

CS 149D, slide set 1

M. Overstreet
Old Dominion University
Spring 2006

How to make an A in this class

- Do the homework
 - Some homework will be taken up, some not
 - Sometimes it will be graded
 - Sometimes you get full credit for turning it in (i.e., we may be too lazy to grade it)
- Ask questions
 - Or get someone else to
 - Almost always, if you don't understand something, many others in the class don't either. Don't let the instructor (me!) get away with this.
 - Talk to other class members! Often they can help.
- Go over your exams carefully after they're returned
 - Possible grading errors. Don't let me get away with this!
 - Lazy instructors (me) often reuse questions on finals
- More the bad news:
 - Come to class
 - Read the book

CS 149D 2

Class Overview

- Class web site: www.cs.odu.edu/~cmo
 - Syllabus, tentative schedule, etc.
 - Old exams, Old assignments, more.
 - Copies of slides
- Will only cover parts of text
- Programming is part of the class
 - C++
 - Some people think programming is fun, some people hate it!
 - In this class, you found out which kind you are (if you don't already know).

CS 149D 3

First Quiz

- Send it to me e-mail so I will have your e-mail address
- Answer questions on next slide; your answers only
- Grade the correctness of your answers, include the number of correct & the number of incorrect answers
- Due before class, Jan. 16

CS 149D

4

Background questions

1. Do you own a computer? If, so which OS?
 - a. Some version of Windows
 - b. Linux
 - c. Mac (some version of X)
 - d. Palm OS
2. How do you connect to the internet
 - a. Work
 - b. ODU
 - c. Home—modem
 - d. Home—isdn
 - e. Home—cable
 - f. Home—other
3. Major?
4. If you are employed, how many hr/week?

5

Why study computers?

- They're everywhere, only more so
 - How many computers in this room?
 - When I came to ODU, the university had two, one for administrative use, one for academic use
- You may be able to use some of what we do in this class in your profession
 - New experimentation methodology is computation
 - Many basic sciences have significant computation component
- Knowledgeable citizens need to understand what computers can do to us!!
 - Is electronic voting a good idea?

CS 149D

6

Homework 1

- Check out the class web site
- Check out the text web site
- Read text, chapter 1 by Monday

CS 149D

7

What's Google Earth?

- Over the weekend, I was checking to see if it is available for Macs yet
- See very, very long discussion between Mac users and Windows users.
 - Each group knew for certain that people in the other group were:
 - Idiots: they wouldn't know a good computer if it bit them!
 - All good computer stuff was invented by Apple
 - All good computer stuff was invented by Microsoft
 - Who's right?
 - Don't forget: what's Google Earth?

CS 149D

8

Chapter 1

Data Storage

Chapter 1: Data Storage

- 1.1 Bits and Their Storage
- 1.2 Main Memory
- 1.3 Mass Storage
- 1.4 Representing Information as Bit Patterns
- 1.5 The Binary System
- 1.6 Storing Integers

CS 149D

10

Chapter 1: Data Storage (continued)

- 1.7 Storing Fractions
- 1.8 Data Compression
- 1.9 Communications Errors

CS 149D

11

Short history – and “bad” names

- First important applications of these machines were for doing complex, messy calculations
 - So they were called “computers”
- They are really “pattern manipulators”
- They add and multiply by having wires that start with patterns and produce other patterns
 - Input: patterns representing 2 numbers
 - Output: pattern representing their sum (or product)

CS 149D

12

Bits and their meaning

- **Bit** = Binary Digit = a symbol whose meaning depends on the application at hand.
- Some possible meanings for a single bit
 - Numeric value (1 or 0)
 - Boolean value (true or false)
 - Voltage (high or low)
 - Light or no light
 - Reflection or no reflection
 - etc., etc.
 - Somebody will come up with some new gadget that stores bits faster, cheaper, more reliably
 - (unrelated: saw an article yesterday that asserted that burned CDs and DVDs have a life of 2 to 5 years!)

CS 149D

13

Bit patterns

- All data stored in a computer are represented by patterns of bits:
 - Numbers
 - Text characters
 - Images
 - Sound
 - Anything else...

CS 149D

14

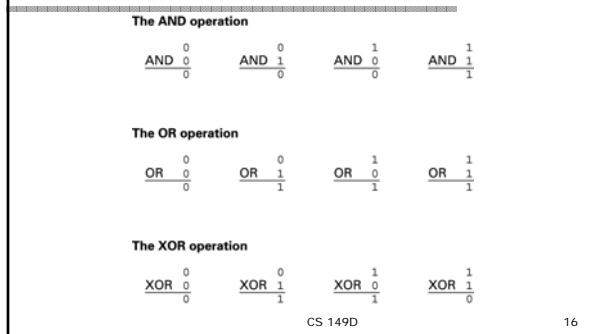
Boolean operations

- **Boolean operation** = any operation that manipulates one or more true/false values
 - Can be used to operate on bits
- Specific operations
 - AND
 - OR
 - XOR
 - NOT

CS 149D

15

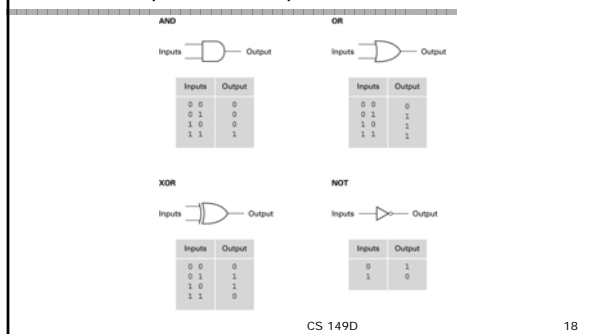
Figure 1.1 The Boolean operations AND, OR, and XOR (exclusive or)



Gates

- **Gates** = devices that produce the outputs of Boolean operations when given the operations' input patterns
 - Often implemented as electronic circuits
 - Provide the building blocks from which computers are constructed

Figure 1.2 A pictorial representation of AND, OR, XOR, and NOT gates as well as their input and output values



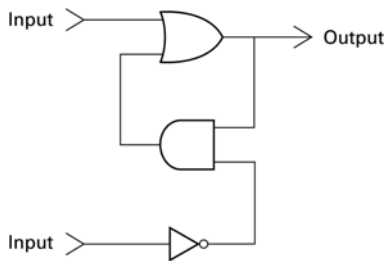
Flip-flops

- **Flip-flop** = a circuit built from gates that can store one bit of data.
 - Has an input line which sets its stored value to 1
 - Has an input line which sets its stored value to 0
 - While both input lines are 0, the most recently stored value is preserved

CS 149D

19

Figure 1.3 A simple flip-flop circuit

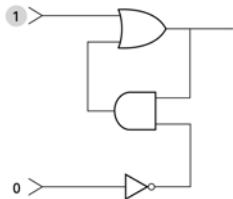


CS 149D

20

Figure 1.4 Setting the output of a flip-flop to 1

a. 1 is placed on the upper input.

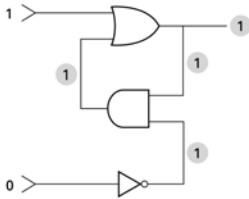


CS 149D

21

Figure 1.4 Setting the output of a flip-flop to 1 (cont'd)

b. This causes the output of the OR gate to be 1 and, in turn, the output of the AND gate to be 1.

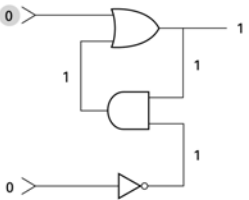


CS 149D

22

Figure 1.4 Setting the output of a flip-flop to 1 (cont'd)

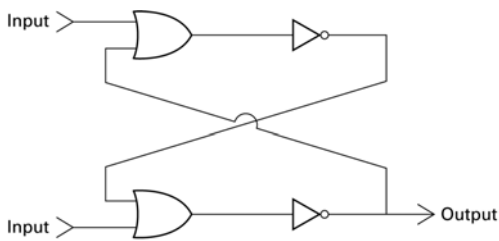
c. The 1 from the AND gate keeps the OR gate from changing after the upper input returns to 0.



CS 149D

23

Figure 1.5 Another way of constructing a flip-flop



CS 149D

24

Other storage techniques

- Dynamic memory – must be replenished periodically – Example: capacitors
- Volatile memory – holds its value until the power is turned off – Example: flip-flops
- Non-volatile memory – holds its value after the power is off – Example: magnetic storage
- Read-only memory (ROM) – never changes – Examples: flash memory, compact disks

CS 149D

25

Hexadecimal notation

- **Hexadecimal notation** = a shorthand notation for streams of bits.
 - Stream = a long string of bits.
 - Long bit streams are difficult to make sense of.
 - The lengths of most bit streams used in a machine are multiples of four.
 - Hexadecimal notation is more compact.
 - Less error-prone to manually read, copy, or write

CS 149D

26

Figure 1.6 The hexadecimal coding system

Bit pattern	Hexadecimal representation
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	A
1011	B
1100	C
1101	D
1110	E
1111	F

27

Figure 1.8 Memory cells arranged by address



CS 149D

31

Measuring memory capacity: Not quite like the metric system

- “Kilo-” normally means 1,000; but
Kilobyte = $2^{10} = 1024$
- “Mega-” normally means 1,000,000;
Megabyte = $2^{20} = 1,048,576$
- “Giga-” normally means
1,000,000,000;
Megabyte = $2^{30} = 1,073,741,824$

CS 149D

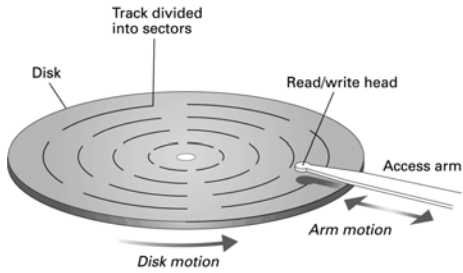
32

Mass Storage Systems

- Non-volatile; data remains when computer is off
- Usually much bigger than main memory
- Usually rotating disks
 - Hard disk, floppy disk, CD-ROM
 - Much slower than main memory
 - Data access must wait for **seek time** (head positioning)
 - Data access must wait for **rotational latency**

CS 149D

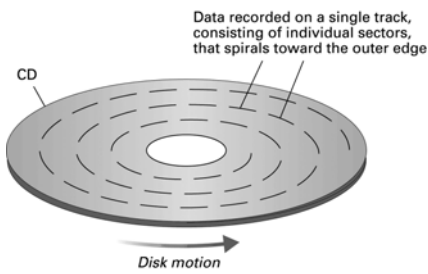
Figure 1.9 A disk storage system



CS 149D

34

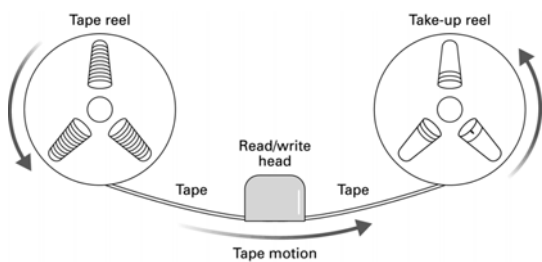
Figure 1.10 CD storage format



CS 149D

35

Figure 1.11 A magnetic tape storage mechanism



CS 149D

36

Files

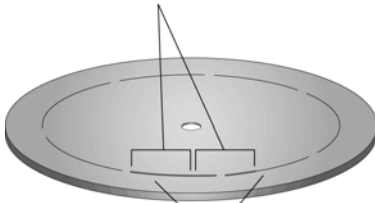
- **File** = the unit of data stored on a mass storage system.
 - **Logical record** and **Field** = natural groups of data within a file
- **Physical record** = a block of data conforming to the physical characteristics of the storage device.
- **Buffer** = main memory area sometimes set aside for assembling logical records or fields of a file

CS 149D

37

Figure 1.12 Logical records versus physical records on a disk

Logical records correspond to natural divisions within the data



Physical records correspond to the size of a sector

CS 149D

38

Figure 1.13 The message "Hello." in ASCII

01001000	01100101	01101100	01101100	01101111	00101110
H	e	l	l	o	.

CS 149D

39

Representing text

- Each printable character (letter, punctuation, etc.) is assigned a unique bit pattern.
 - ASCII = 7-bit values for most symbols used in written English text
 - Unicode = 16-bit values for most symbols used in most world languages today
 - ISO proposed standard = 32-bit values

CS 149D

40

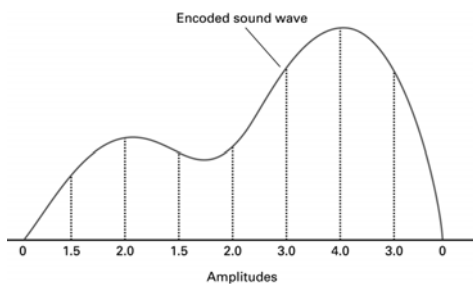
Representing numeric values

- Binary notation – uses bits to represent a number in base two
- Limitations of computer representations of numeric values
 - Overflow – happens when a number is too big to be represented
 - Truncation – happens when the correct pattern to represent a number is too big

CS 149D

41

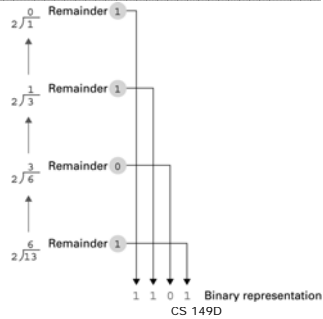
Figure 1.14 A possible sound wave represented by the sequence 0, 1.5, 2.0, 1.5, 2.0, 3.0, 4.0, 3.0, 0



CS 149D

42

Figure 1.18 Applying the algorithm in Figure 1.15 to obtain the binary representation of thirteen



46

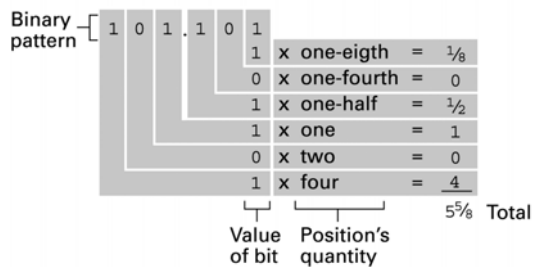
Figure 1.19 The binary addition facts

$$\begin{array}{r}
 0 \\
 +0 \\
 \hline
 0
 \end{array}
 \quad
 \begin{array}{r}
 1 \\
 +0 \\
 \hline
 1
 \end{array}
 \quad
 \begin{array}{r}
 0 \\
 +1 \\
 \hline
 1
 \end{array}
 \quad
 \begin{array}{r}
 1 \\
 +1 \\
 \hline
 10
 \end{array}$$

CS 149D

47

Figure 1.20 Decoding the binary representation 101.101



CS 149D

48

Back to patterns

- Kinda' arbitrary choices
- But when computers were first being designed and built, they were very expensive
- So patterns were chosen to make them a little less expensive
- Today, many of the choices seem poor, but we're stuck with them
 - They work OK.
 - No one wants to spend the effort (or money) to make better choices

CS 149D

49

Representing Integers

- Unsigned integers can be represented in base two
 - Why not base 10? **Ans.** Because it's a little cheaper to build the hardware for base 2
 - Due to basic physics
 - Good for torturing students
- Signed integers = numbers that can be positive or negative
 - Two's complement notation = the most popular representation
 - Excess notation = another less popular representation

CS 149D

50

Figure 1.21 Two's complement notation systems

a. Using patterns of length three

Bit pattern	Value represented
011	3
010	2
001	1
000	0
111	-1
110	-2
101	-3
100	-4

b. Using patterns of length four

Bit pattern	Value represented
0111	7
0110	6
0101	5
0100	4
0011	3
0010	2
0001	1
0000	0
1111	-1
1110	-2
1101	-3
1100	-4
1011	-5
1010	-6
1001	-7
1000	-8

CS 149D

51

Figure 1.22 Coding the value “-6” in two’s complement notation using four bits

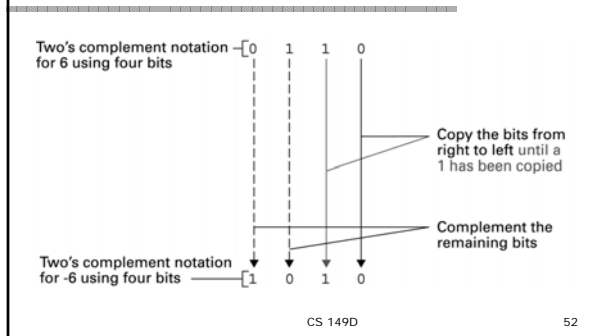


Figure 1.23 Addition problems converted to two’s complement notation

Problem in base ten		Problem in two's complement		Answer in base ten
$\begin{array}{r} 3 \\ + 2 \\ \hline \end{array}$	→	$\begin{array}{r} 0011 \\ + 0010 \\ \hline 0101 \end{array}$	→	5
$\begin{array}{r} -3 \\ + -2 \\ \hline \end{array}$	→	$\begin{array}{r} 1101 \\ + 1110 \\ \hline 1011 \end{array}$	→	-5
$\begin{array}{r} 7 \\ + -5 \\ \hline \end{array}$	→	$\begin{array}{r} 0111 \\ + 1011 \\ \hline 0010 \end{array}$	→	2

CS 149D 53

Figure 1.24 An excess eight conversion table

Bit pattern	Value represented
1111	7
1110	6
1101	5
1100	4
1011	3
1010	2
1001	1
1000	0
0111	-1
0110	-2
0101	-3
0100	-4
0011	-5
0010	-6
0001	-7
0000	-8

CS 149D 54

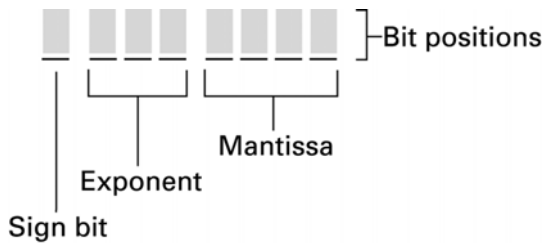
Figure 1.25 An excess notation system using bit patterns of length three

Bit pattern	Value represented
111	3
110	2
101	1
100	0
011	-1
010	-2
001	-3
000	-4

CS 149D

55

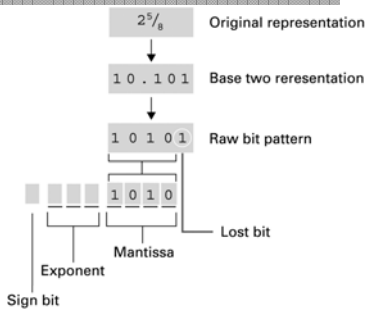
Figure 1.26 Floating-point notation components



CS 149D

56

Figure 1.27 Coding the value $2 \frac{5}{8}$




CS 149D

57

Figure 1.28 Decompressing xyxxyzy (5, 4, x)

a. Count backward 5 symbols.

x y x x y z y


b. Identify the four-bit segment to be appended to the end of the string.

x y **x x y z** y

c. Copy the four-bit segment onto the end of the message.

x y **x x y z** y **x x y z**

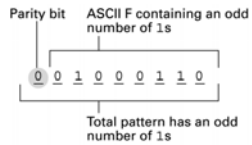
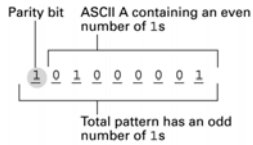
d. Add the symbol identified in the triple to the end of the message.

x y x x y z y x x y z **x**

CS 149D

58

Figure 1.29 ASCII codes for "A" and "F" adjusted for odd parity



CS 149D

59

Figure 1.30 An error-correcting code

Symbol	Code
A	000000
B	001111
C	010011
D	011100
E	100110
F	101001
G	110101
H	111010

CS 149D

60

Figure 1.31 Decoding the pattern 010100 using the code in Figure 1.30

Character	Distance between the received pattern and the character being considered
A	2
B	4
C	3
D	1 <i>Smallest distance</i>
E	3
F	5
G	2
H	4

CS 149D

61

Homework (list also on web)

- Due in class Jan. 25.
 - Pg. 24: 1, 4, 5, 6, 7; Pg. 34: 2, 3, 4
- Due in class Jan. 30.
 - Pg. 40: 1, 5, 6, 10
- Due Feb. 1
 - Pg. 45: 3, 4, 5; Pg. 51: 1, 2, 5
- Due Feb. 6
 - Pg 56: 1, 2, 3; Pg. 65: 1, 5

CS 149D

62

Diversion: Lab 1

- Alice chat bot
 - What's a "bot"?
- Assignment: find Alice on the web and have a conversation with her
 - Mail me your impressions after talking with her
 - See class web site for questions to answer.
 - Figure out if she remembers earlier parts of a conversation.
 - Check out the Chatterbox Challenge. Who won last year?
- Computer science frustration for teachers
 - Stuff in movies often much better than what is possible
 - Remember 2001: A Space Odyssey (1968)?
 - HAL figured out how to read lips
 - He had to WANT to do this
 - Alice can't read lips. Alice really does not exhibition volition.
- Due in recitation Jan. 30. Handholding provided.

CS 149D

63
