

# Floating Point

There are 10 types of people in the world:  
Those who know binary,  
and those who don't!



..  
..

# Today: Floating Point

- Background: Fractional binary numbers
- IEEE floating point standard: Definition
- Example and properties
- Rounding, addition, multiplication
- Floating point in C
- Summary

Floating are not Real  
fixed  
with  
exponent  
mantissa

# Fractional binary numbers

- What is  $1011.101_2$ ?

$$1011.101_2$$

Annotations for the fractional part:

- 1 →  $2^{-1}$
- 0 →  $2^{-2}$
- 1 →  $2^{-3}$
- 1 →  $2^{-4}$

Annotations for the integer part:

- 1 →  $2^0$
- 0 →  $2^1$
- 1 →  $2^2$
- 1 →  $2^3$

# Fractional Binary Numbers

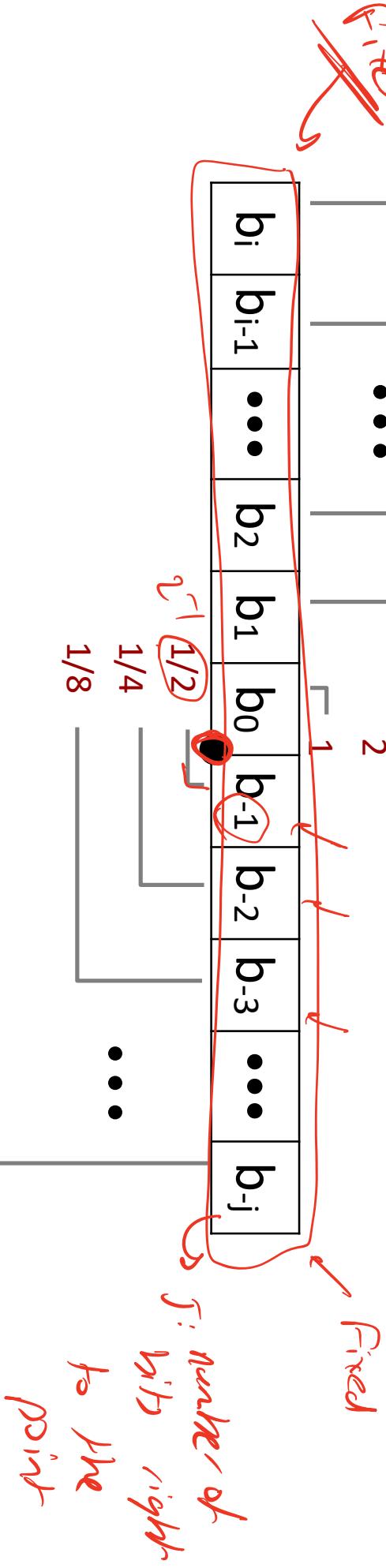
 $2^i$  $2^{i-1}$  $4$  $2$  $1$  $b_{-1}$  $b_0$  $b_{-2}$  $b_{-3}$  $\dots$  $b_1$  $b_2$  $\dots$  $b_{i-1}$  $2^{-j}$ 

## ■ Representation

- Bits to right of “binary point” represent fractional powers of 2

- Represents rational number:

$$\sum_{k=-j}^i b_k \times 2^k$$



# Fractional Binary Numbers: Examples

- Value

5.3/4

2 7/8

1 7/16

- Representation

101.11<sub>2</sub>

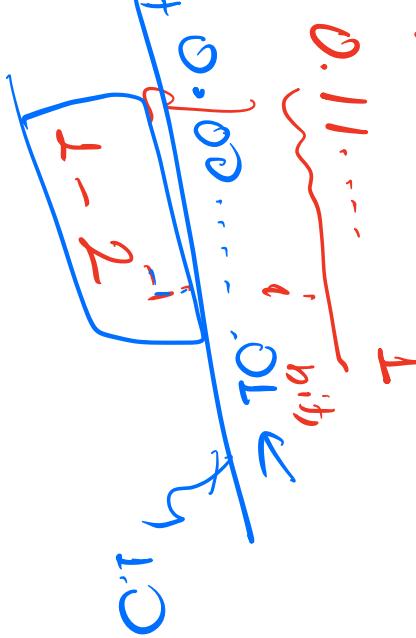
10.111<sub>2</sub>

1.0111<sub>2</sub>

- Observations

- Divide by 2 by shifting right (unsigned)
- Multiply by 2 by shifting left *i* bits
- Numbers of form 0.111111<sub>2</sub> are just below 1.0
  - 1/2 + 1/4 + 1/8 + ... + 1/2<sup>i</sup> + ... → 1.0
  - Use notation 1.0 - ε

*i* bit representation



# Representable Numbers

$$\frac{1}{3} = 0.\overline{33\ldots}$$

Avogadro's number!

$$6.02 \cdot 10^{23}$$

- Limitation #1
  - Can only exactly represent numbers of the form  $x/2^k$
  - Other rational numbers have repeating bit representations

## Value      Representation

■ $\frac{1}{3}$	<u>0.0101010101[01]...<sub>2</sub></u>
■ $\frac{1}{5}$	<u>0.001100110011[0011]...<sub>2</sub></u>
■ $\frac{1}{10}$	<u>0.0001100110011[0011]...<sub>2</sub></u>

$$\underbrace{100000}_{w} \dots \underbrace{01}_{0}$$

$$1.27 \cdot 10^{-m}$$

Largest number

$$2^{w-1}$$

Smallest number  
 $2^{-(w-1)}$

- Limitation #2
  - Just one setting of binary point within the w bits
  - Limited range of numbers (very small values? very large?)

# Today: Floating Point

- Background: Fractional binary numbers
- IEEE floating point standard: Definition
- Example and properties
- Rounding, addition, multiplication
- Floating point in C
- Summary

# IEEE Floating Point

- IEEE Standard 754
- Established in 1985 as uniform standard for floating point arithmetic
  - Before that, many idiosyncratic formats
  - Supported by all major CPUs
- Driven by numerical concerns
- Nice standards for rounding, overflow, underflow
- Hard to make fast in hardware
  - Numerical analysts predominated over hardware designers in defining standard

# Floating Point Representation

- Numerical Form:

$$(-1)^{\text{sig}} \cdot M \cdot 2^E$$

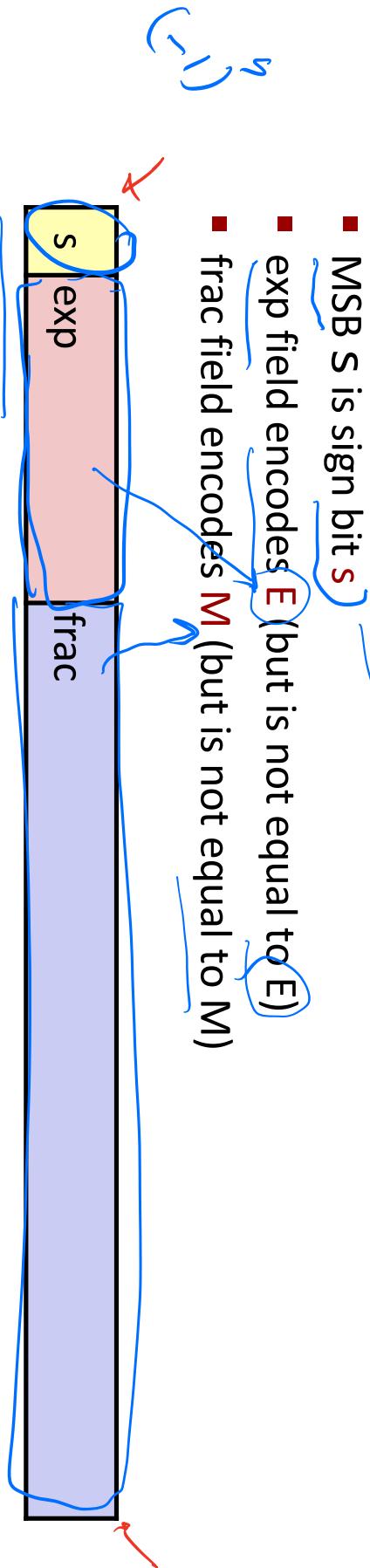
- Sign bit  $s$  determines whether number is negative or positive

- Significand  $M$  normally a fractional value in range [1.0, 2.0).

- Exponent  $E$  weights value by power of two

$$\begin{array}{ll} s=0 \Rightarrow + & (-1)^0 \\ s=1 \Rightarrow - & (-1)^1 \end{array}$$

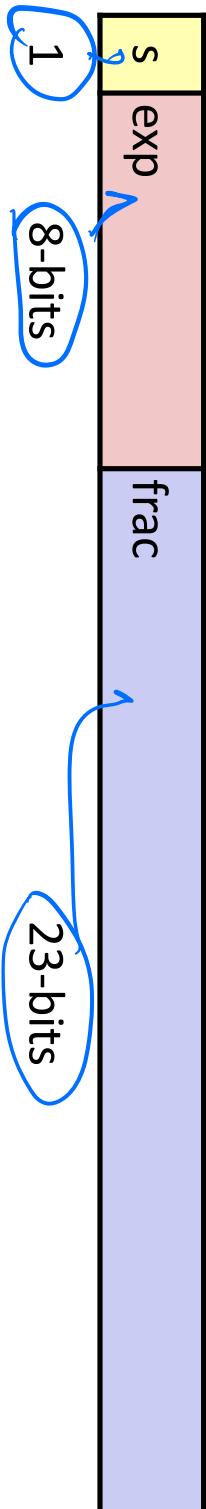
$$6.02 \cdot 10^{-23}$$



# Precision options

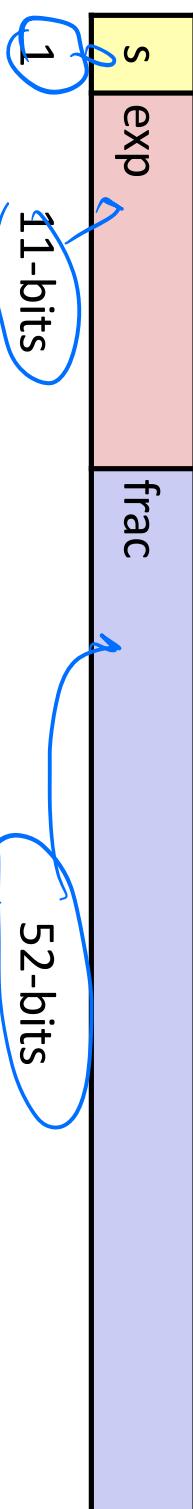
- Single precision: 32 bits

`float = 4 bytes`

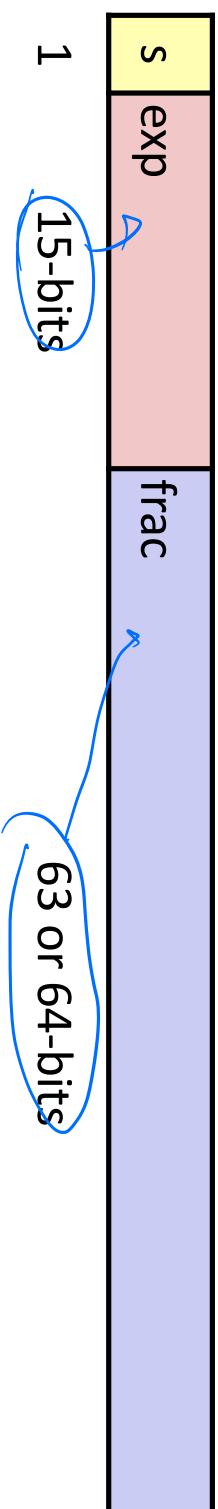


- Double precision: 64 bits

`double = 8 bytes`



- Extended precision: 80 bits (Intel only)



Normalized		Denormalized	Special Values
$s   \exp   \frac{m}{2^{e-1}}$	$0 - \frac{1}{2^{e-1}} + 2^{e-1}$	$\exp = 0 \dots 0 \quad \text{or} \quad \exp = 1 \dots 1$	$\exp = 0 \dots 0 \quad \text{or} \quad \exp = 1 \dots 1$
$(-1)^s \cdot M \cdot 2^E$	$(-1)^s \cdot M \cdot 2^E$	$E = \exp - \text{Bias}$	$E = 1 - \text{Bias}$
$M = 1 \cdot \text{frac}$	$M = 0 \cdot \text{frac}$	$M = 0 \cdot \text{frac}$	$\exp = 0 \dots 0$ if $\text{frac} = 0 \dots 0$ $\Rightarrow \pm\infty$ $\nwarrow (-1)^s$

Con represent 2<sup>32</sup>0

e.g.  $\sqrt{-1}$   
 $\infty - \infty$   
 $\infty * 0$

$\geq NaN$   
(Not-a-Number)

## “Normalized” Values

- When:  $\exp \neq 000\ldots 0$  and  $\exp \neq 111\ldots 1$

- Exponent coded as a biased value:  $E = \text{Exp} - \text{Bias}$

- $\text{Exp}$ : unsigned value of exp field

- Bias =  $2^{k-1} - 1$ , where  $k$  is number of exponent bits

*e*

- Significand coded with implied leading 1:  $M = 1.\text{xxx}\ldots x_2$ 
  - $\text{xxx}\ldots x$ : bits of frac field

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

# “Normalized” Values

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

- When:  $\exp \neq 000\ldots 0$  and  $\exp \neq 111\ldots 1$

$v \approx 1$

$v \approx 1$

- When:  $\exp \neq 000\ldots 0$  and  $\exp \neq 111\ldots 1$

- Exponent coded as a biased value:  $E = \text{Exp} - \text{Bias}$

- Exp: unsigned value of exp field

- Bias =  $2^{k-1} - 1$ , where  $k$  is number of exponent bits

- Single precision: 127 ( $\text{Exp}: 1\ldots 254, E: -126\ldots 127$ )
- Double precision: 1023 ( $\text{Exp}: 1\ldots 2046, E: -1022\ldots 1023$ )

$$\begin{aligned} k &= 8 \text{ bits} \\ \text{Bias} &= 2^7 - 1 = 127 \end{aligned}$$

$$\text{frac}$$

- Significand coded with implied leading 1:  $M = 1.\text{xxx}\ldots x_2$

- xxx...x: bits of frac field

- Minimum when  $\text{frac}=000\ldots 0$  ( $M = 1.0$ )

- Maximum when  $\text{frac}=111\ldots 1$  ( $M = 2.0 - \epsilon$ )

- Get extra leading bit for “free”

$$\epsilon = 2^{-p}$$

$$p \text{ bits}$$

# Normalized Encoding Example

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

$$E = \text{Exp} - \text{Bias}$$

- Value: float F = 15213.0;

$$15213_{10} = 11101101101101_2$$

$$= 1.1101101101101_2 \times 2^{13}$$

- Significand

$$M = 0001.1101101101100000000000000000_2$$

$$11011011011010000000000000000000_2$$

$$\frac{1.1101100}{2} = 1.1101100$$

$$\text{Bias} = 2^{12} - 1 = 4095$$

$$E = 140 - 127 = 13$$

- Exponent

$$E = 13$$

$$\text{Bias} = 127$$

$$\text{Exp} = 140 = 10001100_2$$

- Result:

**S**      **exp ~ 8 bits**

**0 10001100 110110110110100000000000**

**s**

Bryant and O'Hallaron, Computer Systems: A Programmer's Perspective, Third Edition

**frac**  
23 bits

# Denormalized Values

$$v = (-1)^s M \cdot 2^E$$
$$E = 1 - \text{Bias}$$

- Condition:  $\exp = 000\ldots0$

- Exponent value:  $E = 1 - \text{Bias}$  (instead of  $E = 0 - \text{Bias}$ )
- Significand coded with implied leading 0:  $M = 0.x_{15}\ldots x_2$

- $xxxx.x$ : bits of `frac`

# Denormalized Values

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

$$E = 1 - \text{Bias}$$

- Condition:  $\exp = 000\ldots0$

- Exponent value:  $E = 1 - \text{Bias}$  (instead of  $E = 0 - \text{Bias}$ )

- Significand coded with implied leading 0:  $M = 0.\text{xxx}\ldots x_2$

- $\text{xxx...x}$ : bits of `frac`

- Cases

- $\exp = 000\ldots0$ , `frac` = 000...0

- Represents zero value

- Note distinct values: +0 and -0 (why?)

- $\exp = 000\ldots0$ , `frac`  $\neq 000\ldots0$

- Numbers closest to 0.0

- Equispaced

$$v = (-1)^s \underbrace{M}_{1} \cdot 2^{\underbrace{E}_{-1}}$$

# Special Values

- Condition: **exp = 111...1**

- Case: **exp = 111...1, frac = 000...0**

- Represents value  $\infty$  (infinity)

- Operation that overflows

- Both positive and negative

- E.g.,  $1.0/0.0 = +\infty$ ,  $1.0/-0.0 = -\infty$

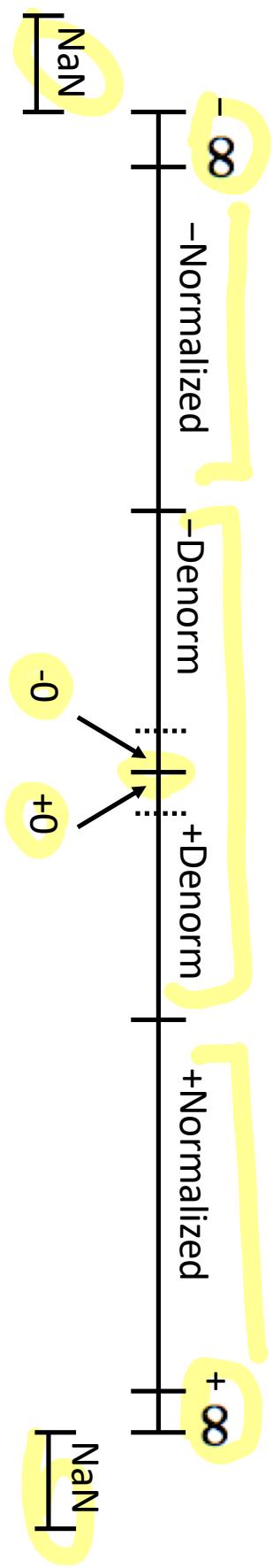
- Case: **exp = 111...1, frac ≠ 000...0** – *Multiple bit representations*

- Not-a-Number (NaN)

- Represents case when no numeric value can be determined

- E.g.,  $\sqrt{-1}$ ,  $\infty - \infty$ ,  $\infty \times 0$

# Visualization: Floating Point Encodings

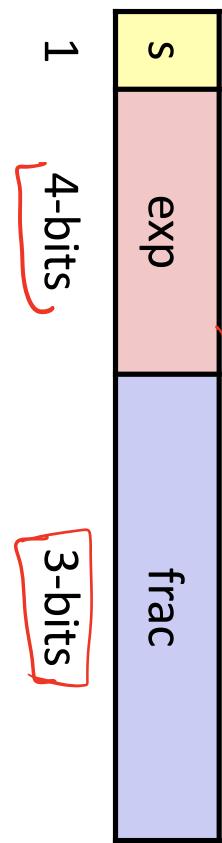


# Today: Floating Point

- Background: Fractional binary numbers
- IEEE floating point standard: Definition
- Example and properties
- Rounding, addition, multiplication
- Floating point in C
- Summary

# Tiny Floating Point Example

$$\begin{aligned} \text{exp} &= 0 - 15 \\ \text{Bias} &= 2^{4-1} - 1 = 7 \end{aligned}$$



## 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
- the last three bits are the **frac**

- Same general form as IEEE Format
  - normalized, denormalized
  - representation of 0, NaN, infinity

# Dynamic Range (Positive Only)

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

$n: E = \text{Exp} - \text{Bias}$

s	exp	frac	E	Value
0	0000	000	-6	0
0	0000	001	-6	$1/8*1/64 = 1/512$
0	0000	010	-6	$2/8*1/64 = 2/512$

Denormalized numbers

...				
0	0000	110	-6	$6/8*1/64 = 6/512$
0	0000	111	-6	$7/8*1/64 = 7/512$
0	0001	000	-6	$8/8*1/64 = 8/512$
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

smallest norm

Normalized numbers

...				
0	0110	110	7	$14/8*128 = 224$
0	1110	111	7	$15/8*128 = 240$
0	1111	000	n/a	inf

largest norm

# $E = 1 - \text{Bias}$

## Dynamic Range (Positive Only)

$M = 0, \text{frac}$

Denormalized numbers

s	exp	frac	$\frac{E}{\text{Bias}}$	Value
0	0000	000	-6	$0 = (-1)^0 \cdot 0.000 \cdot 2^{-6}$
0	0000	001	-6	$1/8 * 1/64 = 1/512$
0	0000	010	-6	$2/8 * 1/64 = 2/512$

s	exp	frac	$\frac{E}{\text{Bias}}$	Value
0	0001	000	-6	$8/8 * 1/64 = 8/512$
0	0001	001	-6	$9/8 * 1/64 = 9/512$
...				
0	0110	110	-1	$14/8 * 1/2 = 14/16$
0	0110	111	-1	$15/8 * 1/2 = 15/16$
Normalizing numbers	00111	000	0	$8/8 * 1 = 1$
00111	001	0	$9/8 * 1 = 9/8$	closest to 1 above
00111	010	0	$10/8 * 1 = 10/8$	closest to 1 below
...				
0	1110	110	7	$14/8 * 128 = 224$
0	1110	111	7	$15/8 * 128 = 240$
0	1111	000	n/a	inf

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

$$n: E = \text{Exp} - \text{Bias}$$

closest to zero

$$\text{Bias} = 7$$

$$(-1)^0 \cdot 0.001 \cdot 2^{-6} = 2^{-9}$$

smallest norm

# Dynamic Range (Positive Only)

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

$$n: E = \text{Exp} - \text{Bias}$$

$$d: E = 1 - \text{Bias}$$

closest to zero

## Denormalized numbers

	s	exp	frac	E	Value
etd + 0.0	0	0000	110	-6	$6/8*1/64 = 6/512$
	0	0000	111	-6	$7/8*1/64 = 7/512$
	0	0001	000	-6	$8/8*1/64 = 8/512$
	0	0001	001	-6	$9/8*1/64 = 9/512$
...					
0	0110	110	-1		$14/8*1/2 = 14/16$
0	0110	111	-1		$15/8*1/2 = 15/16$
Normalized numbers	0	0111	000	0	$8/8*1 = 1$
	0	0111	001	0	$9/8*1 = 9/8$
	0	0111	010	0	$10/8*1 = 10/8$
...					
0	1110	110	7		$14/8*128 = 224$
0	1110	111	7		$15/8*128 = 240$
0	1111	000	n/a	inf	largest norm

# Dynamic Range (Positive Only)

$s \exp \text{frac} \quad E \quad \text{Value}$

$$\begin{matrix} x \\ y \\ b \end{matrix} = \text{Exp frac}$$

$$\begin{matrix} E = 1 - \gamma = -6 \\ M = 1.000 \\ V = (-1)^0 \cdot 1.000 \cdot 2^{-6} \end{matrix}$$

$\downarrow$

$$\begin{matrix} 0 & 0001 & 000 \\ 0 & 0001 & 001 \end{matrix}$$

$$\begin{matrix} -6 & 8/8*1/64 = 8/512 \\ -6 & 9/8*1/64 = 9/512 \end{matrix}$$

largest denorm

smallest norm

$$V = (-1)^0 \cdot \underbrace{1.001}_{9/8} \cdot 2^{-6}$$

$$\begin{matrix} \text{extreme} \\ \text{bias} \end{matrix} \quad \begin{matrix} 0 & 0110 & 110 \\ 0 & 0110 & 111 \end{matrix} \quad \begin{matrix} -1 & 14/8*1/2 = 14/16 \\ -1 & 15/8*1/2 = 15/16 \end{matrix} \quad \begin{matrix} \text{closest to 1 below} \\ \gamma - \gamma \end{matrix}$$

$$\begin{matrix} \text{Normal} \\ \text{numbers} \end{matrix} \quad \begin{matrix} 0 & 0111 & 000 \\ 0 & 0111 & 001 \end{matrix} \quad \begin{matrix} 0 & 8/8*1 = 1 \\ 0 & 9/8*1 = 9/8 \end{matrix} \quad \begin{matrix} \text{closest to 1 above} \\ \gamma + \gamma \end{matrix}$$

$$\begin{matrix} 0 & 0111 & 010 \\ \dots & & \end{matrix} \quad \begin{matrix} 0 & 10/8*1 = 10/8 \\ \dots & \end{matrix}$$

$$\begin{matrix} 0 & 1110 & 110 \\ 0 & 1110 & 111 \end{matrix} \quad \begin{matrix} 7 & 14/8*128 = 224 \\ 7 & 15/8*128 = 240 \end{matrix}$$

largest norm

$v = (-1)^s M \cdot 2^E$
$n: E = \text{Exp} - \text{Bias}$
$d: E = 1 - \text{Bias}$

# Dynamic Range (Positive Only)

s exp frac E Value

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

n: E = Exp - Bias

d: E = 1 - Bias

$$e = e + \text{Bias}$$

$$m = 1 \cdot \text{frac}$$

closest to 1 below

	0 0001 000	-6	$8/8*1/64 = 8/512$	
	0 0001 001	-6	$9/8*1/64 = 9/512$	
	...			
	0 0110 110	-1	$14/8*1/2 = 14/16$	
	0 0110 111	-1	$15/8*1/2 = 15/16$	
Normalized numbers	0 0111 000	0	$8/8*1 = 1$	
	0 0111 001	0	$9/8*1 = 9/8$	
	0 1111 010	0	$10/8*1 = 10/8$	
	...			
Normal	0 1110 110	7	$\underline{14}/\underline{8} * \underline{128} = \underline{224}$	$\sqrt[14]{(-1)^7 \cdot 1.110 \cdot 2^{14-2}}$
	0 1110 111	7	$15/8 * 128 = 240$	
	0 1111 000	n/a	inf	

Bryant and O'Hallaron, Computer Systems: A Programmer's Perspective, Third Edition

# Dynamic Range (Positive Only)

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

$$n: E = \text{Exp} - \text{Bias}$$

d:  $E = 1 - \text{Bias}$

s	exp	frac	E	Value
0	0000	000	-6	0
0	0000	001	-6	$1/8*1/64 = 1/512$
0	0000	010	-6	$2/8*1/64 = 2/512$

Denormalized numbers

...				
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	$8/8*1/64 = 8/512$
0	0001	001	-6	$9/8*1/64 = 9/512$

smallest norm

...

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	$9/8*1 = 9/8$
	0	0111	010	0	$10/8*1 = 10/8$

closest to 1 above

...				
0	1110	110	7	$14/8*128 = 224$
0	1110	111	7	$15/8*128 = 240$

largest norm

0	1111	000	n/a	inf
---	------	-----	-----	-----

END

The  
of this

lecture

on Thursday

See you  
Quiz