

# CS101 Lecture 11: Data Representation: Binary Numbers

Number Systems  
Binary Numbers

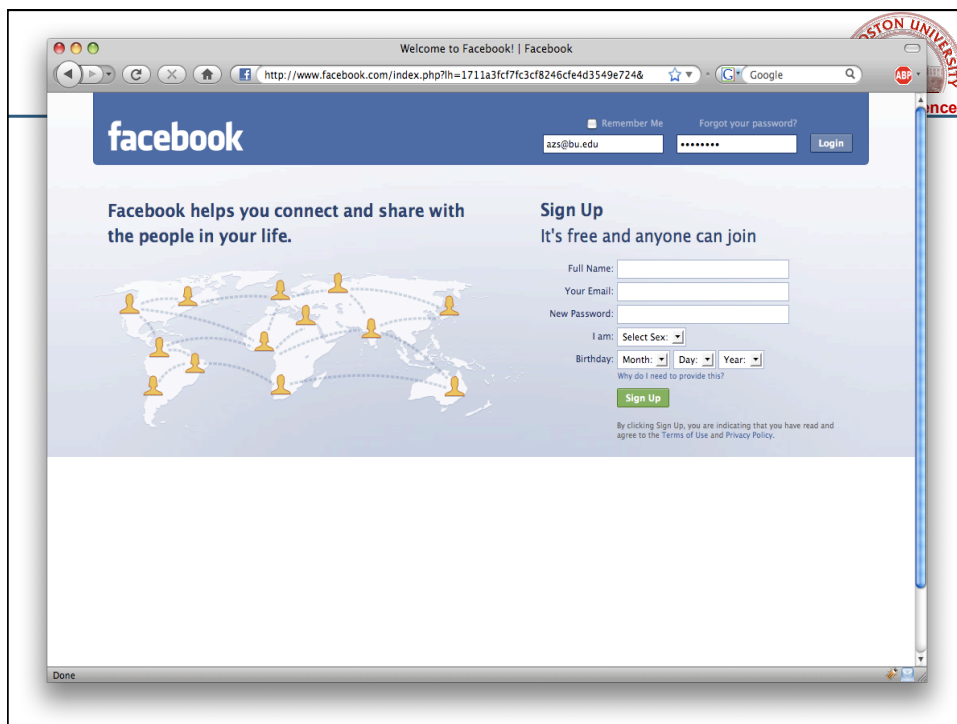
Aaron Stevens (azs@bu.edu)  
23 January 2013




```

11110010000110001001001111000011101010010110011010000100010110000011010001110100110110111
01100100000101000010101011000010100110010010011000011100000100010101101101111101111010111
1100111000011101111011110001001010100100010010000010110001001000101010010011011001100100
00011000110111011000010000010100111101001110010110100111000111101001011110111100010110
01101000101010100000100000100001011010001110100111010011100001110101011110111110111101
111110111110111101110111110011101110110111110010110011110111101010010100101011111
110001101000111010011011011101100110001101111000100101010010001001011111100011010001110
100110110111011001100011011011100111101001110010110100111000111101001011101100100111010
01100100100010111110101011100110100011010011011011101100100000111000110110110110011011
101100111110110001011010001110100111010011100001110101011110111110111110111110111101110
111011101111001110110110111111001011001111011111000111100111100111100110111111000110
10001110100110110111011001000101000001110001101101110110011101101100110000111011011001
111111011000101100101110111010001010000011011001100001110111011000101100101110
11101000101000001101001110010011110110001011001110000111001011100010110111110111
11101011100010111101010111001101000110010111000011100100111101010111001110100110100111
101001101100110010111110101011110010111011001100011110111110110111001011000001110100110
11111000001000101100001110001111001011100010110111110111110101110000110000011110010000
01000110110000111000111100101110001011011111011111010111110010111111010011010011110100
1101100110010111110101011100110111001011110100110000110000011010001110100111010011100
0010110111001011110001110101110100111101111101100010100001111011111011101101001100101
11011101110100101111010011110011110000110010110001010000011000111101111101110111010011
001011101110111010011110110001011101001100101111000111010010111110100011101001101101101
100111011100000110001111010001100001110010111001011100101111010011110111101001110011
010101111000100010100000101111110101011110010111010010111010011000011000001101000111
1010011101001110000101011100101110001111010111010011110111011011001011000011101111101
1101110100110010111011101101001011101100110000111011101100111111010111000011100111100
1011000101000001100011110111110111011001100101110111011010011110110001011001011101110
100010100000101111111101010111100110111101001111010011000011000001101000111010011101001
11000010110111001011100011110101110100111101111011000101011000101011010110000011011
011000011110111110101111000011000011101001100011100010110110011000101100010100000110001
1110111110111011101001100101110111010011110110001010010011000101111101100010111011011
11010111011001100001111010011001011001001100010111011100010100000101111111010101110011
100111100011111001011010011110000111010010000011101001111000111000101111011000101110
10011001011111000111010010111110101011000011110111000011110011110001111010011101110
00011101001000101111101010101111110110010000110111100001110001001000001101010010000
0110110111010100001100101110101110111011000111101001101001110111110111010000110100011
10010110010111001101010011000001110111000001110111000011110010100000111010111001011010
01101111111001011001011000001110110000010111101110101000111111101010000111111101010
1010111011101110101011100101111111110001110111010101110100111101010011111110100011

```





BOSTON UNIVERSITY  
Computer Science

# !!! MATH WARNING !!!

TODAY'S LECTURE CONTAINS  
TRACE AMOUNTS OF  
ARITHMETIC AND ALGEBRA

PLEASE BE ADVISED THAT CALCULATORS  
WILL BE ALLOWED ON THE QUIZ  
(and that you probably won't need them)

## Overview/Questions



- What gives a number its value?
- What is a number system?
- I've heard that computers use binary numbers. What's a binary number?
- What kind of numbers do computers store and manipulate?

## These are the numbers you know



### **Natural Numbers**

Zero and any number obtained by repeatedly adding one to it.

Examples: 100, 0, 45645, 32

### **Negative Numbers**

A value less than 0, with a – sign

Examples: -24, -1, -45645, -32

## These are the numbers you know



### Integers

A natural number, a negative number, zero

Examples: 249, 0, -45645, -32

### Rational Numbers

An integer or the quotient of two integers

Examples: -249, -1, 0,  $3/7$ ,  $-2/5$

## Numbering Systems



A numbering system assigns meaning to the position of the numeric symbols.

For example, consider this set of symbols:

642

What number is it? Why?

# Numbering Systems



It depends on the numbering system.

642 is  $600 + 40 + 2$  in **BASE 10**

The **base** of a number determines the number of digits (e.g. symbols) and the value of digit positions

## Positional Notation



Continuing with our example...

**642 in base 10 *positional notation* is:**

$$\begin{aligned} 6 \times 10^2 &= 6 \times 100 = 600 \\ + 4 \times 10^1 &= 4 \times 10 = 40 \\ + 2 \times 10^0 &= 2 \times 1 = 2 \quad = 642 \text{ in base 10} \end{aligned}$$

This number is in  
base 10

The power indicates  
the position of  
the number

# Positional Notation



$$642 = 6_3 * 10^2 + 4_2 * 10^1 + 2_1 * 10^0$$

B is the base

As a general form:

$$d_n * B^{n-1} + d_{n-1} * B^{n-2} + \dots + d_1 * B^0$$

n is the number of digits in the number

d is the digit in the  $i^{\text{th}}$  position in the number

## What Would Pooh Do?



# Binary Numbers



Digital computers are made up of electronic circuits, which have exactly 2 states: **on and off**.

Computers use a numbering system which has exactly **2 symbols**, representing on and off.

# Binary Numbers



Decimal is base 10 and has 10 digits:

**0,1,2,3,4,5,6,7,8,9**

Binary is base 2 and has 2, so we use only 2 symbols:

**0,1**

For a given base, valid numbers will only contain the digits in that base, which range from 0 up to (but not including) the base.

# Binary Numbers and Computers



A **binary digit** or **bit** can take on only these two values.

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

Binary numbers are built by concatenating a string of bits together.

Example: 10101010

## Positional Notation: Binary



Recall this general form:

$$d_n * B^{n-1} + d_{n-1} * B^{n-2} + \dots + d_1 * B^0$$

The same can be applied to base-2 numbers:

$$1011_{\text{bin}} = 1 * 2^3 + 0 * 2^2 + 1 * 2^1 + 1 * 2^0$$

$$1011_{\text{bin}} = (1 * 8) + (0 * 4) + (1 * 2) + (1 * 1)$$

$$1011_{\text{bin}} = 8 + 0 + 2 + 1 = 11_{\text{dec}}$$



## Converting Binary to Decimal



*What is the decimal equivalent of the binary number 01101110?*

*(you try it! Work left-to-right)*

## Converting Binary to Decimal



*What is the decimal equivalent of the binary number 01101110?*

$$\begin{aligned}
 0 \times 2^7 &= 0 \times 128 = 0 \\
 + 1 \times 2^6 &= 1 \times 64 = 64 \\
 + 1 \times 2^5 &= 1 \times 32 = 32 \\
 + 0 \times 2^4 &= 0 \times 16 = 0 \\
 + 1 \times 2^3 &= 1 \times 8 = 8 \\
 + 1 \times 2^2 &= 1 \times 4 = 4 \\
 + 1 \times 2^1 &= 1 \times 2 = 2 \\
 + 0 \times 2^0 &= 0 \times 1 = 0 \\
 &= 110 \text{ (decimal)}
 \end{aligned}$$

## Converting Binary to Decimal



*Try another one. What is the decimal equivalent of the binary number 10101011?*

*(you try it! Work left-to-right)*

## Converting Binary to Decimal



*Try another one. What is the decimal equivalent of the binary number 10101011?*

$$\begin{aligned}1 \times 2^7 &= 1 \times 128 = 128 \\+ 0 \times 2^6 &= 0 \times 64 = 0 \\+ 1 \times 2^5 &= 1 \times 32 = 32 \\+ 0 \times 2^4 &= 0 \times 16 = 0 \\+ 1 \times 2^3 &= 1 \times 8 = 8 \\+ 0 \times 2^2 &= 0 \times 4 = 0 \\+ 1 \times 2^1 &= 1 \times 2 = 2 \\+ 1 \times 2^0 &= 1 \times 1 = 1 \\&= 171 \text{ (decimal)}\end{aligned}$$

## Converting from Decimal to Other Bases



### Algorithm (process) for converting number in base 10 to other bases

While (the quotient is not zero)

Divide the decimal number by the new base\*

Make the remainder the next digit to the left in the answer

Replace the original decimal number with the quotient

*\* Using whole number (integer) division only.*

*Example: 3 / 2 gives us a quotient of 1 and a remainder 1*

## Converting Decimal to Binary



What is the binary equivalent of the decimal number 103?

$103 / 2 = 51$ , remainder 1 → rightmost bit

$51 / 2 = 25$ , remainder 1

$25 / 2 = 12$ , remainder 1

$12 / 2 = 6$ , remainder 0

$6 / 2 = 3$ , remainder 0

$3 / 2 = 1$ , remainder 1

$1 / 2 = 0$ , remainder 1 → leftmost bit

$103_{\text{dec}} = 1\ 1\ 0\ 0\ 1\ 1\ 1_{\text{bin}}$

## Converting Decimal to Binary



Now you try one. What is the binary equivalent of the decimal number 201?

### *Recall the algorithm:*

While (the quotient is not zero)  
 Divide the decimal number by the new base\*  
 Make the remainder the next digit to the left in the answer  
 Replace the original decimal number with the quotient

## Converting Decimal to Binary



What is the binary equivalent of the decimal number 201?

$201 / 2 = 100$ , remainder 1  $\rightarrow$  rightmost bit  
 $100 / 2 = 50$ , remainder 0  
 $50 / 2 = 25$ , remainder 0  
 $25 / 2 = 12$ , remainder 1  
 $12 / 2 = 6$ , remainder 0  
 $6 / 2 = 3$ , remainder 0  
 $3 / 2 = 1$ , remainder 1  
 $1 / 2 = 0$ , remainder 1  $\rightarrow$  leftmost bit

$$201_{\text{dec}} = 11001001_{\text{bin}}$$

# Bits, Bytes, Words



## Byte

8 bits – a common unit of computer memory.

## Word

A computer **word** is a group of bits which are passed around together during computation.

The **word length** of the computer's processor is how many bits are grouped together.

- 8-bit machine (e.g. Nintendo Gameboy, 1989)
- 16-bit machine (e.g. Sega Genesis, 1989)
- 32-bit machines (e.g. Sony PlayStation, 1994)
- 64-bit machines (e.g. Nintendo 64, 1996)

# Bytes, Kilobytes, Megabytes



## Data Usage Summary

Messaging Unlimited		
Used	198	
200MB OF DATA		
Plan MB	200	
MB Used	275	
5. MB Billed at \$15.00/200MB	75	15.00
1 Gigabyte (GB) = 1024MB, 1 Megabyte (MB) = 1024KB		

A byte is 8 bits...

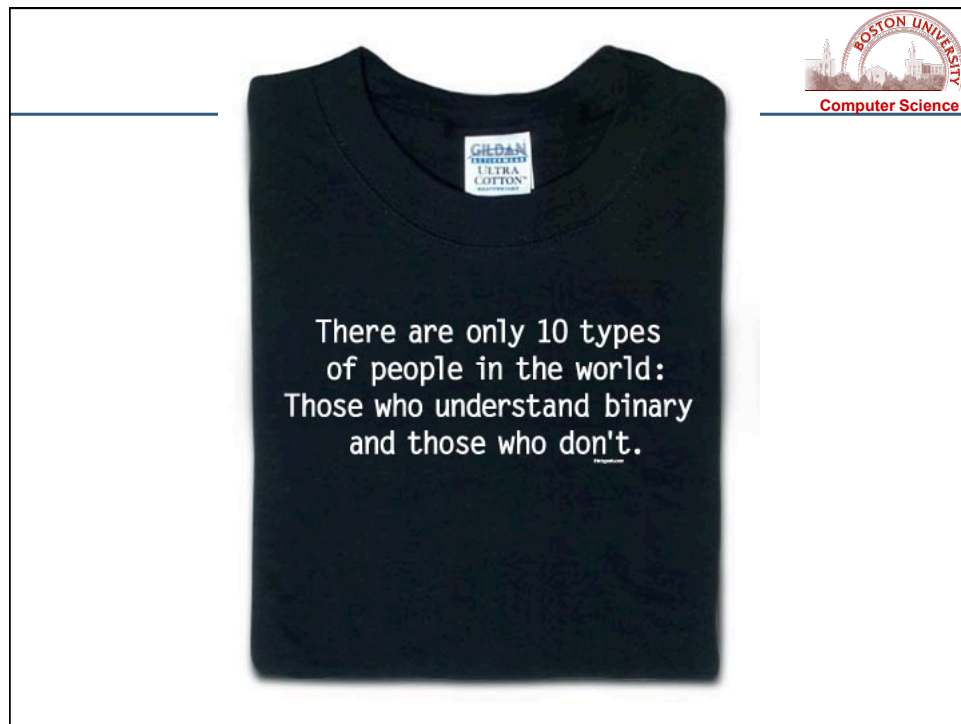
... enough to hold one character.

A kilobyte (KB) is 1024 bytes...

...enough to hold about one page of text.

A megabyte (MB) is 1024 kilobytes, or 1,048,576 bytes...

... about enough to hold a digital picture.



## What You Learned Today



- Encoding: Symbols Represent Values
- Number Systems
- Binary Numbers, Bits, and Bytes
- Algorithms: converting binary to decimal, converting decimal to binary

## Announcements and To Do



- Homework 1 due tonight!
- Readings:
  - Reed ch 5, pp 83-87, 89-90 (today)
  - Reed ch 2, pp 19-26 (Friday)
- Lab 2 on THURSDAY
  - Go to your scheduled lab time.
  - You must submit a document from the lab as evidence of attendance!

## Want to learn more?



If you've read this far, maybe you'd like to learn about other binary representations of other types of numbers?

Read about this on Wikipedia and we can discuss your questions:

- [Two's complement](#) (negative numbers)
- [IEEE754](#) (real numbers)