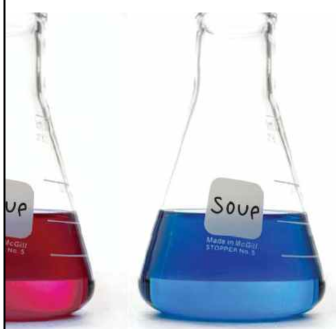
soup & science

McGill
Faculty of Science
Office for Undergraduate Research in Science



Learn about cutting-edge research over lunch with cool profs

September 12 - 16, 2011
11:30 AM
Redpath Museum

More info:
www.mcgill.ca/science/

COMP 102: Excursions in Computer Science

Lecture 2: Bits&bytes, Switches, and Boolean Logic

Instructor: Joelle Pineau (jpineau@cs.mcgill.ca)

Class web page: www.cs.mcgill.ca/~jpineau/comp102

Computation

- What is computation?
- What is the basic unit of computation?
- What is the link between computation and information?
- What is the smallest unit of information?

The lowly bit

What is the smallest unit of information?

- Chemistry has its molecules.
- Physics has its strings.
- Computer science has its bits:
  - True / False
  - On / Off
  - 1 / 0
- Think of it as a switch: 

Recall

- The vacuum tube:

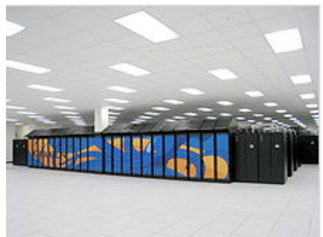
- The transistor:

- These are electronic on/off switches.
- The difference engine used mechanical on/off switches (think "lever").

What's a Bit?

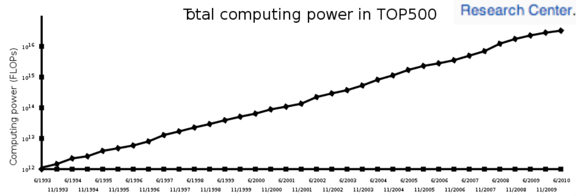
- Word "Bit" is a contraction of "Binary digit"
- What's a **binary digit** ?
  - Base 10: In decimal number system, a digit can be any of the ten values 0, …, 9
  - Base 2: In binary number system, a digit can be any of the two values 0, 1
- Bits are nice because they are:
  - **Simple*:*** There's no smaller discrete distinction to be made.
  - **Powerful**: Sequences of bits can represent seemingly anything.

Putting the bits together

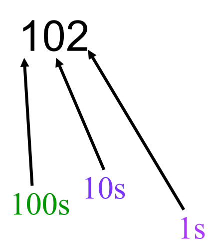


Kraken, a Cray XT5 supercomputer at Oak Ridge National Laboratory

The Columbia Supercomputer, located at the NASA Ames Research Center.



Representing numbers



- Decimal System uses 10 digits: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
  - >> Also called: Base 10
- Position of a digit interpreted to give the value

Representing numbers: Decimal system

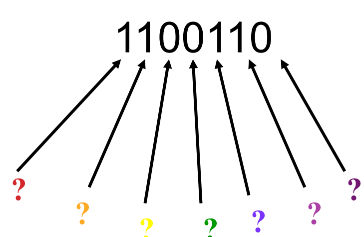
- 102 = 1 x 100 + 0 x 10 + 2 x 1

  = 1 x 10^2 + 0 x 10^1 + 2 x 10^0

- 1 decimal digit produces 10 distinct values
- 2 decimal digits produce 100 distinct values
- 3 decimal digits produce 1000 distinct values

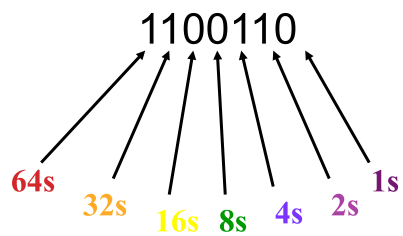
- *n* decimal digits produce 10^n distinct values

Representing numbers: Binary system



- Decimal System uses 2 digits: 0, 1
- Base 2

Representing numbers: Binary system



- Binary system uses 2 digits: 0, 1
- Base 2

Representing numbers: Binary system

$1{,}100{,}110_2 = 1 \times 64 + 1 \times 32 + 0 \times 16 + 0 \times 8 + 1 \times 4 + 1 \times 2 + 0 \times 1$

$= 1 \times 2^6 + 1 \times 2^5 + 0 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0$

- 1 binary digit produces 2 distinct values
- 2 binary digits produce 4 distinct values
- 3 binary digits produce 8 distinct values

- *n* binary digits produce 2^n distinct values

Binary Numbers in Computing

- Easy to make fast, reliable, small devices that have only 2 states.

- 1/0 represented by
  - hole/no hole in punched card
  - hi/low voltage (memory chips)
  - light bounces off/light doesn't bounce off (CDs/DVDs)
  - magnetic charge present/no magnetic charge (disks)

Measuring Data

We can group number of binary digits and refer to the group sizes by special names:

- 1 bit(b) = 2^1 = represents 2 different values
- 1 byte(B) = 8 bits = 2^8 = 256 values
- 1 kilobyte(KB) = 1024 bytes = 2^{10} bytes
- 1 megabyte(MB) = 1024 KB = 2^{20} bytes
- 1 gigabyte(GB) = 1024 MB = 2^{30} bytes
- 1 terabyte(TB) = 1024 GB = 2^{40} bytes

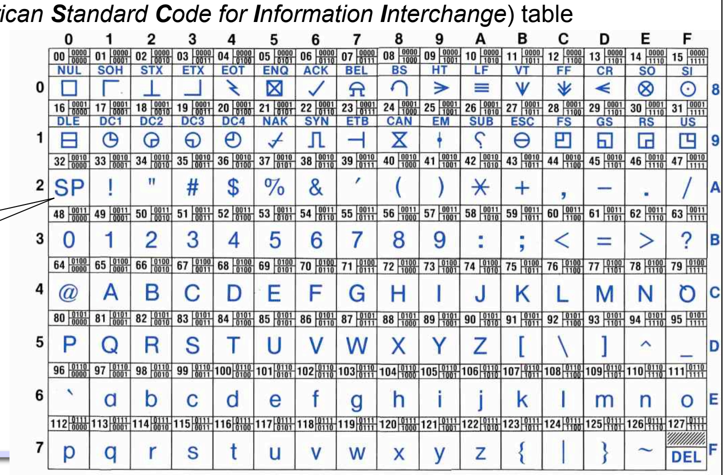
Combining bits to represent complex information

- Remember a bit can only be 0 or 1.
- We can combine multiple bits to represent more complex data.
  - Text
  - Images
  - Sound
  - Video
  - Etc.

Representing Text

- Each character is encoded using 1 byte
- ASCII (***A**merican **S**tandard **C**ode for **I**nformation **I**nterchange*) table

Space: “ ”



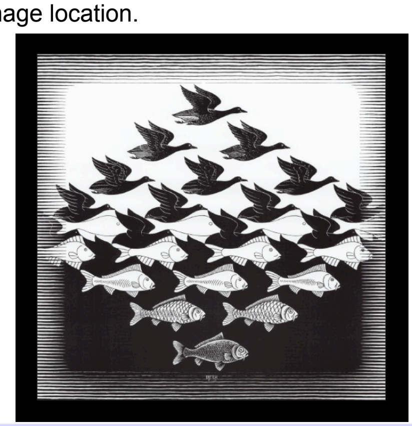
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
0	00 NUL	01 SOH	02 STX	03 ETX	04 EOT	05 ENQ	06 ACK	07 BEL	08 BS	09 HT	10 LF	11 VT	12 FF	13 CR	14 SO	15 SI	8
1	16 DLE	17 DC1	18 DC2	19 DC3	20 DC4	21 NAK	22 SYN	23 ETB	24 CAN	25 EM	26 SUB	27 ESC	28 FS	29 GS	30 RS	31 US	9
2	32 SP	33 !	34 "	35 #	36 $	37 %	38 &	39 '	40 (	41)	42 *	43 +	44 ,	45 –	46 .	47 /	A
3	48 0	49 1	50 2	51 3	52 4	53 5	54 6	55 7	56 8	57 9	58 :	59 ;	60 <	61 =	62 >	63 ?	B
4	64 @	65 A	66 B	67 C	68 D	69 E	70 F	71 G	72 H	73 I	74 J	75 K	76 L	77 M	78 N	79 O	C
5	80 P	81 Q	82 R	83 S	84 T	85 U	86 V	87 W	88 X	89 Y	90 Z	91 [	92 \	93]	94 ^	95 _	D
6	96 `	97 a	98 b	99 c	100 d	101 e	102 f	103 g	104 h	105 i	106 j	107 k	108 l	109 m	110 n	111 o	E
7	112 p	113 q	114 r	115 s	116 t	117 u	118 v	119 w	120 x	121 y	122 z	123 {	124 |	125 }	126 ~	127 DEL	F

Representing Text

“M	A	R	C”
1st byte	2nd byte	3rd byte	4th byte
77	97	114	99
01001101	01100001	01110010	01100011

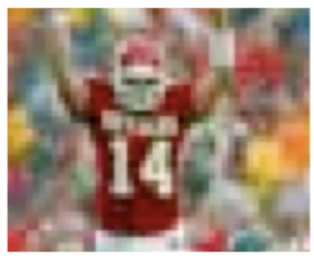
Almost everything can be represented with bits

- Escher’s drawing:
  - Use one bit to represent the colour (black=0, white=1) at each particular image location.



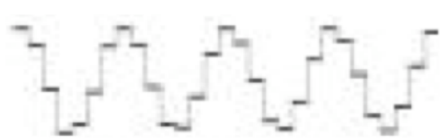
Almost everything can be represented with bits

- Digital images:
  - A **group of bits** represents the colour at each particular image location: we call this a **pixel**.
  - An image pixel is one of Red, Blue or Green.
    - How do we encode this information with bits?
    - How many bits do we need?



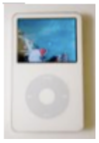
Almost everything can be represented with bits

- Digital sound:
  - Average sound intensity (= a number) over a short time interval is represented using a group of bits.



Modern technologies need lots of bits!

- Consider the iPod: 160Gb.
  - 1Gb = one billion bytes (1 byte = 8 bits).



- Sound:
  - 128Kbps of sound (Kbps = Kilobits per second).
  - So 160,000 minutes of sound, or 40,000 songs (at 4min. per song).
- Screen:
  - 1.5Mbps of video (Mbps = Megabits per second) + 128Kbps of sound.
  - So 237 minutes of video without sound, or 218 minutes of video with sound.
  - For the video: 320x240(=76,800) pixels
    x 1 byte per pixel for colour (=Red-Blue-Green)
    x 30 frames per second = 55.3 Mbps (That's a lot! We'll explain this later)

Logical variable

- Bits are not just for sound and images.
- Bits can store logical variables.
- A logical variable is something that we can imagine as being True or False.
  - *TodayIsThursday* = False
  - *ItIsDarkOutside* = False
  - *IAmTeachingCOMP102* = True
- *TodayIsThursday*, *ItIsDarkOutside* and *IAmTeachingCOMP102* are logical variables. They can be True or False.
- Logical variables are also sometimes called Boolean variables.

And, Or, Not

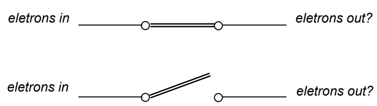
- Logical variables can be combined with logical operations.
- The most important logical operations are **AND**, **OR**, and **NOT**.
  1. *x* **AND** *y* is True only if both *x* is True and *y* is True.
  2. *x* **OR** *y* is True if either *x* or *y* are True.
  3. **NOT** *x* is True only if *x* is False.
- Logical operations have the intuitive English meaning.

Logical expressions

- Logical expressions combine logical variables and logical operations into more complex expressions.
  - *IAmTeachingCOMP102* **AND** *ItIsDarkOutside* = False
  - **NOT** *ItIsDarkOutside* = True
  - *IAmTeachingCOMP102* **OR** *TodayIsThursday* = True
  - (*TodayIsThursday* **OR** *IAmTeachingCOMP102*) **AND** (*IAmTeachingCOMP102* **AND** (**NOT** *ItIsDarkOutside*)) = ??

Implementing logic

- How do we implement logical variable?
  - Easy! One switch per logical variable



Closed switch = True

Open switch = False

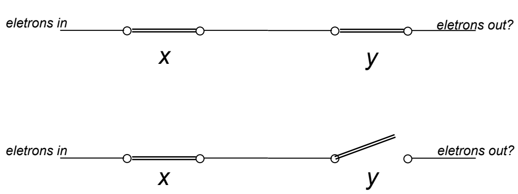
- How do we implement logical operations?

- How do we implement logical expressions?

Implementing logical operations

Key Idea: Combining switches

- **AND** operation: Combine switches in <u>series</u>

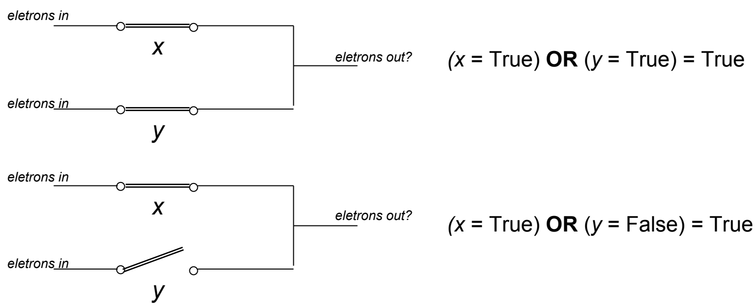


(*x* = True) **AND** (*y* = True) = True

(*x* = True) **AND** (*y* = False) = False

Implementing logical operations

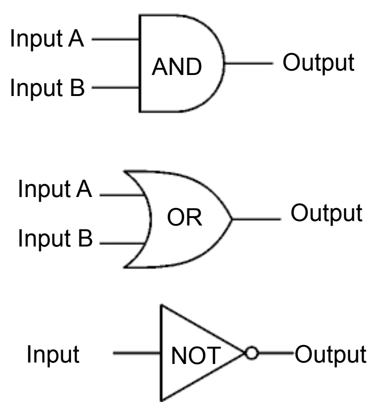
- **OR** operation: Combine switches in parallel



Implementing logical operations

- **NOT** operation is slightly more complicated.

- Use abstract "gates":



Implementing logical expressions

- Combine multiple switches.

E.g. (*TodayIsThursday* **OR** *IAmTeachingCOMP102*) **AND**

(*IAmTeachingCOMP102* **OR** *ItIsDarkOutside*) = ??

Practice example

- You are given the responsibility of building an automatic voting machine.
  - Assume there are 2 candidates.
  - Assume there are 3 voters, everyone gets a single vote.
  - The candidate with the most votes wins.

- What logical variables would you use?

- Can you write a logical expression, which evaluates who wins (True = Candidate A, False = Candidate B)?

Take-home message

- Understand the concept of a **bit**.
- Know how to combine multiple bits to represent complex information (text, images, sound, video).
- Understand what are **logical variables**.
- Know the three basic **logical operations**.
- Be able to evaluate **logical expressions**.

Final notes

- Office hours for TAs are posted on the course syllabus:
  http://www.cs.mcgill.ca/~jpineau/comp102/syllabus.html
- Assignments are due **in class, every Thursday**, starting Sep.8.
- Homework 1 is posted on the class schedule:
  http://www.cs.mcgill.ca/~jpineau/comp102/Homework/homework1.pdf
  - For the last question, you should go beyond the class notes, do a bit of research, and don't forget to cite your sources!