 11101101101101_2$
    $= (-1)^0 \times 1.1101101101101_2 \times 2^{13}$
- Sign: S
  S = 0
- Significand: M
  M = 1.1101101101101_2
  frac = $11011011011010000000000_2$
- Exponent: E
  E = 13
  bias = $2^{k-1}-1 = 2^{8-1}-1 = 127$
  exp = $13 + 127 = 140 = 10001100_2$

E = *exp* – bias
E = *exp* – 127
exp = E + 127



!!!!!

9

9

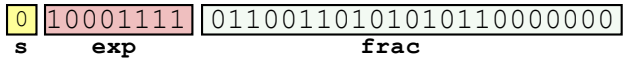
Normalized Decoding Example

$v = (-1)^s M 2^E$
$E = exp - bias$

- Sign: S
  S = 0
- Significand: M
  frac = $01100110101010110000000_2$
  M = $1.01100110101010110000000_2$
- Exponent: E
  bias = $2^{k-1}-1 = 2^{8-1}-1 = 127$
  exp = $10001111_2 = 143$
  E = 143 – 127 = 16

E = *exp* – bias

$(-1)^0\ 1.0110011010101011\ 2^{16} = 91819$



10

10

Denormalized Values

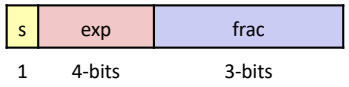
$v = (-1)^s M 2^E$
$E = 1 - bias$

- Condition: **exp = 000…0**
- Exponent value: E = **1 – bias**
- Significand coded with implied **leading 0**: M = 0.xxx…x_2
  - **xxx…x**: bits of **frac**
- Cases
  - **exp** = 000…0, **frac** = 000…0
    - Represents zero value
    - Note distinct values: +0 and –0
  - **exp** = 000…0, **frac** ≠ 000…0
    - Numbers closest to 0.0
    - Equispaced

11

11

Tiny Floating Point Example



- 8-bit Floating Point Representation
  - the sign bit is in the most significant bit
  - the next four bits are the exponent, with a bias of 7 (since $2^{4-1} - 1 = 7$)
  - the last three bits are the **frac**
- Same general form as IEEE Format
  - normalized, denormalized
  - representation of 0, NaN, infinity

12

12

Dynamic Range (Positive Only)

$v = (-1)^s M 2^E$
n: E = exp – bias
d: E = 1 – bias
(bias = 7)

	s	exp	frac	E	Value	
Denormalized numbers	0	0000	000	-6	0	
	0	0000	001	-6	1/8*1/64 = 1/512	closest to zero
	0	0000	010	-6	2/8*1/64 = 2/512	
	…					
	0	0000	110	-6	6/8*1/64 = 6/512	
	0	0000	111	-6	7/8*1/64 = 7/512	largest denorm
	0	0001	000	-6	8/8*1/64 = 8/512	smallest norm
	0	0001	001	-6	9/8*1/64 = 9/512	
	…					
	0	0110	110	-1	14/8*1/2 = 14/16	
	0	0110	111	-1	15/8*1/2 = 15/16	closest to 1 below
Normalized numbers	0	0111	000	0	8/8*1 = 1	
	0	0111	001	0	9/8*1 = 9/8	closest to 1 above
	0	0111	010	0	10/8*1 = 10/8	
	…					
	0	1110	110	7	14/8*128 = 224	
	0	1110	111	7	15/8*128 = 240	largest norm
	0	1111	000	n/a	inf	

13