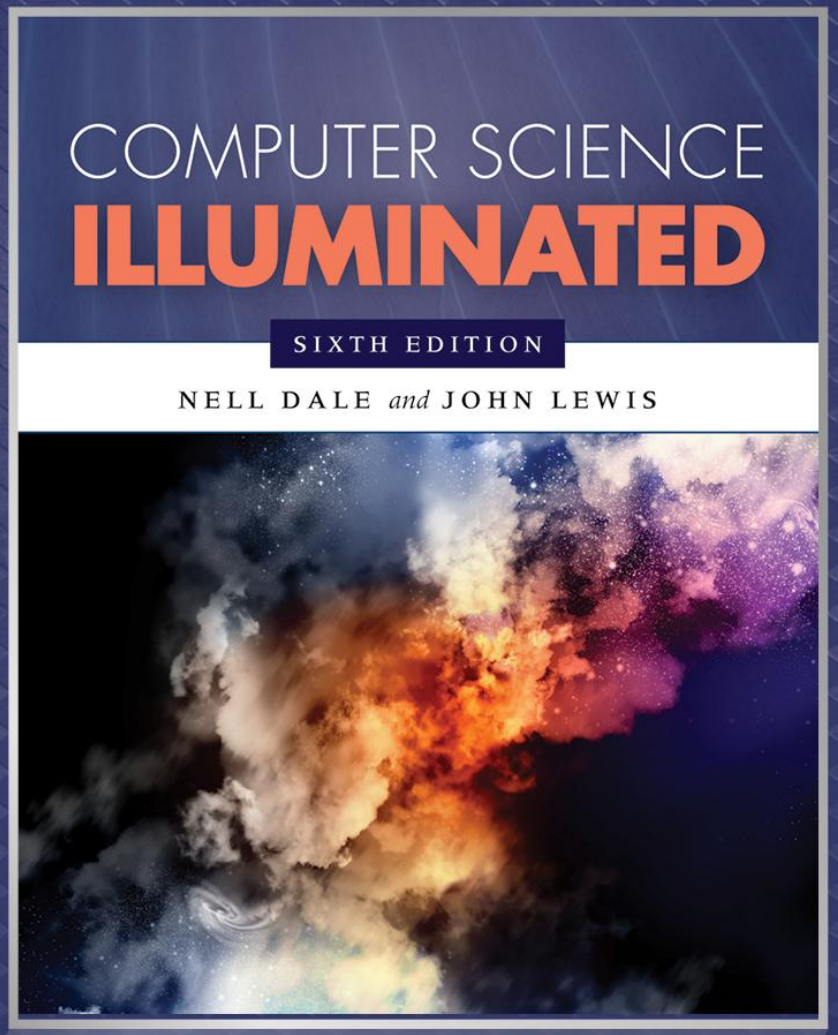
Data Representation



Chapter Goals

- Distinguish between analog and digital information
- Explain the binary formats for negative and floating-point values
- Describe the characteristics of the ASCII and Unicode character sets

Data and Computers

Computers are **multimedia** devices, dealing with a vast array of information categories

Computers store, present, and help us modify

- Numbers
- Text
- Audio
- Images and graphics
- Video

All stored as binary digits (**bits**)

Binary Numbers and Computers

Computers have storage units called binary digits or bits

Low Voltage = 0
High Voltage = 1 **all bits have 0 or 1**

… or the other way around, but we don't need to worry about that

Binary and Computers

Byte
8 bits

The number of bits in a word determines the word length of the computer, which is usually a multiple of 8

- **32-bit machines**
- **64-bit machines etc.**

Analog and Digital Information

Computers are finite!

How do we represent an infinite world?

We represent enough of the world to satisfy our computational needs and our senses of sight and sound

Analog and Digital Information

Information can be represented in one of two ways: **analog** or **digital**

Analog data

A continuous representation, analogous to the actual information it represents

Digital data

A discrete representation, breaking the information up into separate elements

Analog and Digital Information

A mercury thermometer is an analog device

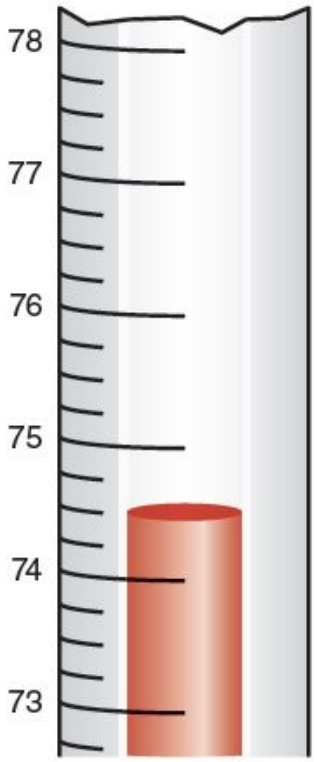


FIGURE 3.1 A mercury thermometer continually rises in direct proportion to the temperature

Analog and Digital Information

Computers cannot work well with analog data, so we digitize the data

Digitize

Breaking data into pieces and representing those pieces separately

Why do we use binary to represent digitized data?

Electronic Signals

Important facts about electronic signals:

- An analog signal continually fluctuates in voltage up and down
- A digital signal has only a high or low state, corresponding to the two binary digits
- All electronic signals (both analog and digital) degrade as they move down a line
- The voltage of the signal fluctuates due to environmental effects

Electronic Signals *(Cont'd)*

FIGURE 3.2 An analog signal and a digital signal

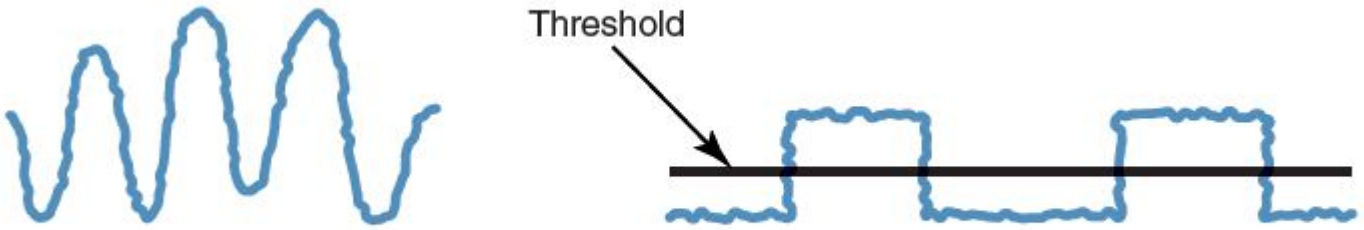


FIGURE 3.3 Degradation of analog and digital signals

Periodically, a digital signal is **reclocked** to regain its original shape

Binary Representations

- Each bit can be either 0 or 1, so it can represent a choice between two possibilities (or “two things”)

- Two bits can represent four things *(Why? Hint:* 00, 01, 10, 11*.)*

How many things can three bits represent?

How many things can four bits represent?

How many things can eight bits represent?

Binary Representations

1 Bit	2 Bits	3 Bits	4 Bits	5 Bits
0	00	000	0000	00000
1	01	001	0001	00001
	10	010	0010	00010
	11	011	0011	00011
		100	0100	00100
		101	0101	00101
		110	0110	00110
		111	0111	00111
			1000	01000
			1001	01001
			1010	01010
			1011	01011
			1100	01100
			1101	01101
			1110	01110
			1111	01111
				10000
				10001
				10010
				10011
				10100
				10101
				10110
				10111
				11000
				11001
				11010
				11011
				11100
				11101
				11110
				11111

FIGURE 3.4 Bit combinations

Binary Representations

How many bits are needed to represent 32 things? One hundred things?

How many things can n bits *represent?*

Why?

What happens every time you increase the number of bits by one?

Representing Natural Numbers

8-bit Binary Representation	Natural Number
01111111	127
01111110	126
...	...
00000011	3
00000010	2
00000001	1
00000000	0

- Easy! Just convert to binary
- Computers store data in fixed-size chunks, so we have leading zeros

What do the integers include that the natural numbers do not?

Representing Negative Values

Signed-magnitude number representation

- Used by humans

- The sign represents the ordering (the negatives come before the positives in ascending order)

- The digits represent the magnitude (the distance from zero)

Representing Negative Values

Problem: Two zeroes (positive and negative)

No problem for humans, but would cause unnecessary complexity in computers

Solution: Represent integers by associating them with natural numbers

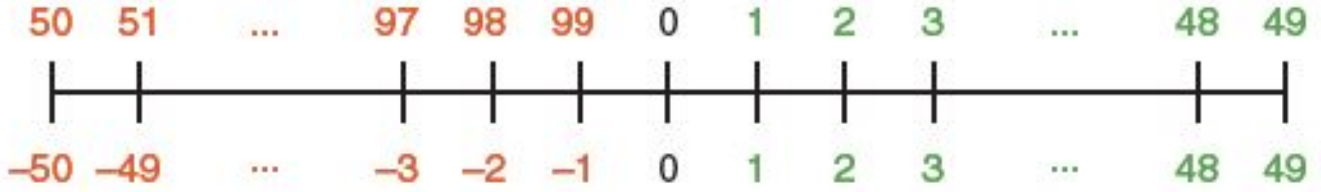
- Half the natural numbers will represent themselves
- The other half will represent negative integers

Representing Negative Values

Using two decimal digits:

let 0 through 49 represent 0 through 49

let 50 through 99 represent -50 through -1



Representing Negative Values

To perform addition, add the numbers and discard any carry to the hundreds digit

Signed-Magnitude	New Scheme
5 + −6 −1	5 + 94 99
−4 + 6 2	96 + 6 2
−2 + −4 −6	98 + 96 94

Representing Negative Values

To perform addition, add the numbers and discard any carry to the hundreds digit

Signed-Magnitude	New Scheme
5 + −6 −1	5 + 94 99
−4 + 6 2	96 + 6 2
−2 + −4 −6	98 + 96 94

Now you try it

48 (signed-magnitude)
- 1
47

How does it work in the new scheme?

Representing Negative Values

To perform subtraction, use A – B = A + (-B)

Add the negative of the second to the first

Signed-Magnitude	New Scheme	Add Negative
– 5 – 3 – 8	95 – 3	95 + 97 92

Try these:

4	*4*	*- 1*
- 3	*- (- 3)*	*- 2*

Representing Negative Values

Ten's complement representation:

We can use formula to compute representation of a negative number:

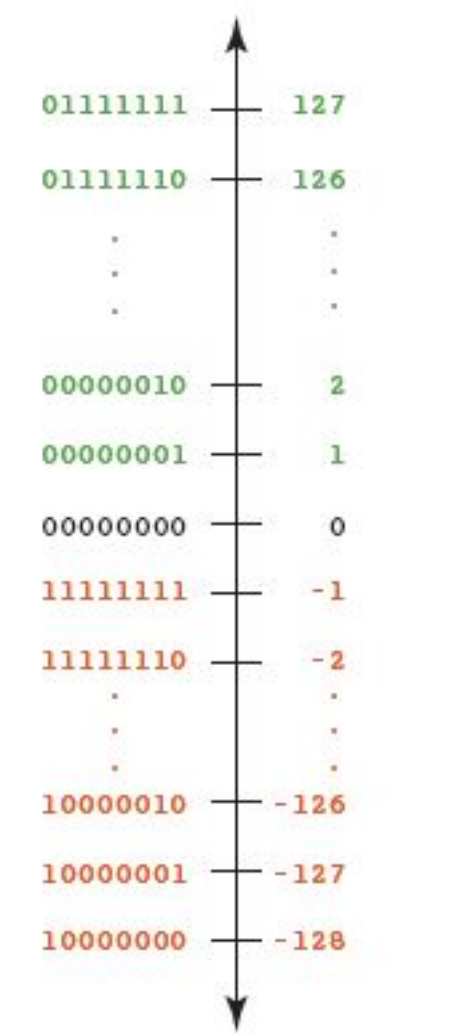
$$\text{Negative}(I) = 10^k - I, \text{ where } k \text{ is the number of digits}$$

For example, -3 is Negative(3), so using two digits, its representation is

Negative(3) = 100 – 3 = 97

What do we get if we try this in binary?

Representing Negative Values



Two's Complement

- Say we have 8 bits
- The number 5 is therefore 00000101

Given our equation:

Negative(I) = 2^k - I

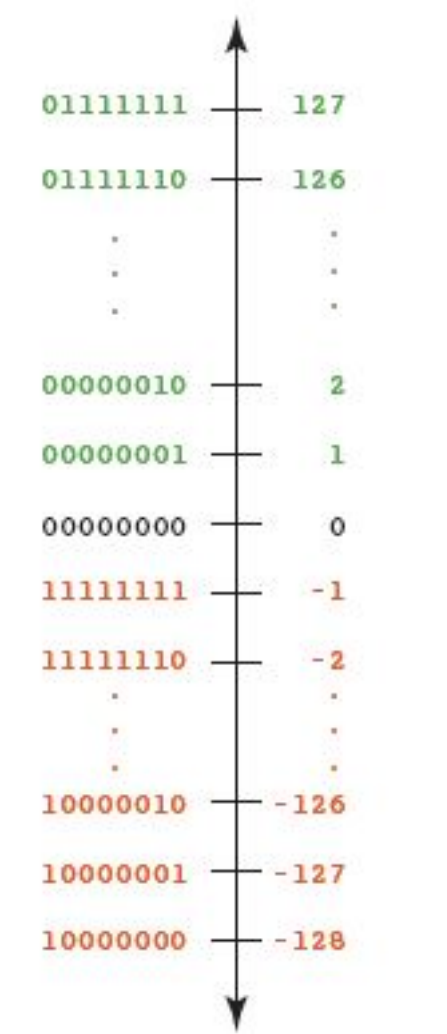
Negative(00000101)

= 2^8 - 00000101

= 100000000 -
000000101

= 11111011

Representing Negative Values



Two's Complement (cheat's method)

Positive number:

- Normal binary value

Negative number:

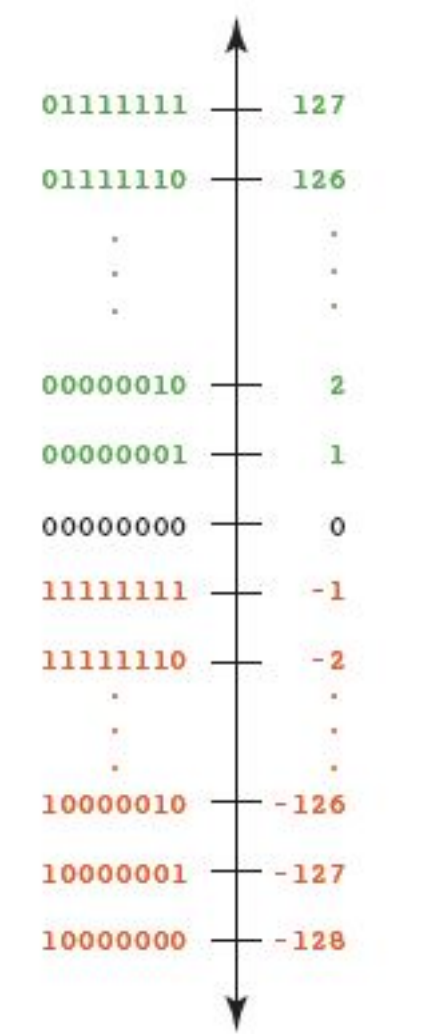
- Normal binary value
- Invert all of the bits
- Add 1

Number 5 in binary: 0000 0101

Invert bits: 1111 1010

Add 1: 1111 1011

Representing Negative Values



Two's Complement

The binary number line is easier to read when written vertically

Remember our table showing how to represent natural numbers?

Do you notice something interesting about the left-most bit?

Representing Negative Values

Addition and subtraction are the same as in ten's complement arithmetic

-127	10000001
+ 1	00000001
-126	10000010

Representing Negative Values

What is:

-126
- -127

What is -126? What is -127?

Representing Negative Values

126 = 01111110

-126 = 10000001 + 1

= 10000010

127 = 01111111

-127 = 10000000 + 1

= 10000001

Representing Negative Values

So what is 10000010 - 10000001 ?

Do you want to do all that borrowing?

We had an easier way before!

Remember A - B = A + (-B)

10000010 - 10000001 (or -126 - -127)

= 10000010 + (01111110 + 1)

= 10000010 + 01111111

= 100000001 ← This is 9 bits, but we only allow 8!
Discard the leading carried 1.

= 00000001

Representing Negative Values

Addition and subtraction are the same as in ten's complement arithmetic

What if the computed value won't fit?

Integer Overflow

Given a fixed number of digits in our representation scheme, overflow occurs when a value is created which is outside of the range representable with that number of digits.

For example, if we only have two decimal digits in our representation:

```
  99
+ 01
 100
```

which results in a result of 00!

Integer Overflow

If each value is stored using 8 bits in a two's complement scheme, then 127 + 3 overflows:

```
  01111111
+ 00000011
  10000010
```

Apparently, 127 + 3 is -126!

Remember when we said we would always fail in our attempt to map an infinite world onto a finite machine?

Most computers use 32 or 64 bits for integers, but there are always infinitely many that aren't represented

Integer Overflow

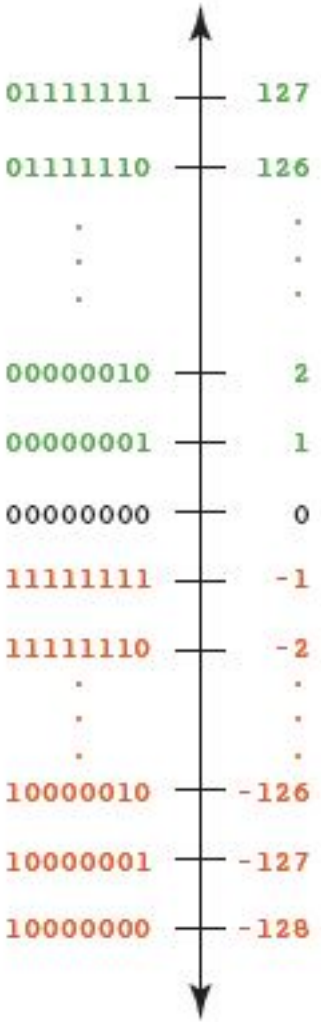
Overflow can happen in both directions:

```
  10000000
- 00000001
```

Or, using A + (-B):

```
  10000000
+ 11111111
 101111111
```

Apparently, -128 - 1 is 127!



Integer Overflow

Duck Hunt: https://www.youtube.com/watch?v=AjUpe7Oh1j0

PacMan: https://www.youtube.com/watch?v=FcI42czB2q4

Casino refuses to pay out $42,949,672.76

Exploding rockets!

Stolen crypto tokens!

Number Overflow

How do we get around this?

Sometimes with fancy tricks.

Sometimes we can’t.

Coming in Concepts of Computer Science II !

We will look at precision and errors that come from the limitations of a computer.