華南理工大學
South China University of Technology

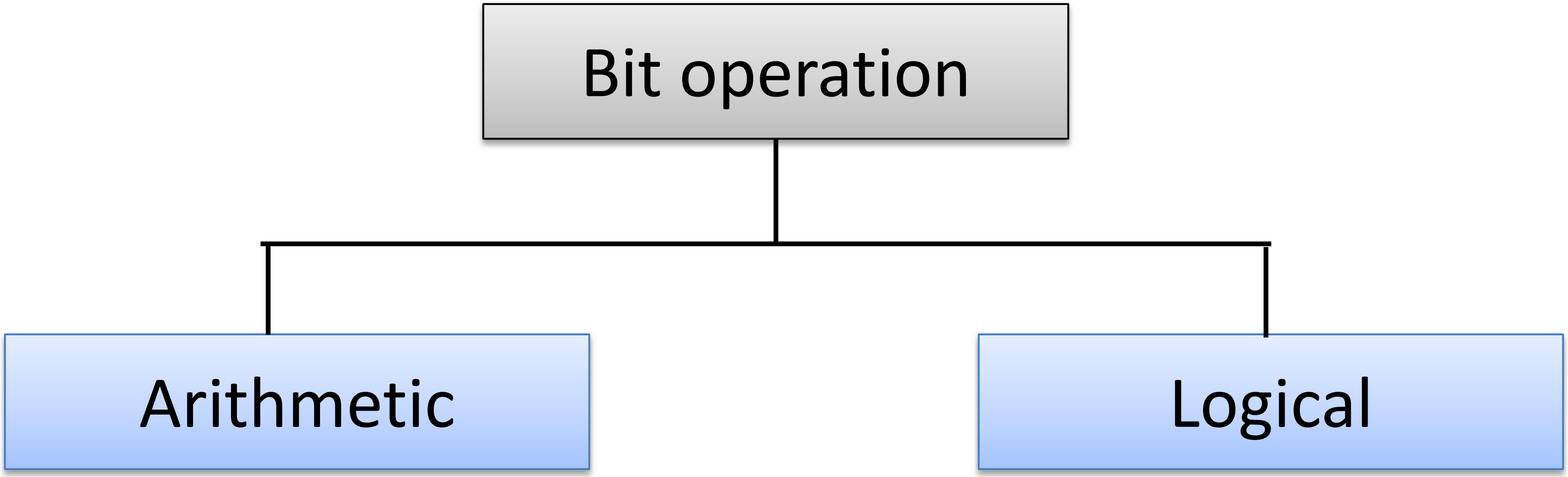
Intro to Computer Science and Software Engineering

Operations on bits

Dr Yubei Lin
yupilin@scut.edu.cn
School of Software Engineering

Operations on bits

- Two broad categories: arithmetic and logical



所有图片均来自网络

Arithmetic operations

- Arithmetic operations involve Addition, Subtraction, Multiplication and Division, on integers and float-point numbers.
- We assume that all number are represented in two’s complement code.
- We only look at addition and subtraction!
  - Multiplication → repeated addition
  - Division → repeated substraction

所有图片均来自网络

Addition in Two's Complement

Adding bits

Number of 1s	Result	Carry
None	0	
One	1	
Two	0	1
Three	1	1

Rule of Adding Integers in Two's Complement

Add 2 bits and propagate the carry to the next column. If there is a final carry after the leftmost column addition, discard it.

Example 0

(+17) + (+22) = (+39)

Carry			1							
	0	0	0	1	0	0	0	1	+	(+17)
	0	0	0	1	0	1	1	0		(+22)
--										
Result	0	0	1	0	0	1	1	1		(+39)

Example 1

(+24) + (-17) = (+7)

	Discard!										
Carry	1	1	1	1	1						
		0	0	0	1	1	0	0	0	+	(+24)
		1	1	1	0	1	1	1	1		(-17)
Result		0	0	0	0	0	1	1	1		(+7)

Example 2

(-35) + (+20) = (-15)

Carry			1	1	1					
	1	1	0	1	1	1	0	1	+	(-35)
	0	0	0	1	0	1	0	0		(+20)
Result	1	1	1	1	0	0	0	1		(-15)

Example 2

(+127) + (+3) = (+130) ✓

Carry	1	1	1	1	1	1	1			
	0	1	1	1	1	1	1	1	+	(+127)
	0	0	0	0	0	0	1	1		(+3)
Result	**1**	0	0	0	0	0	1	0		(-126)

Overflow!

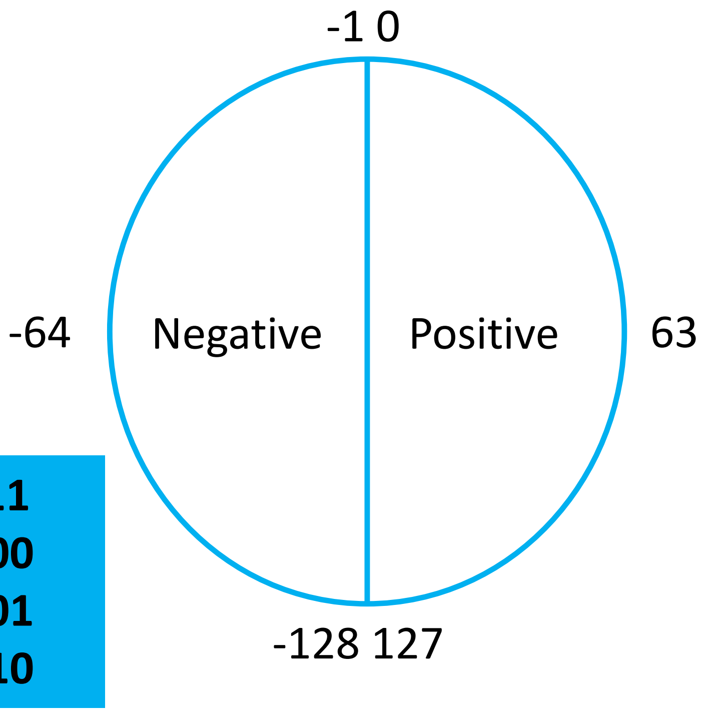
✘

Overflow

- Overflow is an error that occurs when you try to store a number that is not within the range defined by the allocation.
  – The result of operations exceed the range!

Rule of numbers in Two’s Complement representation

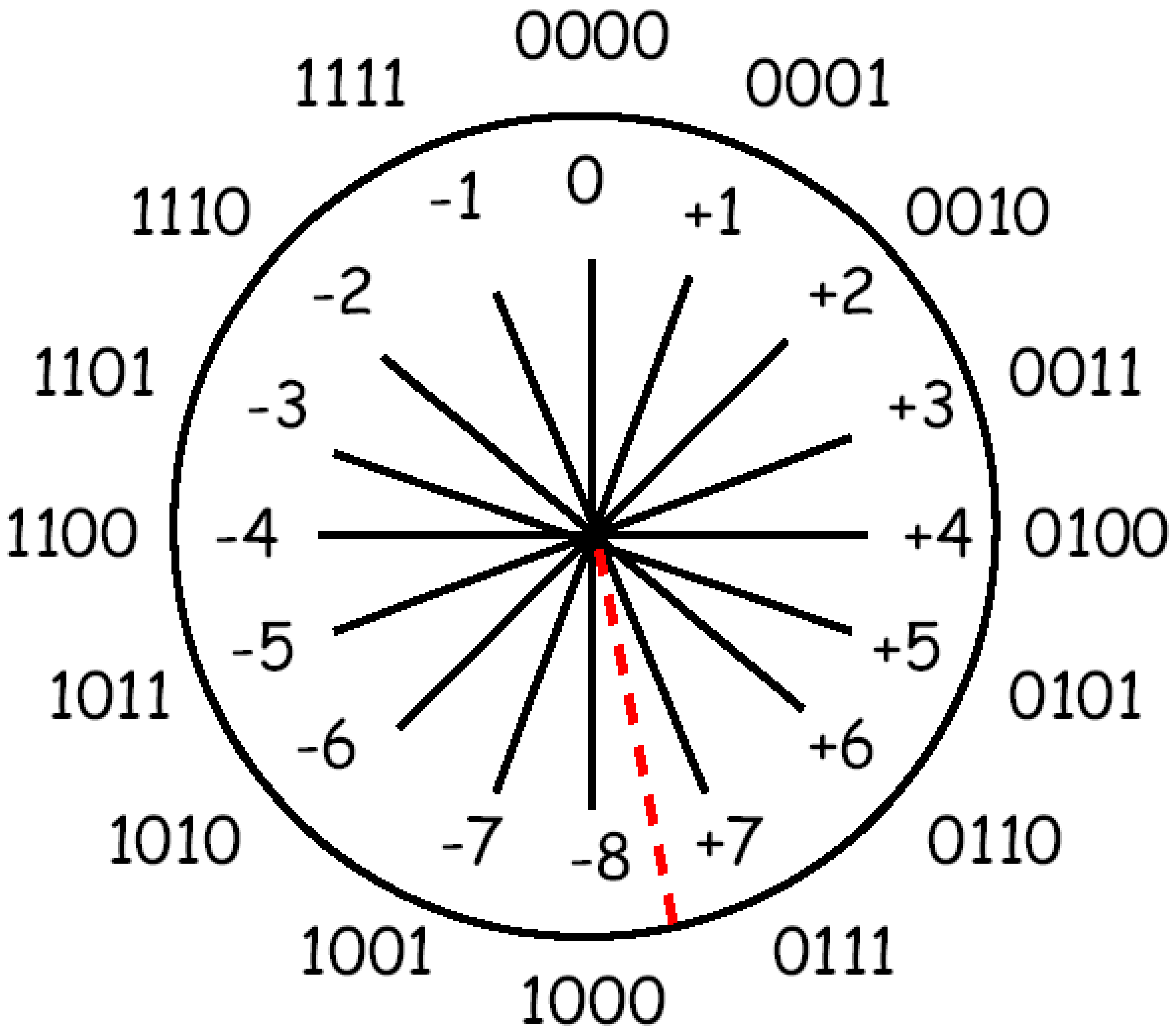
N	$-(2^{N-1})$	---	0	---	$+(2^{N-1}-1)$
8	$-(2^{8-1})$	---	0	---	$+(2^{8-1}-1)$
	-128				+127



127: 0111 1111
-128: 1000 0000
-127: 1000 0001
-126: 1000 0010

Overflow

Number Circle for 4-bit Two's Complement numbers



所有图片均来自网络

Overflow

Rule of Avoiding Overflow

When you do arithmetic operations on numbers in a computer, remember that number and the result should be in the range defined by the bit allocation.

所有图片均来自网络

Subtraction in Two’s Complement

To subtraction, negate the second number and add!

Negate = Two’s Complement

Num1 - Num2 ←→ Num1 + (-Num2)

Example 0

(+101) – (+62)	**← →**	**(+101) + (-62)**	**→**	**(+39)**

Carry	1	1									
		0	1	1	0	0	1	0	1	+	(+101)
		1	1	0	0	0	0	1	0		(-62)
Result		0	0	1	0	0	1	1	1		(+39)

所有图片均来自网络

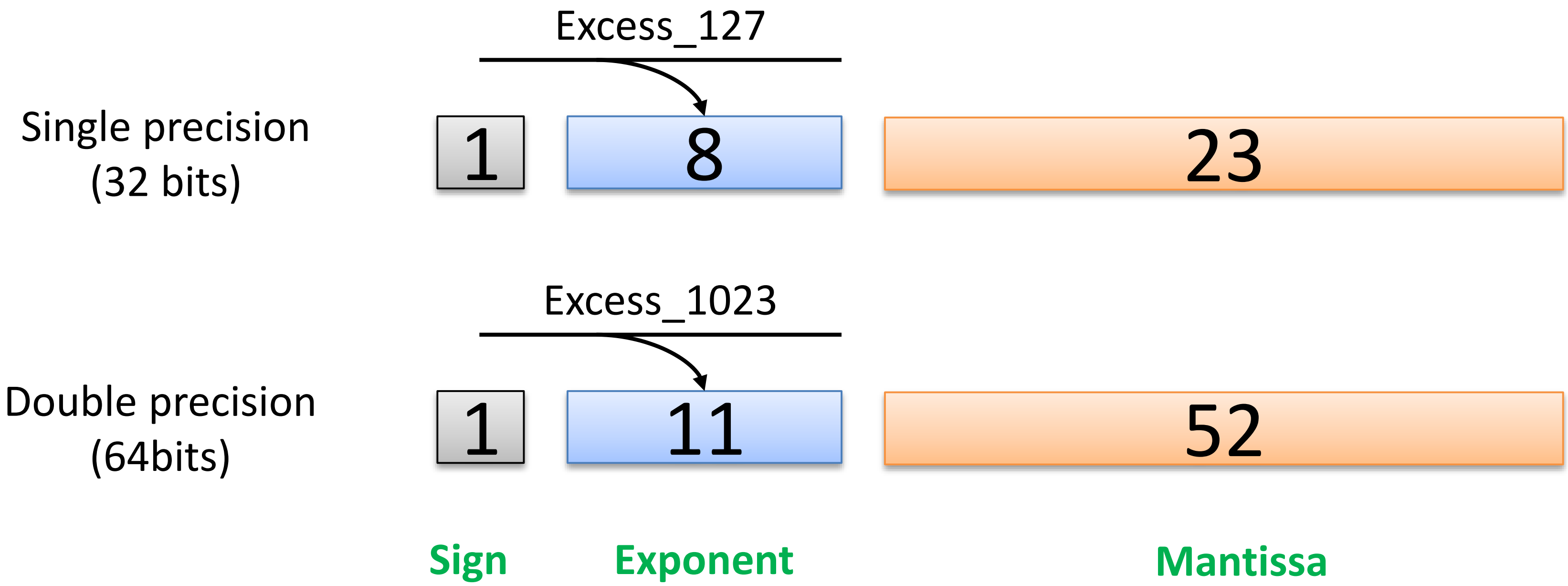
Arithmetic Operations on Floating-Point Numbers

- We can apply addition, subtraction, multiplication and division to **floats**.
- We assume that floats are stored in IEEE format.
- And cover overall concept only ~
  - Addition/Subtraction
  - By example!

所有图片均来自网络

IEEE Standards for floating-point

- Three methods with two for storing number in memory
  - Single precision and double precision



所有图片均来自网络

Example 0

Sign	Exponent	Mantissa	Float Num
0	1000 0100	1011 0000 0000 0000 0000 000	+ 2^5 * 1.1011
0	1000 0010	0110 0000 0000 0000 0000 000	+ 2^3 * 1.011

+ 2^5	*	1.1011
+2^3	*	1.011
+ -----------------------------------		

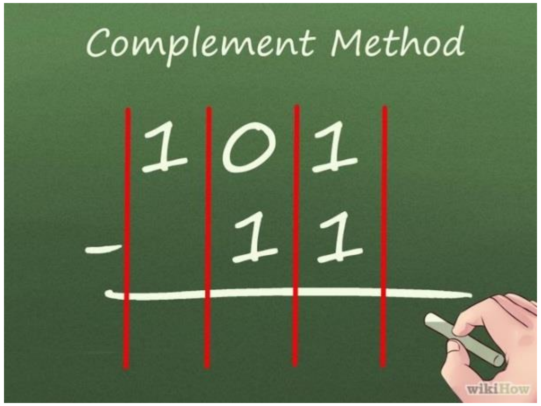
+ 2^5	*	1.1011
+2^5	*****	**0.01011**
+ -----------------------------------		
+2^5	*	10.00001

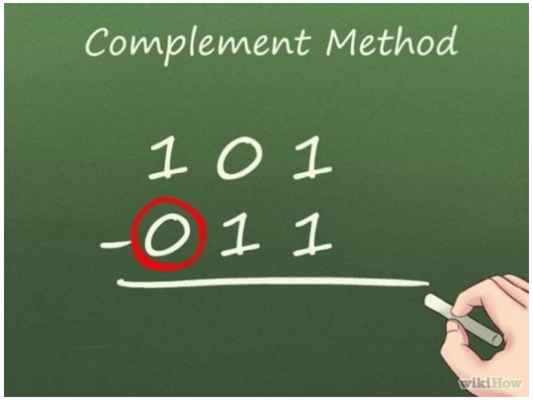
+2^6	*	1.000001

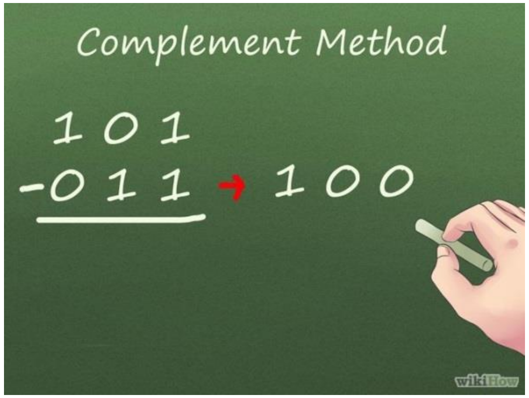
0	1000 0101	0000 0100 0000 0000 0000 000	+ 2^6 * 1.000001

Subtracting bits

- Two methods
  - Using the Borrow Method
  - Using the Complement Method







所有图片均来自网络

Example 1

Sign	Exponent	Mantissa	Float Num
0	1000 0100	1011 0000 0000 0000 0000 000	+ 2^5 * 1.1011
1	1000 0010	0110 0000 0000 0000 0000 000	- 2^3 * 1.011

+ 2^5	*	1.1011
- 2^3	*	1.011
- -------------------------------------		

+ 2^5	*	1.1011
+ 2^5	*****	**0.01011**
- -------------------------------------		
+2^5	*	**1.01011**

+2^5	*	1.01011

0	1000 0100	0101 1000 0000 0000 0000 000	+ 2^5 * 1.01011

Example 1

- 1.1011 – 0.01011 = ?

1.10110
0.01011

1.10110
1.10101

1.10110
1.10101
11.01011

✗

所有图片均来自网络