COMP 122/L Lecture 3

Kyle Dewey

Outline

- Operations on binary values
 - Addition
 - Subtraction
- Floating point introduction

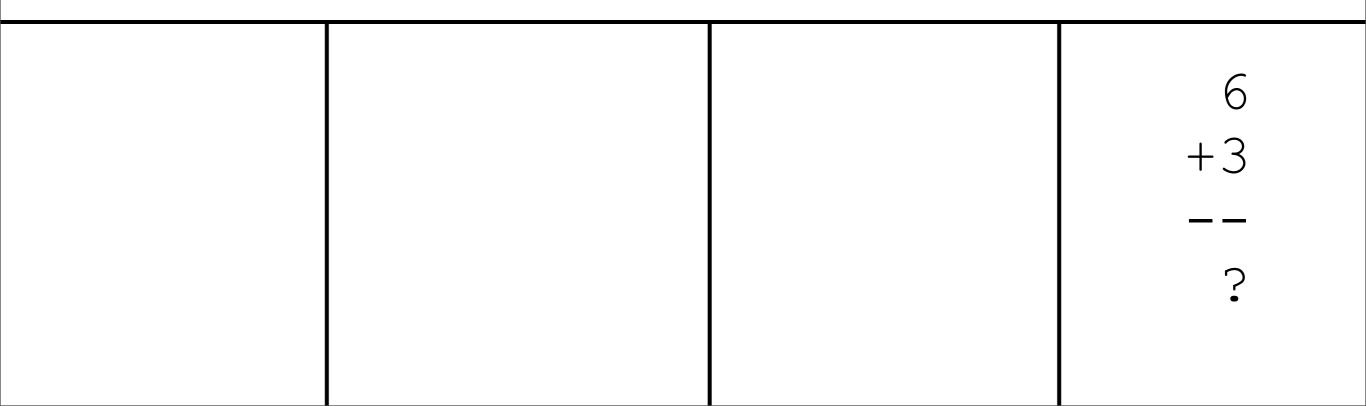
Addition

• Question: how might we add the following, in decimal?

986 +123 ----?

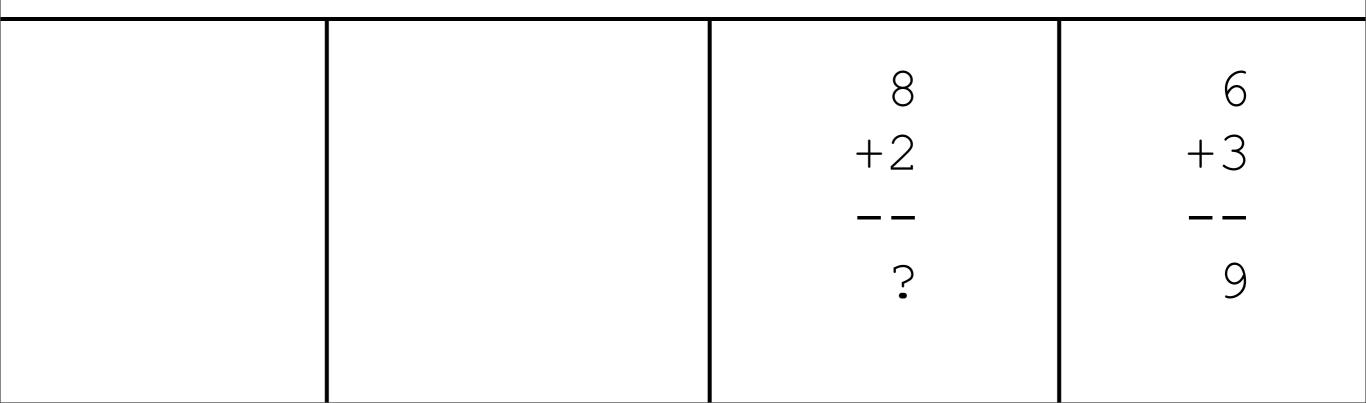
• Question: how might we add the following, in decimal?

986 +123 ----



• Question: how might we add the following, in decimal?

986 +123 ____?



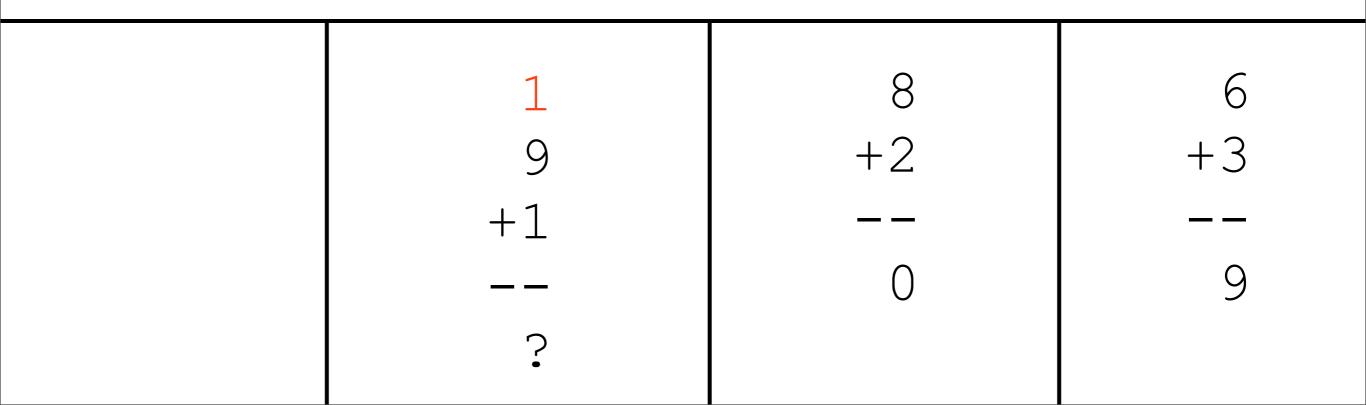
• Question: how might we add the following, in decimal?

986 +123 ----

Carry: 1	8	6
	+2	+3
		— —
	0	9

• Question: how might we add the following, in decimal?

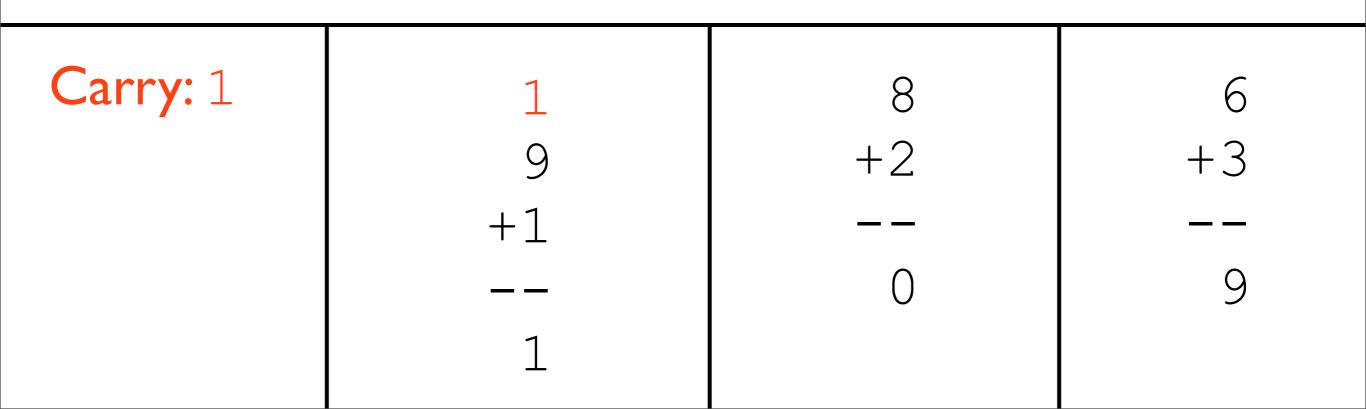
986 +123 ----



• Question: how might we add the following, in decimal?

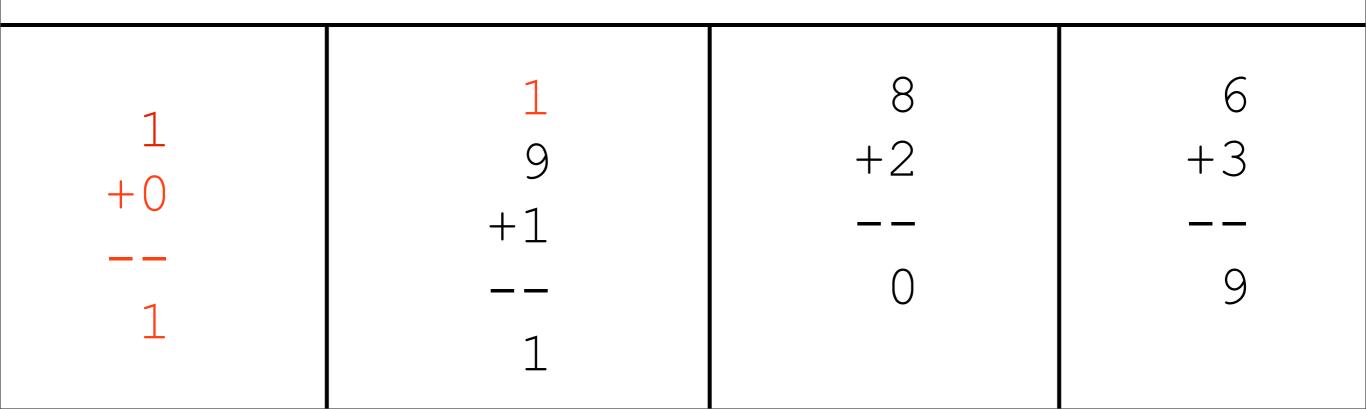
986 +123 ----

?



• Question: how might we add the following, in decimal?

986 +123 ----?

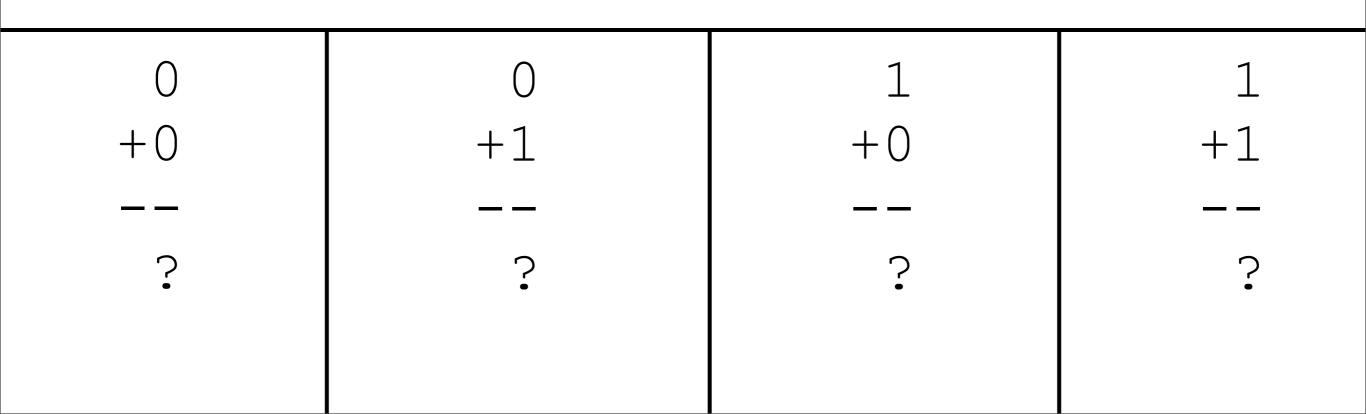


Core Concepts

- We have a "primitive" notion of adding single digits, along with an idea of *carrying* digits
- We can build on this notion to add numbers together that are more than one digit long

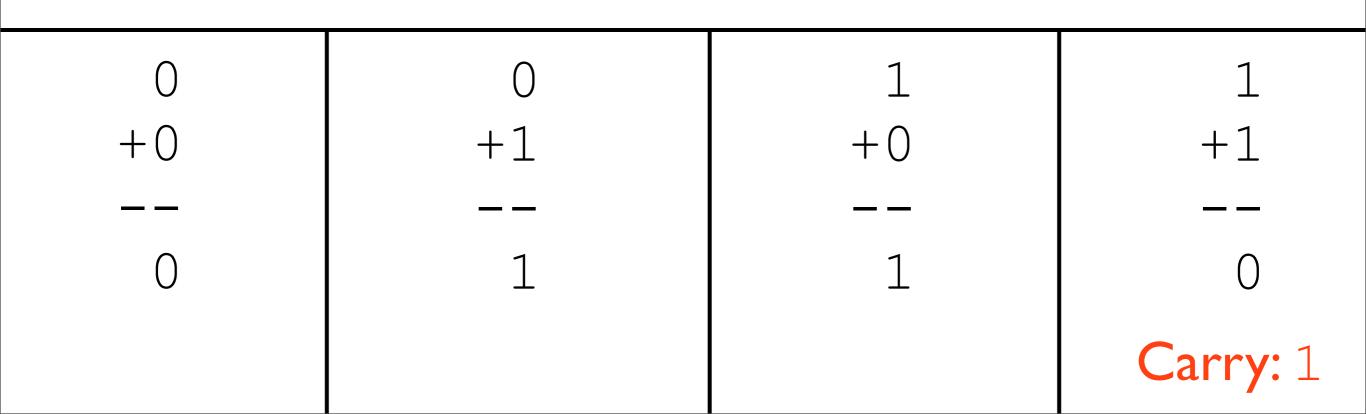
Now in Binary

• Arguably simpler - fewer one-bit possibilities



Now in Binary

• Arguably simpler - fewer one-bit possibilities



Chaining the Carry

• Also need to account for any input carry

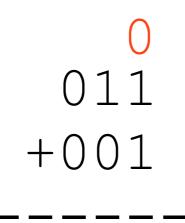
0	0		0		0	
0	0		1		1	
+0	+1		+0		+1	
— —	——					
0	1		1		0	Carry: 1
1	1		1		1	
0	0		1		1	
+0	+1		+0		+1	
——						
1	0	Carry: 1	0	Carry: 1	1	Carry: 1

• How might we add the numbers below?

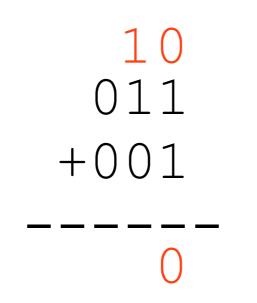
011

+001

• How might we add the numbers below?



• How might we add the numbers below?



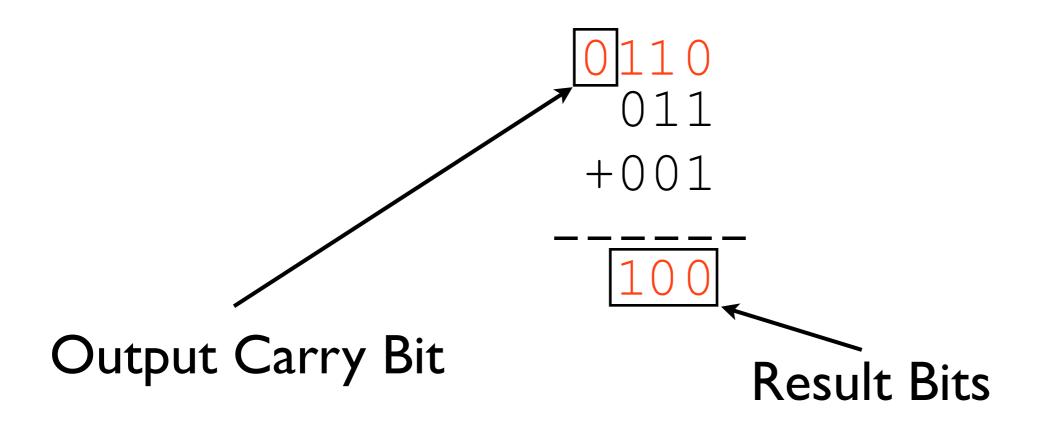
• How might we add the numbers below?

110 011 +001 _____

• How might we add the numbers below?

0110 011 +001 _____

• How might we add the numbers below?

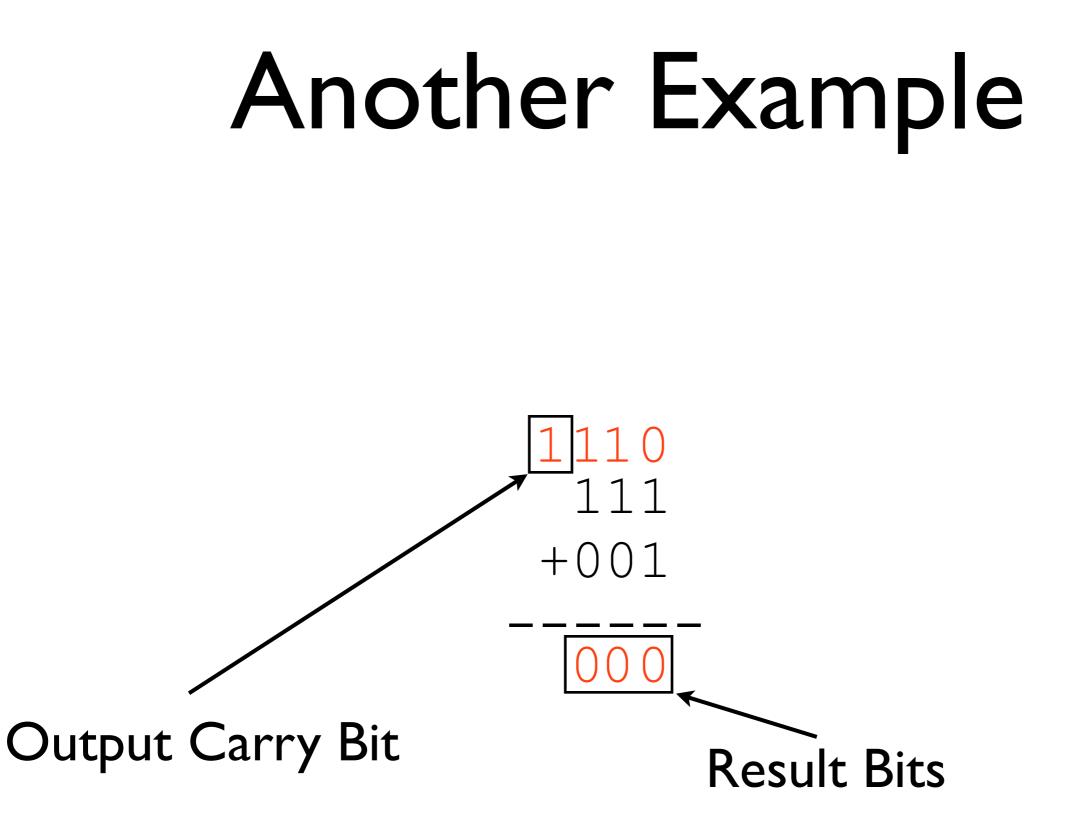


111 +001

0 111 +001

10 111 +001 _____

110 111 +001 _____

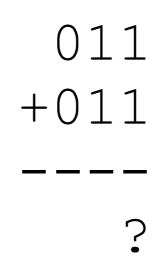


Output Carry Bit Significance

- For unsigned numbers, it indicates if the result did not fit all the way into the number of bits allotted
- May be an error condition for software

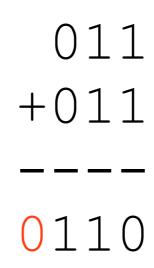
Signed Addition

• Question: what is the result of the following operation?



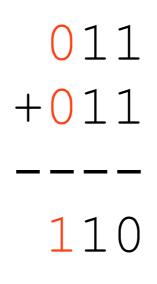
Signed Addition

• Question: what is the result of the following operation?



Overflow

• In this situation, overflow occurred: this means that both the operands had the same sign, and the result's sign differed



• Possibly a software error

Overflow vs. Carry

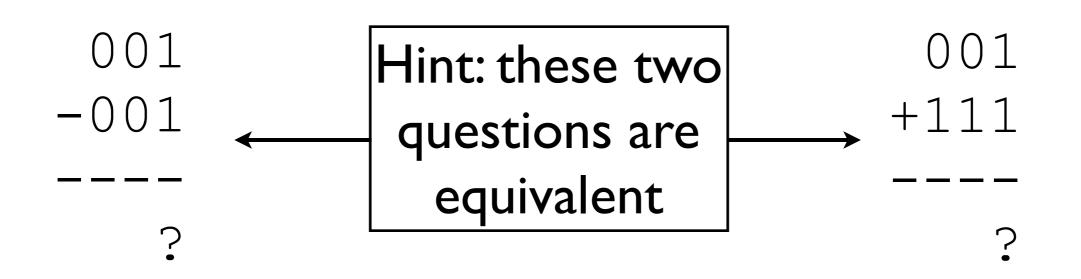
- These are **different ideas**
 - Carry is relevant to **unsigned** values
 - Overflow is relevant to **signed** values

No Overflow; Carry	Overflow; No Carry	Overflow; Carry	No Overflow; No Carry
000	110	011	010
111 +001 	011 +011 	111 +100 	001 +001

Subtraction

Subtraction

- Have been saying to invert bits and add one to second operand
- Could do it this way in hardware, but there is a trick



Subtraction Trick

- Assume we can cheaply invert bits, but we want to avoid adding twice (once to add 1 and once to add the other result)
- How can we do this easily?

Subtraction Trick

- Assume we can cheaply invert bits, but we want to avoid adding twice (once to add 1 and once to add the other result)
- How can we do this easily?
 - Set the initial carry to 1 instead of 0

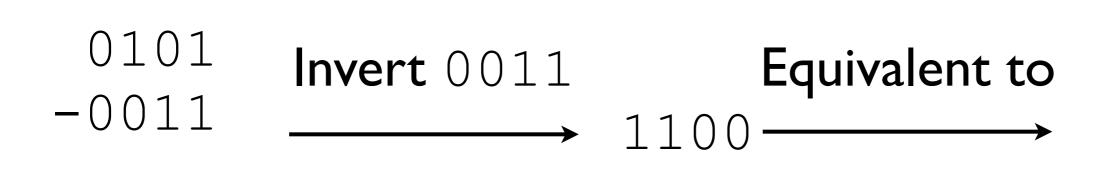
Subtraction Example

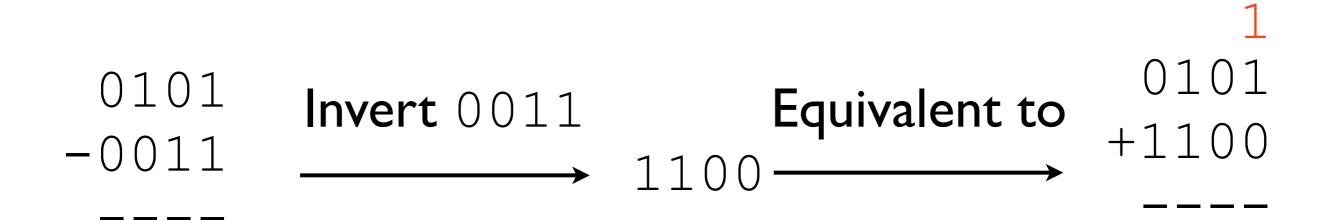
0101 -0011

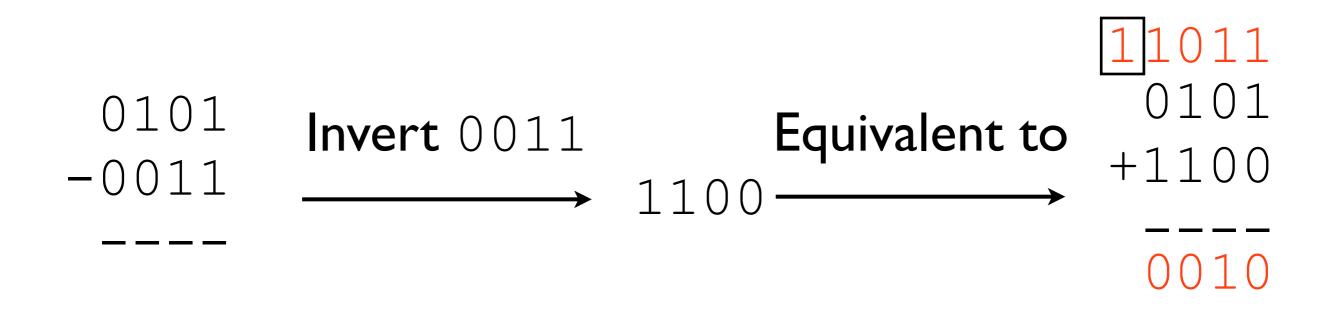
Subtraction Example











Floating Point Introduction

Question

How might we represent floating point numbers?

1.25 47.9 0.82

Enter IEEE-754

- Standardized floating point representation and operations
- Modern systems all use this
- Complex and weird

Enter IEEE-754

- Standardized floating point representation and operations
- Modern systems all use this
- Complex and weird

$$min(X, Y) =? min(Y, X)$$

Enter IEEE-754

- Standardized floating point representation and operations
- Modern systems all use this
- Complex and weird

Based on the idea of scientific notation

Based on the idea of scientific notation

4.23 * 10⁷

Based on the idea of scientific notation

4.23 × 10⁷

Save these

Based on the idea of scientific notation

Save these

Caveat: this is in binary

1.1 * 2-1

Components

1.1 * 2-1

- Sign bit (+/-)
- Exponent
- Fraction / mantissa