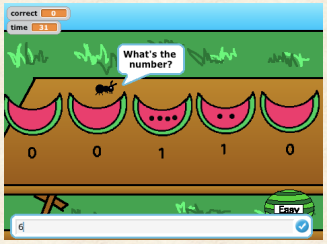
UNIVERSITY of WISCONSIN-MADISON
Computer Sciences Department

CS 202
Introduction to Computation

Professor Andrea Arpaci-Dusseau
Fall 2010

Lecture 7: How does a computer… represent information and data?



Theme 2: How do computers…

… Represent data and information?
… Represent numbers, words, pictures, and movies?
… Act so logically?

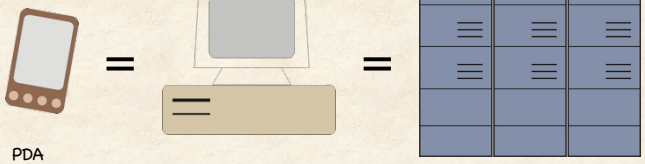
… Manipulate and remember data?
… Execute instructions?
… Access data quickly?

… Run multiple programs simultaneously?
… Store data permanently?
… Send messages?

… Find web pages?

Big Idea #1: Universal Computing Device

All computers (given enough time and storage) are capable of computing exactly the same things



PDA

Workstation

Supercomputer or Server Farm

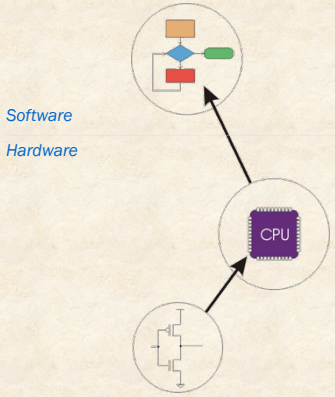
From Theory to Practice

Theory: Computer can *compute* anything that's possible to compute (given enough time and storage)

Practice: S*olving problems* involves computing under constraints

- Time
  - weather forecast, next frame of animation, …
- Cost
  - cell phone, automotive engine controller, …
- Power
  - cell phone, handheld video game, …

Big Idea #2: Abstractions Hide Complexity



Application Program

Algorithms

Language

Software

Hardware

Instruction Set Architecture (and I/O Interfaces)

Microarchitecture

Circuits

Devices

Abstractions Hide Complexity

How do we solve a problem using a computer?

Sequence of transformations between layers of abstraction

Stated using "natural language"

Problem

Software Design: choose algorithms and data structures

Step-by-step procedure

Algorithm

Programming: use language to express design

Express algorithm w/ computer language

Program

Compiling/Interpreting: convert language to machine instructions

Set of instructions computer can perform

Instr Set Architecture

More and more layers...

Instr Set Architecture

Processor Design: choose structures to implement ISA

Organization of a processor implementation

Microarch

Logic/Circuit Design: gates and low-level circuits to implement components

Layout and combination of gates (and, or, not, nor, nand)

Circuits

Process Engineering & Fabrication: develop and manufacture lowest-level components

Properties of materials, manufactuability

Devices

Example: Many Choices at Each Level

Solve a system of equations

Red-black SOR | Gaussian elimination | Jacobi iteration | Multigrid

Scratch | FORTRAN | C | C++ | Java

Sun SPARC | Intel x86 | Compaq Alpha

Pentium II | Pentium III | AMD Athlon

Ripple-carry adder | Carry-lookahead adder

CMOS | Bipolar | GaAs

Tradeoffs: cost performance power (etc.)

How do computers… Represent data?

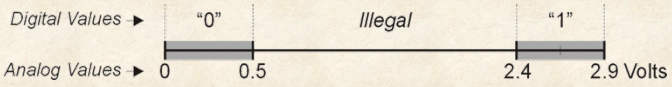
Lowest level: modern computer = electronic machine

- Works by controlling the flow of electrons

Easy to recognize two conditions:

- presence of voltage – state "1"
- absence of voltage - state "0

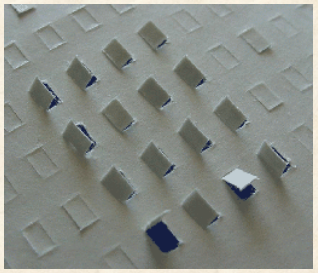
More difficult to detect and control analog values



Representing Data

Recognize this photo?

Was a vote cast or not?



Modern Computers = Binary Digital Systems

Digital system (not analog)

- finite number of symbols

Binary (base two) system:

- has two states: 0 and 1

Basic unit of information is *binary digit*, or *bit*

- Can be represented in any technology with two states

1 1 1 1 1

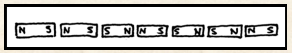
Bits in Computers

Transistors and wires: electrons flowing or not?

Capacitors and memory: holding a charge or not?

Optical CD-ROMs and DVDs: Reflecting or not?

Hard disk drive: Magnetized north or south?



How Can Bits represent Data?

Everything in computer is represented with 1 and 0

- All text you see or type, movies you watch, music you listen to
- Everything stored on disk, CD, or flash drive
- Everything you send between computers
  - Email, web pages
- Even instructions computer uses to run programs

What kinds of data must bits represent?

Logical: True, False

- Straight-forward: Two states
- True: 1, False: 0

Numbers

- Signed, unsigned, integers, floating point, complex, rational, irrational, …

Text

- Characters, words, strings, …

Images

- Pixels, colors, shapes, movies …

Sound

Instructions

Unsigned Integers

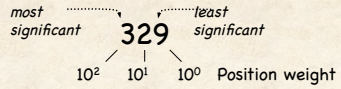
Approach 1: Non-positional notation

- Represent a number ("5") w/ string of ones ("11111")
- Problems?

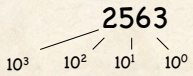
Unsigned Integers: Weighted Positional Notation

- Position and base determines value of symbol
- Example: Decimal numbers (base-ten)
  - Base-ten implies digit can be one of 10 different symbols: 0, 1, 2, 3, 4, 5, 6, 7, 8 and 9

Base-10



$3x100 + 2x10 + 9x1 = 329$



$2x1000 + 5x100 + 6x10 + 3x1 = 2563$

Unsigned Integers: Weighted Positional Notation

- Same properties hold for binary numbers (base-two)
- Base-two: Each digit holds two different symbols: 0 or 1

Base-10

329

10^2 10^1 10^0

3x100 + 2x10 + 9x1 = 329

Base-2

most significant 101 *least significant*

2^2 2^1 2^0

1x4 + 0x2 + 1x1 = 5 (decimal)

2563

10^3 10^2 10^1 10^0

2x1000+ 5x100 + 6x10 + 3x1 = 2563

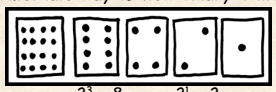
most significant 1011 *least significant*

2^3 2^2 2^1 2^0

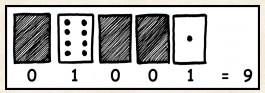
1x8 + 0x4 + 1x2 + 1x1 = 11 decimal

Converting Binary to Decimal

Alternate way to view binary numbers



What would 01001 be in decimal?



00011?

2 + 1 = 3

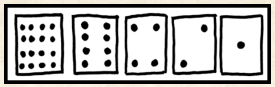
10001?

16 + 1 = 17

11111?

16 + 8 + 4 + 2 + 1 = 31 (also, 32 – 1)

Converting Decimal to Binary



How would you make decimal 5 in binary?

4 + 1 → 00101

16?

16 → 10000

21?

16 + 4 + 1 → 10101

How many different numbers can you make with 5 cards?

$2 * 2 * 2 * 2 * 2 = 2^5 = 32$

What is the largest number you can make with 5 cards?

$2^5-1 = 31$

What is the largest number you can make with N cards?

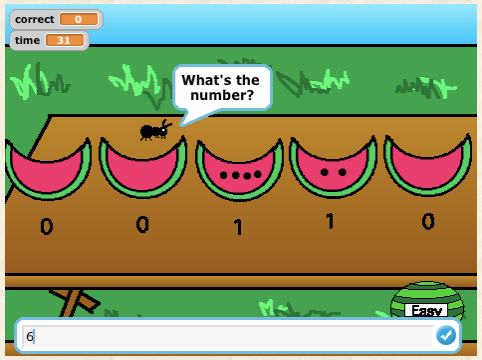
2^N-1

Counting in Binary

Binary	Decimal
000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7

What do you notice about bits in binary number?

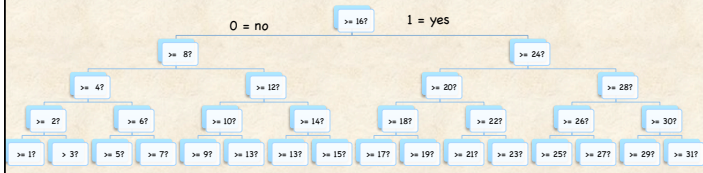
Practice Available with Scratch Game



Decision Tree for 0..31

How many questions needed to find answer between 0 and 31 (32 numbers)?

If 0 represents "no" or "false" and 1 represents "yes" or "true" what do you notice about the answers?



What is the height of this tree? (i.e. how many questions?)

5 questions for 32 numbers; $\log_2(32) = 5$

How many bits are needed to represent number between 0..31?

5 bits

Other Useful Units

Bytes

- Collection of 8 bits: 1101 0011
- How many different values represented in a byte?
  - $2^8 = 256$ values
- Abbreviation: Use B for Bytes vs. b for bits

Hexadecimal numbers

- Base-16
- Why might hexadecimal numbers be useful?
  - Two hex digits per byte
- What characters should we use?
  - 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, a, b, c, d, e, f
- 1101 0011 in hex is 0xd3

Today's Summary

Today's topics

- All computing devices equivalent given sufficient time and storage
- Abstractions of lower layers hide complexity
- Bits: Two states (on vs. off, true vs. false)
- Represent unsigned numbers with binary numbers
  - N bits can represent 2^N different values

Reading

- Pages 130-138 of "Invitation to Computer Science"

Announcements

- Play binary number game
- Homework 3 due Friday