

Lecturer: John Morrison (prof)

webpage: cuc.ucc.ie/jpm/CS1110

Will go from the idea of representing information with numbers up to building computers. (something like that, I kind of missed it)

Lab sessions start in a week or two.

20% Continuous Assessment

80% Final Exam (in December)

4 questions on the final exam, 20(/100) marks per question

He draws notes on a tablet. They'll go up in pdf form.

Overview:

- representing information with numbers
- number bases
- Boolean logic/algebra
- digital design
- combinatorial circuit design
- sequential circuit design [maybe]
- various hardware components (e.g. memory, disk, I/O) [maybe]
- hardware organisation

Core Concept:

All information can be modelled by numbers, together with some appropriate conventions.

examples:

- each letter in the alphabet and extended characters can be assigned a number.
- music such as audio files in an mp3 file where the numbers correspond to voltage values. You can sample an audio wave and then reproduce it. You can process the numbers before reproduction.
- visual information can be represented by dots on a screen, those dots can be represented by numbers. For example: x-position; y-position; colour.

Which numbers?

More accurately, which number base?

In our everyday life we use base 10.

In this system there are ten symbols: 0; 1; 2; 3; 4; 5; 6; 7; 8; 9.

When counting we use what's called a Positional Counting System.

...	100s	10s	units
			0
			...
			9

In this system, when you run out of symbols, you reuse them in the next column.

The units column changes every 10^0 times, the 10s column every 10^1 times, and so on.

Base 10 is 'natural' for humans. Is it natural for a machine? If a machine uses base 10, it has to manipulate 10 different 'symbols', which is a lot. Requires complex stuff.

Today's machines use base 2 (binary). We can use any two symbols, so 0 and 1 are used because they're familiar/easy.

...	2^2	2^1	units	(base 10)
	0	0	0	0
	0	0	1	1
	0	1	0	2
	0	1	1	3
	1	0	0	4

Numbers produced using any positional counting system can be directly compared, because they use the same system. for example, you can also count in base 3.

But the symbols (0, 1, ...) only mean something in the context of the base they're in.

Why do today's machines use base 2?

It comes out of the technology. It's easy to make switches. A switch is a component with two states – on and off. In this context, those switches are transistors.

Why don't we use base 2 for everything? We (humans) tend to like to use many different symbols to represent things, such as sounds, images, the alphabet. We are good at recognising visual images, sounds, etc.

Computers don't have this ability, so we need to take everything that we normally use and represent it with numbers instead, using certain conventions. Then the computer can process that information.

By doing this we lose some information. We are limited in the number of digits a computer can hold/represent. Accuracy is lost in translating things for a computer.

So base 2 is a natural base for machines, because switches are easy.

Computers typically work with voltage levels. When a transistor is in the "off-state", its output is 0V, and when it is in the "on-state" the output is 5V.

We represent On with the binary value 1.

We represent Off with the binary value 0.

This is our first level of abstraction.

Now instead of talking in terms of voltages, we can talk in terms of binary values. Then we can start manipulating these values, create designs for doing things, and turn those designs into the hardware level of the computer.