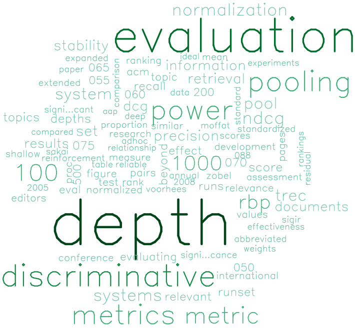
LBSC 690: Information Technology
Lecture 01
Computers and Data

William Webber
CIS, University of Maryland

Spring semester, 2012

Instructor



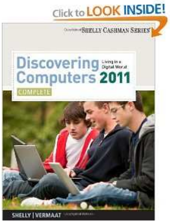
- William Webber, `wew@umd.edu`
- I finished my PhD in Computer Science at the University of Melbourne last year.
- I'm in the first year of my postdoc at the UMD iSchool
- My research wordcloud is to the left (last ten papers)

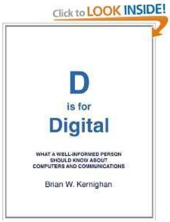
Contact methods

- Google group:
  `groups.google.com/group/lbsc690-spring12-0101`
  - Official announcements via this group
- Course website:
  `www.umiacs.umd.edu/~wew/teaching/690/spring12/`
- My email: `wew@umd.edu`
- My office: Hornbake 2123B

Consultations: 8:15pm to 9:15pm Tuesdays, directly after class.
NOTE: doors will be locked; please come up with me.
Consultations also by appointment.

Textbooks





- Gentle, broad, verbose introduction to computer technology
- Expensive ($\approx$ \$100)
- Older edition is ok (but check chapters)

- Concise introduction to computers and communication
- By a “famous name” in the history of computing
- Cheap ($\approx$ \$15)

Readings will be provided from both books. Buying one of them is recommended (but not compulsory).

Readings

- Background readings will be taken from one of the textbooks
- Primary readings will be drawn from free online sources
- Please read material for a week's class before coming to class

Assessment

- There will be a midterm and an end-of-term exam, worth 35% combined.
- There will be a single term project, worth 40%. This is a group project, to be done in groups of 3 (or 2 for rounding)
- Most weeks there will be homework, worth 15%.
- Participation (in classes and on the Google group) is worth 10%.

Excursus: enough about me; what about you?

The continuity of reality

- At a macro level, reality is continuous (though at a quantum level, it is quantised)
- There is an indefinite divisibility in the dimensions of reality (until we reach the quantum level):
  - Length, width, and breadth
  - Loudness of sound
  - Brightness of light
  - Tones of color
  - Flow of time
  - Physical location

Analogue representation

- When we come to represent reality (to copy it, store it, replay it, manipulate it, reason about it), a natural representation is likewise a continuous one.
- For instance, the soundwave of a performed music piece can be represented by the grooves in an LP record:
  - The louder the sound, the wider the groove oscillations
  - The higher the pitch, the more frequent the oscillations
  - Time tracks continually with the length of the groove
- The representation is a direct image or *analogue* of the original, like a seal in wax; but inevitable imperfections mean the representation departs from the original.
- The more times the representation is copied, the more the imperfections accumulate. Think (if you remember them) of copying from audio tape to audio tape.

Digital representation

- An alternative representation of reality is a discrete one: convert the continuous magnitudes of reality into discrete values in our representation.
- For instance, to represent a sound wave:
  - Quantize time. Only measure the wave at fixed intervals.
  - Quantize amplitude. Measure height (depth) of wave to a fixed precision
- Each discrete value can be recorded as a whole or integral number, written to a certain number of digits – hence, a *digital* representation
- Concatenate all these digital values together, and our representation (of a song, of a picture, of a video) is one (very long) integral number.

Digitised objects as numbers

So, for instance, the 4th movement of Beethoven's 9th Symphony on my laptop (for a particular performance, encoded in mp3) is, in part:

$$\ldots 38056584543680340 \ldots$$

whereas a photo of God creating Adam from the ceiling of the Sistine Chapel is, in part:

$$\ldots 26521209259038976 \ldots$$

and the opening words of the Book of Genesis in the King James Version is:

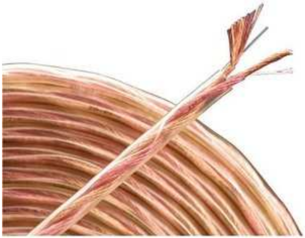
$$70676998972496260 \ldots$$

Why digital?

- A digital representation is a simplification of reality; we round off (sample and quantize).
- But once an object has been converted to a number, it can be copied indefinitely with no loss of accuracy (provided we can copy the number correctly!).
- The one representation (a number) can be used for all forms of objects (sound, video, text) . . .
- can be stored on the same formats (DVDs, hard disks) . . .
- transferred on the same communication channels (fiber networks, satellite) . . .
- and processed by the same object (a digital computer)

Excursus: expensive audio components

- This HDMI cable costs $120 retail.
- Will performance differ from a $5 HDMI cable?
- Is it worth it?



- This speaker cable costs $80 retail.
- Will performance differ from $5 speaker wire?
- Is it worth it?

Where does component quality matter in a digital hi-fi system?

Representing numbers as states

- A whole number takes on one of a discrete number of values
- For instance, a single-digit decimal number takes on one of 10 values: $0, 1, 2, \ldots, 9$
- To represent this number physically, we need a physical object that is able to take on that many states.
- For instance, a six-sided dice can be in one of six states

Up/down, on/off

- The easiest (useful) number of states to represent is 2, for the values 0, 1.
  - Why is a representation holding a single state not useful?
- For instance:
  - a high voltage can be 1, a low voltage 0
  - an electrical switch being on can be 1, being closed 0
  - presence of a magnetic field on a tape can be 1, absence 0
  - a pit on an aluminium-covered surface (e.g. on a DVD) can be 1, the absence can be 0.
- This gives us the *binary* encoding system

Binary numbers

- To represent a number greater than 9 in the decimal system, we add (decimal) digits.
- Similarly, to represent a number greater than 1 in the binary system, we add (binary) digits or *bits*.
- In a b-base system, digit x in place n has value $x * n^{(b-1)}$:

Decimal: 235

Place-value	100	10	1	
Digit	2	3	5	
Value	200	+ 30	+ 5	= 235

Binary: 11101011

Place-value	128	64	32	16	8	4	2	1	
Digit	1	1	1	0	1	0	1	1	
Value	128	+ 64	+ 32	+ 0	+ 8	+ 0	+ 2	+ 1	= 235

Excursus: conversions and counting

From binary to decimal:	From decimal to binary:
10	3
101	10
01011	33
100001	63

- What number can you count up to on your fingers, in binary?
- The Chinese have a method for counting up to ten using the five fingers on one hand. Assuming fingers have only two states (up, down), is five the minimum number of fingers you need to count up to ten?
- If you also used your toes, what number could you count up to in binary?

Binary thoughts

- Not only are data (images, sound, text, numbers) represented as binary numbers in a digital computer.
- Also, the instructions to the computer (the computer program) is represented as a sequence of binary numbers.
- For instance, this binary number:

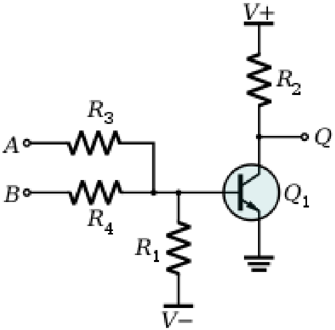
  00000000001000100011000000100000

  tells a MIPS processor (as might be found in a digital TV) to add numbers found in two registers (high-speed memory locations) and store the results in a third.

Logic gates

$$A \text{ NOR } B \quad \Leftrightarrow \quad \neg(A \vee B)$$

INPUT		OUTPUT
A	B	A NOR B
0	0	1
0	1	0
1	0	0
1	1	0



- Logical relations implemented in simple (transistor) circuits.
- Combined to perform more complex (math) operations.
- The above circuit, repeated often enough, is sufficient to implement the arithmetic-logical unit of digital computer.

Excursus: transferable intelligence

- A program's data ("memory") and its code ("reason") can be represented as a bitstream (or, if you prefer, a number)
- This representation (collection of bits) is independent of, transferable between, different physical implementations (collection of atoms).
- What implications does this have for understanding, future of, human consciousness?
- Is our cranial wetware reducible to software running on hardware?

Bits, bytes, words

10100010 01100010 00010010 000000001

- A binary digit is called a *bit*
- Bits are organized into groups called *bytes*
  - Now, a byte always has 8 bits, but historically this varied
- Bytes in turn are organized into *words* of 2, 4, or 8 bytes

Word size

- The word size of an architecture is the “standard” processing unit
- Architectures are identified by their word size, e.g. 32 bit, 64 bit
- Size of word crucially determines maximum memory size of architecture

Technical details

- The processor accesses data in memory by using the address of that data.
- Each byte in memory has its own address.
- An address is simply a number, held in a machine word.

Excursus: memory space

- How much memory can a 16-bit machine hold?
- How much memory can a 32-bit machine hold?
- How much memory can a 64-bit machine hold?

Larger memory, storage sizes

		Size	
Prefix	Abbrev	Old	New
Kilo-	KB	2^{10}	thousand
Mega-	MB	2^{20}	million
Giga-	GB	2^{30}	billion
Tera-	TB	2^{40}	trillion
Peta-	PB	2^{50}	quadrillion

- Larger memory sizes traditionally organized into powers of 2 that are multiples of 10
- $2^{10n} \approx 10^{3n}$
- Modern usage is in powers of ten
  - A hard-disk manufacturer actually got sued over this issue!

Transfer speeds

- Network transfer speeds have the same prefixes, but are in *bits*, not *bytes*, per second.
  - E.g. 128 Kbps is 128,000 bits per second
- Divide by 8 to get bytes per second (which is what you want)
  - E.g. 128 Kbps is 16,000 bytes or 16 KB per second

Technology	Transfer speed
Modem	300 bps to 56 Kbps
(A)DSL	1.5 Mbps to 24Mbps down, 0.5 Mbps to 3.5 Mbps up
Cable	1 Mbps to 30 Mbps down, 128 Kbps to 768 Kbps up
Fibre to the X	100Mbps to 1 Gbps and beyond

Figure: Transfer speeds of different network technologies.

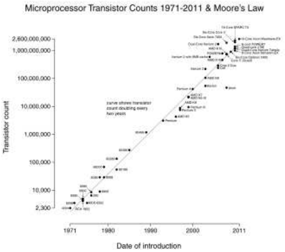
Excursus: what does "network speed" mean

- How fast does a bit of information travel?
- How long does it take a bit to travel the 20,000 km from one side of the earth to the other?
- What really determines network speed?
- Why is the term "bandwidth" used in relation to speed?

CPU speed

- Work of CPU driven by “system clock”; each “tick” signals an operation
- CPU speed is measured in terms of clock cycles (ticks) per second, or hertz (hz).
  - For instance, my laptop has 2 processors running at 2.4GHz.
  - How many ticks per second is that?
- Raw clock speed increasingly unreliable as measure of effective system performance.

Moore’s law



- Number of transistors in integrated circuit (IC) doubles approximately every two years (or every 18 months)
  - Names after Gordon Moore, later Intel founder, who proposed the regularity in 1965
- Num. transistors approximates processing capacity of IC.
- Transistors “spent” on different things (CPUs, cache)
  - modern chips have many CPUs per chip, not higher clock speeds for single chip
- Various physical limitations (e.g. size of an atom, amount of heat generated) are approaching.

The memory mountain

Memory type	Size	Time to access	2 Ghz cycles waiting
Processor register	256 B	0.5 ns	1
L1 cache	64 KB	2ns	4
L2 cache	3 MB	10ns	20
Main memory	4 GB	100ns	200
Local hard disk	2 TB	10ms	20 million
Internet server	1 PB	100ms	200 million

- Processor speeds increasing quickly
- Memory size increasing quickly
- ...but memory access speed increasing only slowly

Excursus: waiting for your memory

- What is the hertz of your cardiovascular system?
- If your memory wait times were proportional in hertz to a modern CPU's, then:
  - How long would you wait to retrieve a memory from L2 cache ($\approx$ something you thought about recently)?
  - How long would you wait to retrieve a memory from your main memory ($\approx$ something you haven't thought about for a while)?
  - How long would you wait to retrieve a memory from local hard disk ($\approx$ something you had to look up in a book)?