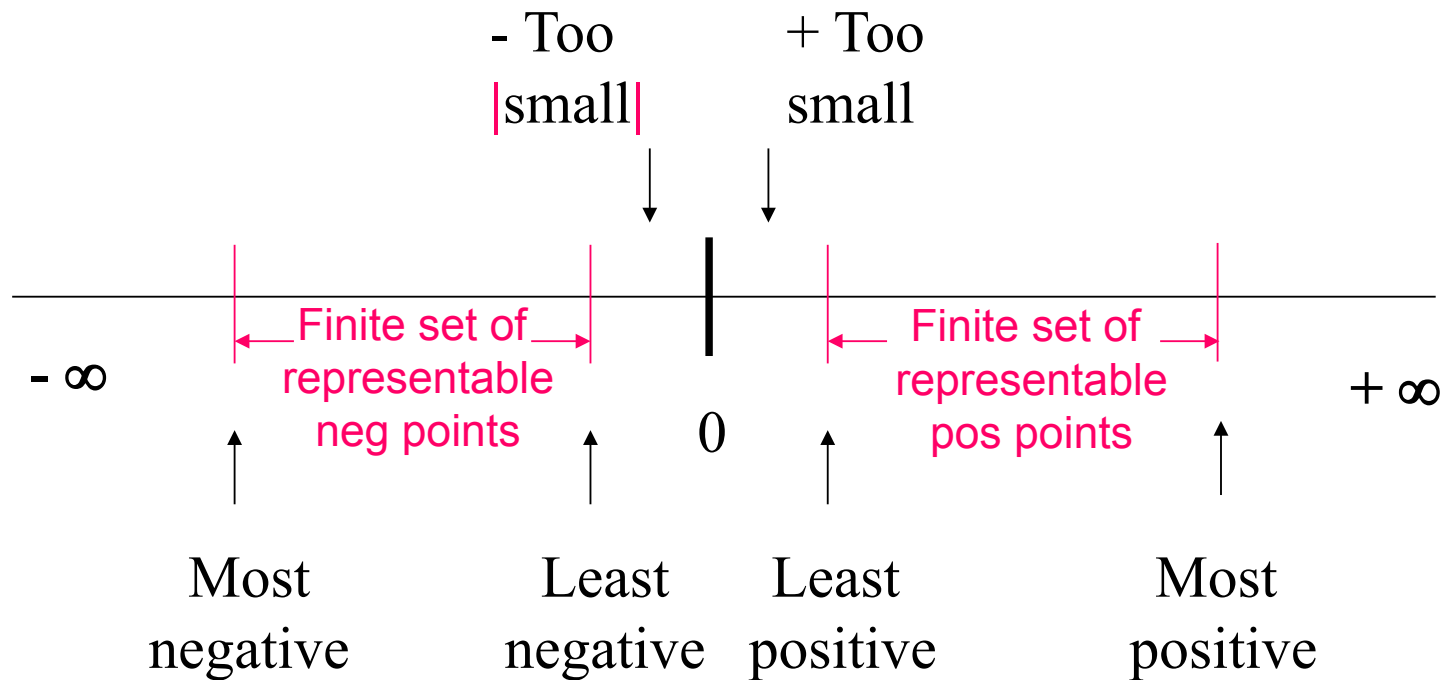
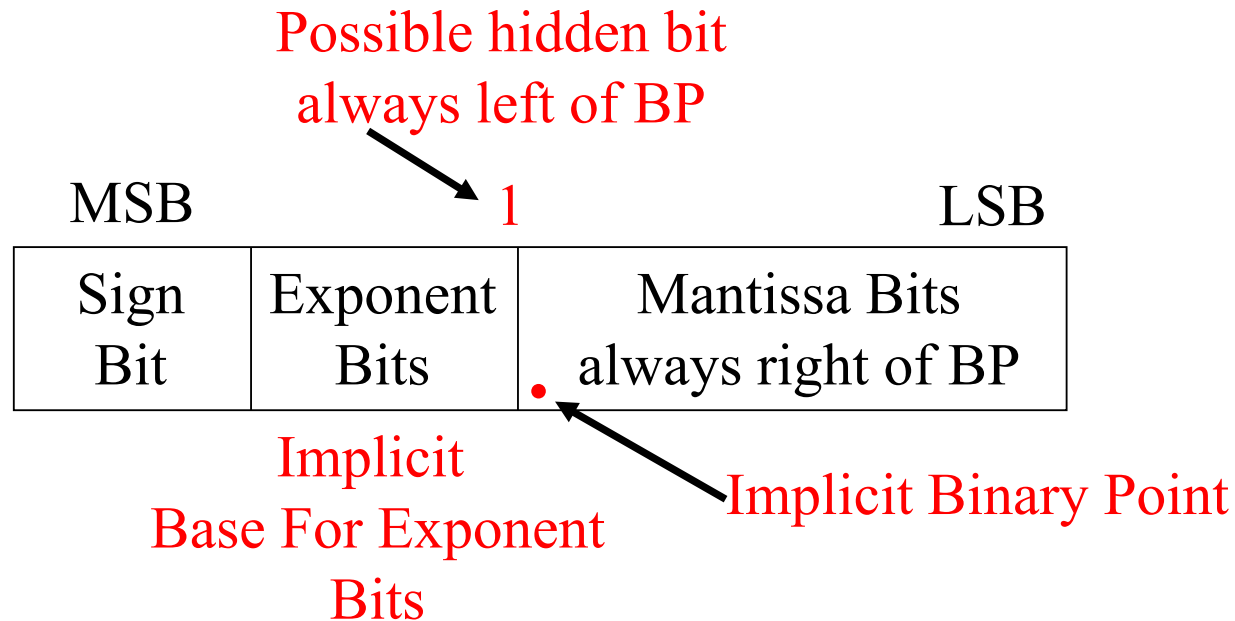


Floating Point Representation



General Floating Format



Maybe 16 so 16 exponent bits * mantissa

Maybe 2 so 2 exponent bits * mantissa

IBM Format

No hidden bit in IBM Format

Sign	Exponent of base 16 excess 64 notation	Mantissa, all right of BP $0 \leq \text{mant} < 1$
------	---	---

Implicit Binary Point

0	100 0001	1100	...	0000
↑	↑	↑		
Positive sign bit	Excess 64, here is 65 so 16^{+1}	Mantissa has $1/2 + 1/4$ so .75		

So value is $16^{+1} * .75 = 12.0$

IBM Format

0 100 0101 → 69 0110 ... 0000

$$16^{+5} \quad * \quad .375 \quad = \quad + \quad 393216.0$$

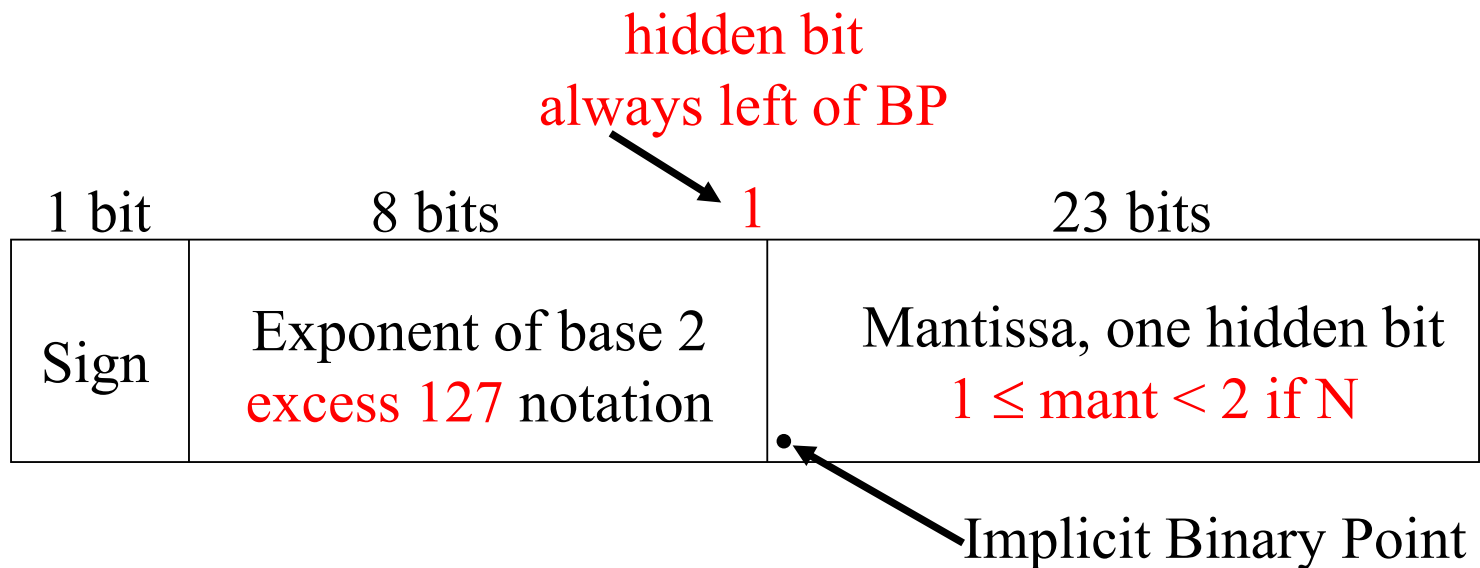
0 011 1111 → 63 1010 ... 0000

$$16^{-1} \quad * \quad .625 \quad = \quad + \quad .0390 \dots$$

1 011 1101 → 61 1100 ... 0000

$$16^{-3} \quad * \quad .75 \quad = \quad - \quad .000183 \dots$$

IEEE 754 Format



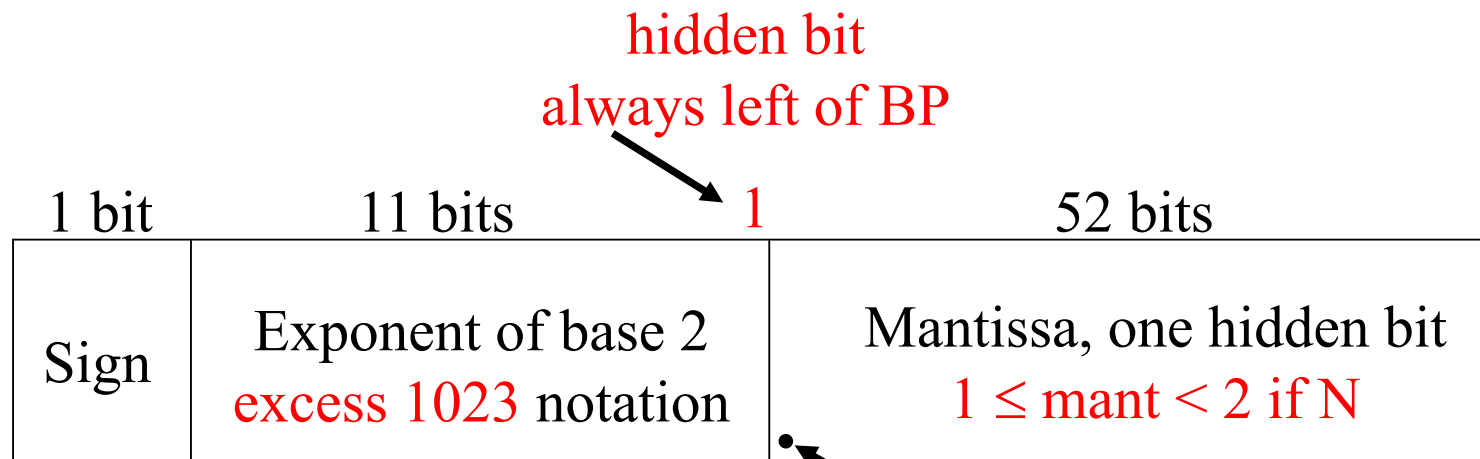
0 1000 0001 1100 ... 0000
 ↑ ↑ ↑

Excess 127, here
is 129 so 2^{+2}

Mantissa is $1(\text{HB}) + 1/2 + 1/4$
so 1.75

So value is $2^{+2} * 1.75 = 7.0$

IEEE 754 Format, Double Precision



0
1000 0000 001
1100
...
0000

↑
↑
↑

Excess 1023 here
Mantissa is 1(HB) + 1/2 + 1/4

is 1025 so 2^{+2}
so 1.75

So value is $2^{+2} * 1.75 = 7.0$

IEEE 754 Format

0 1000 0101 → 133 0110 ... 0000

$$2^{+6} \quad * \quad 1.375 \quad = \quad + 88.0$$

0 0111 1110 → 126 1010 ... 0000

$$2^{-1} \quad * \quad 1.625 \quad = \quad + .8125$$

1 0111 1100 → 124 1100 ... 0000

$$2^{-3} \quad * \quad 1.75 \quad = \quad - .21875$$

Floating Point Possibilities, IEEE 754

Least positive N	0 0000 0001 0000 ... 0000 = 2^{-126} , $\cong 10^{-38}$
Most positive	0 1111 1110 1111 ... 1111 $\cong 2^{+128}$, $\cong 10^{+38}$
Least negative N	1 0000 0001 0000 ... 0000 = -2^{-126} , $\cong -10^{-38}$
Most negative	1 1111 1110 1111 ... 1111 $\cong -2^{+128}$, -10^{+38}
Least pos/neg DN	$1/0$ 0000 0000 0000 ... 0001 = $\pm 2^{-149}$, $\cong 10^{-45}$
Zero	0 0000 0000 0000 ... 0000
Pos/neg infinity	$1/0$ 1111 1111 0000 ... 0000 = $\pm \infty$
NAN	$1/0$ 1111 1111 Any non-zero pattern = NAN

Floating Point Possibilities, IEEE 754 (cont'd)

$$\begin{array}{rclcl}
 0 & 1000\ 0101 & & .0110 & \dots\ 0000 \\
 & 2^{+6} & * & 1.375 & = +\ 88.0
 \end{array}$$

$$\begin{array}{rclcl}
 0 & 0000\ 0001 & & .0000 & \dots\ 0000 \\
 & 2^{-126} & * & 1.0 & \cong +\ 1.175 * 10^{-38}
 \end{array}$$

$$\begin{array}{rclcl}
 0 & 0000\ 0000 & & .0000 & \dots\ 0000 \\
 & & & & = +\ 0.0
 \end{array}$$

$$\begin{array}{rclcl}
 1 & 1111\ 1111 & & .0110 & \dots\ 0000 \\
 & & & & = \text{NAN}
 \end{array}$$

$$\begin{array}{rclcl}
 1 & 1111\ 1111 & & .0000 & \dots\ 0000 \\
 & & & & = -\ \infty
 \end{array}$$

Un-normalized small

$$\begin{array}{rclcl}
 0 & 0000\ 0000 & & .00001 & \dots\ 0000 \\
 & 2^{-126} & * & 2^{-5} & = +\ 2^{-131} \cong +\ 3.674 * 10^{-40}
 \end{array}$$

please enter a floating point number and new-line: 234765.579335

mantissa:	0x654365	or:	110 0101 0100 0011 0110 0101
exponent:	0x90	or:	1001 0000
sign:	0x0	or:	0
in base 10:	234765.6	or:	0 1001 0000 110 0101 0100 0011 0110 0101

please enter a floating point number and new-line: 234765.579

mantissa:	0x654365	or:	110 0101 0100 0011 0110 0101
exponent:	0x90	or:	1001 0000
sign:	0x0	or:	0
in base 10:	234765.6	or:	0 1001 0000 110 0101 0100 0011 0110 0101

please enter a floating point number and new-line: 234765.571

the floating value for INPUT_NUMBER is broken out as:

mantissa:	0x654365	or:	110 0101 0100 0011 0110 0101
exponent:	0x90	or:	1001 0000
sign:	0x0	or:	0
in base 10:	234765.6	or:	0 1001 0000 110 0101 0100 0011 0110 0101

please enter a floating point number and new-line: 234765.57

mantissa:	0x654364	or:	110 0101 0100 0011 0110 <u>0100</u>
exponent:	0x90	or:	1001 0000
sign:	0x0	or:	0
in base 10:	234765.6	or:	0 1001 0000 110 0101 0100 0011 0110 0100

```
-bash-4.1$ ./subtract
```

```
please enter a floating point number and new-line: 234765.579335  
please enter a floating point number and new-line: 234765.571  
the difference between the two numbers entered is: 0.000000
```

```
-bash-4.1$ ./subtract
```

```
please enter a floating point number and new-line: 234765.579335  
please enter a floating point number and new-line: 234765.57  
the difference between the two numbers entered is: 0.015625
```

Adding 754 FP Numbers

0	1000 0101		1 HB	.0110	...	0000
	2^{+6}	*		1.375	=	+ 88.0
0	0111 1110		+ 1 HB	.1010	...	0000
	2^{-1}	*		1.625	=	+ .8125

0 0111 1110 1 HB .1010 ... 0000

Shift mantissa of smaller number right 7 places, note hidden bit

	0	1000 0101		.0000 0011 0100	...	0000
			1 HB			
+	0	1000 0101		.0110	...	0000
<hr/>						
	0	1000 0101		1 HB .0110 0011 0100	...	0000

Result is normalized as is:

$$2^{+6} * 1 + 1/4 + 1/8 + 1/128 + 1/256 + 1/1024 = + 88.8125$$

Adding 754 FP Numbers (cont'd)

0	1000 0101		1 HB	.1110	...	0000
	2^{+6}	*		1.875	=	+ 120.0
0	1000 0010		1 HB	.1011	...	0000
	2^{+3}	*		1.6875	=	+ 13.5

Shift mantissa of smaller number right 3 places, **note hidden bit**

0	1000 0101		1 HB	.0011 0110	...	0000	000
+	0	1000 0101		.1110	...	0000	
<hr/>							
0	1000 0101		10 HB + Carry Out	.0001 0110	...	0000	
Normalize result			1 HB	.0000 10110	...	0000	0
0	1000 0110						
	2^{+7}	*		1.04296875	=	+ 133.5	

Adding 754 FP Numbers (cont'd)

0	1000 0101		1 HB	.1110	...	0000
	2^{+6}	*		1.875	=	+ 120.0
			+	1 HB		
0	1000 0101			.1011	...	0000
	2^{+6}	*		1.6875	=	+ 108.0

	0	1000 0101		1 HB	.1011	...	0000
+	0	1000 0101		1 HB	.1110	...	0000
<hr/>							
	0	1000 0101		11 HB + Carry Out	.1001	...	0000

Normalize result 

0	1000 0110		1 HB	.1100 1000	...	0000
	2^{+7}	↑	*	1.78125	=	+ 228