Binary Numbers

Dr. Baldassano

chrisb@princeton.edu

Yu’s Elite Education

For computers, everything is a number

- Integers and floating point numbers
- Pictures
- Videos
- Music
- Text
- Even programs!

Storing numbers with electricity

- How can we represent a number using electricity?
- Imagine we have a row of lightbulbs that can turn on and off

Binary numbers

- We can write any positive integer as a sum of powers of two:

	8	4	2	1
10	8		2	
4		4		
14	8	4	2	
3			2	1

Binary numbers

- We can write any positive integer as a sum of powers of two:

	8	4	2	1
10	1	0	1	0
4	0	1	0	0
14	1	1	1	0
3	0	0	1	1

Practice converting to binary

	32	16	8	4	2	1
15						
17						
27						
39						
0						
32						

Practice converting from binary

	32	16	8	4	2	1
	0	0	1	1	1	0
	0	0	0	1	1	1
	1	0	0	0	0	1
	1	0	1	0	1	0
	1	1	0	0	0	0
	0	0	1	0	0	1

Counting in binary

Adding in binary

What about negative integers?

- First idea: have a sign bit at the front:
  - 0 = negative, 1 = positive

	Sign	16	8	4	2	1
	0	0	1	1	1	0
	0	0	0	1	1	1
	1	0	0	0	0	1
	1	0	1	0	0	1

Problems with sign bit alone

- What happens if we count in binary and convert each number to decimal?

Sign	2	1	
0	0	0	-0
0	0	1	-1
0	1	0	-2
0	1	1	-3
1	0	0	0
1	0	1	1
1	1	0	2
1	1	1	3

Problems with sign bit alone

- Two weird things:
  - There are two zeros!
  - Number line has big jump between negatives and zero – would require special-purpose circuitry in the computer
- Instead, let's lay out the number line in order:

Sign	2	1	Before	After
0	0	0	-0	-4
0	0	1	-1	-3
0	1	0	-2	-2
0	1	1	-3	-1
1	0	0	0	0
1	0	1	1	1
1	1	0	2	2
1	1	1	3	3

Flip the sign bit

- To make addition easier to compute, and to be consistent with regular binary numbers, let’s switch sign=1 to negative, sign=0 to positive

Sign	2	1	
1	0	0	-4
1	0	1	-3
1	1	0	-2
1	1	1	-1
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3

Two’s complement

- This is called “two’s complement” representation
- To convert a negative two’s complement binary number, flip all bits and add 1

Sign	4	2	1	
1	1	1	0	-> flip to 001 -> 1 + 1 -> -2
1	1	1	1	-> flip to 000 -> 0 + 1 -> -1
1	0	0	1	-> flip to 110 -> 6 + 1 -> -7
1	0	0	0	-> flip to 111 -> 7 + 1 -> -8

Two’s complement practice

	Sign	16	8	4	2	1
	1	0	1	1	1	0
	1	0	0	1	1	1
	1	0	0	0	0	1
	0	0	1	0	1	0
	1	1	0	0	0	0
	1	0	1	0	0	1

Adding with two’s complement

- Biggest advantage of two’s complement is that we can add positive and negative numbers

	Sign	8	4	2	1	
	1	1	1	1	0	
+	0	0	0	1	1	
	0	0	0	0	1	

Adding examples

	Sign	8	4	2	1	
	1	0	1	1	1	
+	0	0	1	1	0	

Adding examples

	Sign	8	4	2	1	
	1	1	1	1	1	
+	0	1	0	0	0	

Adding examples

	Sign	8	4	2	1	
	1	1	1	0	1	
+	1	1	1	1	0	

Hexadecimal numbers

- Writing out binary numbers takes a long time and is easy to mess up
- Instead we usually write binary numbers in “hexadecimal” (base 16) by looking at groups of 4 bits

Integers in programming languages

- Most programming languages require you to say how many bits you want to use, and (for integers) whether you want negatives
- C++:
  - unsigned short int – 16 bits, positive
  - signed short int – 16 bits, negative/positive
  - long int – 32 bits, negative/positive
  - long long int – 64 bits, negative/positive

Numbers in python

Floating-point numbers