

Day 14

`\startx`

TAKE - CPU, video and brick (images of CPU masks)

Attendance

Exercises - 1 from c3, write-out the first few terms and "look" for the pattern, humans have a strong visual system

Chapter 4 (contd)

Internal and external representations, room full of compute agents that speak a different language.

Internal types (primitive)

numbers - positive integers, negative integers, floating-point
characters - really numbers...

Internal types (complex)

files (of all sorts of types)

External types

numbers
characters
images
audio
video

smell

Why use binary?

Easy to distinguish between no voltage and voltage, much easier than say 10 different levels of voltage.

Shorter list of rules for adding, multiplying, etc.

Why use fixed size basic data types?

Much easier to work with fixed sized things rather than variable. At the lowest hardware levels there is no choice. At higher levels it is possible to have arbitrarily large numbers but the performance cost is very significant.

Very fast working with fixed size things, e.g. network packets. Similar hardware/software for manipulating variable sized things much more complex and typically slower.

Pattern alert! (like time/space tradeoff, 80/20 rule)

Consequences

Overflow, underflow, etc. are problems.

Code Mappings

Lookup tables to go from external <-> internal

ASCII, Figure 4.3 (8 bit maximum, 1 byte, standard "unit"), 0-255, 256 values

UNICODE (16 bit, lots more space (show))

Sound and Images

Analog to digital translation, "digitizing" or "sampling"

Sound example with diagram - amplitude (loud soft), period (high low pitch), frequency. BENG or Fitz knows more than I do.

Picture example - first break into pixels (cells), color is continuous, map it to RGB values

Reflecting light off the surface and then detecting reflection (scanning)

Lots of sampling works this way (radar, sonar, scanning)

Just capture light (CCD lenses in video cameras)

Sampling rate affects quality and size (one up, other up)

Sound and images are /huge/ compared to text, compression (lossless if possible) is very important to both formats. One way to do lossless compression is to note changes and not repeat stretches of sameness.

Sidebar - (600 year timeline)

558 years of printed matter (1450 Johannes Gutenberg)

148 years of recorded voice (1860, French, on paper no playback;

Edison 17 years later with playback)

50 years of recorded video (1956, \$10K, by 1966 \$K and "home" use)

30 years ago just text and number,

20 years ago image and sound digitation possible but hard/expensive,

10 years ago pretty easy to get it in but mostly record/play only

Now look at what people can build GIYF:"snek pek" first link YouTube

Accelerating pace, sometimes caused by a ...

Virtuous cycle - one good thing leads to another (economists)

Now it's the Internet <-> user generated content

And we're increasing the percentage of people in the world that can get to these technologies and content, albeit slowly

Which is one type of feedback loop, where the output depends on the input and the current output.

Bistable environments as basis for all this - reliable and fast, makes it worth all the work of converting from external <-> internal

Transistors

Switch, can be built-up through series of abstractions into powerful devices. Turned on and off electrically.

Early computing devices, say Bombs at Bletchley Park, were also built out of switches. Same basic principle but different underlying implementation technology.

THIS IS A REALLY BIG DEAL. The notion that you can separate what you are doing from how you are doing it. Allows algorithms and the computing systems we build with them to evolve.

Very small, ~50M per CM². Moore's law doubles this every ~18 months

Show Stephen Wheat's slides

We're in the midst of the transistor age, but it's likely that something else will replace it during your lifetime.

How switches can be made into Boolean logic, which in turn can do all

kinds of things.

AND gate - three transistors, third has power from one and switch from the other

Review what's coming next and For Mon, Wed, and Fri

Boolean logic and the gates (small programs) we build with them
Abstraction!

Collect and stragglers and Lab 6

For Monday

Read 4.3 (Boolean logic and gates) and 4.4.1 (Building computer circuits)

For Wednesday

Read 4.5

Exercises 1, 2, 3, 4, 9, 10, 11, 14 (come w qs on mon)

For Friday

Lab <something>

\stopX