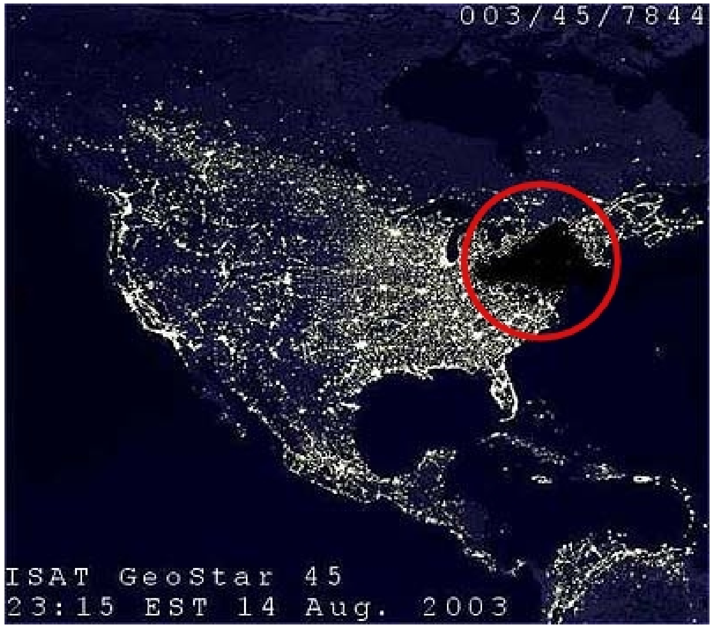
Does your computer do what it's supposed to?

Proving Correctness of Computer Systems

Sara Adams

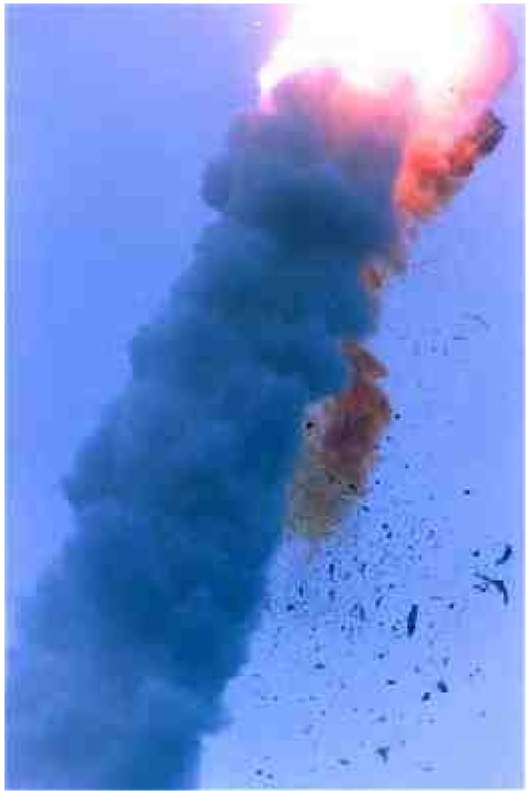
```
                              Windows

A fatal exception 0E has ocurred at 0028:C0011E36 in VXD VMM(01) +
00010E36. The current application will be terminated.

*   Press any key to terminate the current application.
*   Press  CTRL+ALT+DEL again to restart your computer. You will
    lose any unsaved information in all applications.

                        Press any key to continue _
```

Things can go terribly wrong



14 August 2003

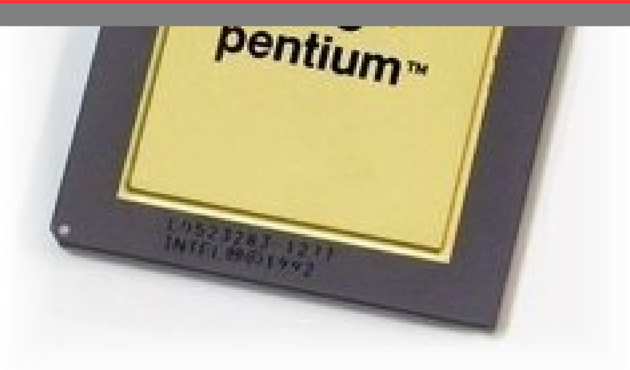
13 June 1994



4 June 1996

Things can go terribly wrong

14 August 2003



13 June 1994

4 June 1996

With Verification
these could have been avoided

Verification

Specification

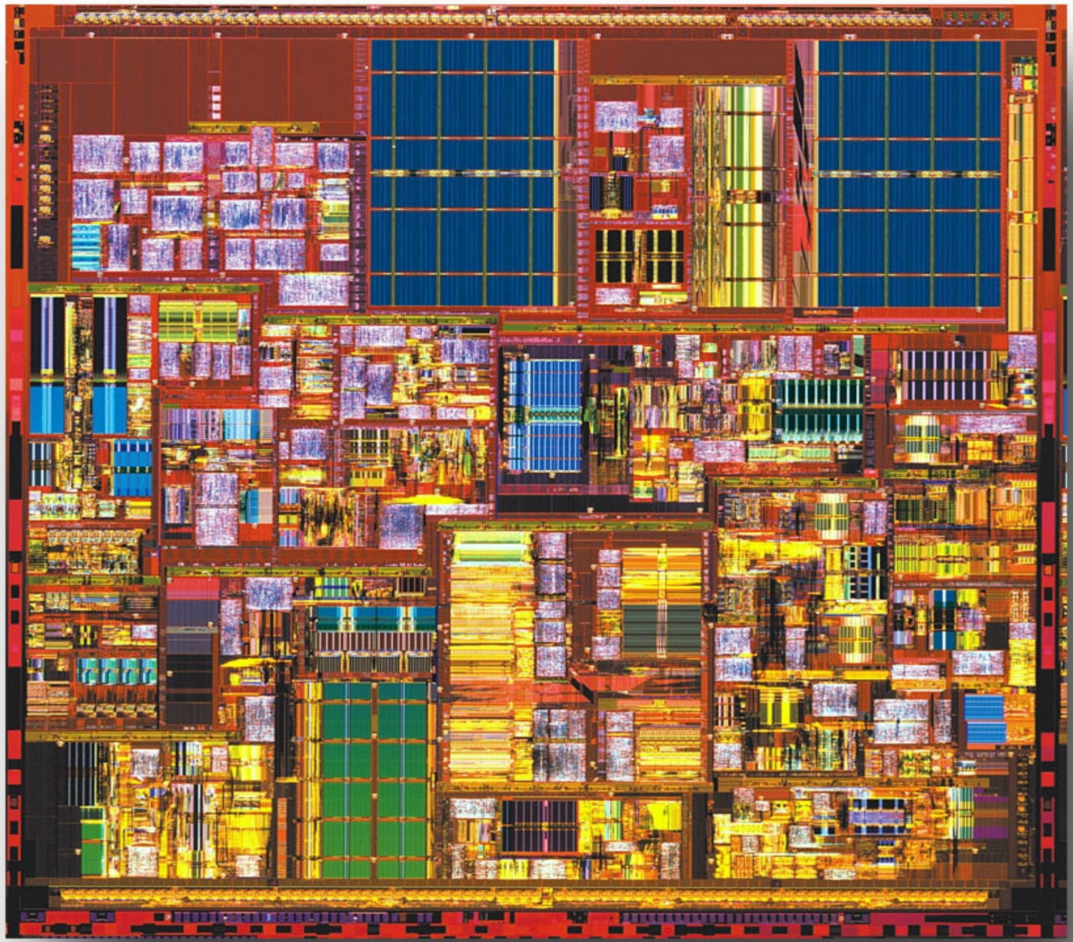
Description of the input-output behaviour

Implementation

A system designed to have the input-output behaviour

Do they match?

Requirements for Hardware

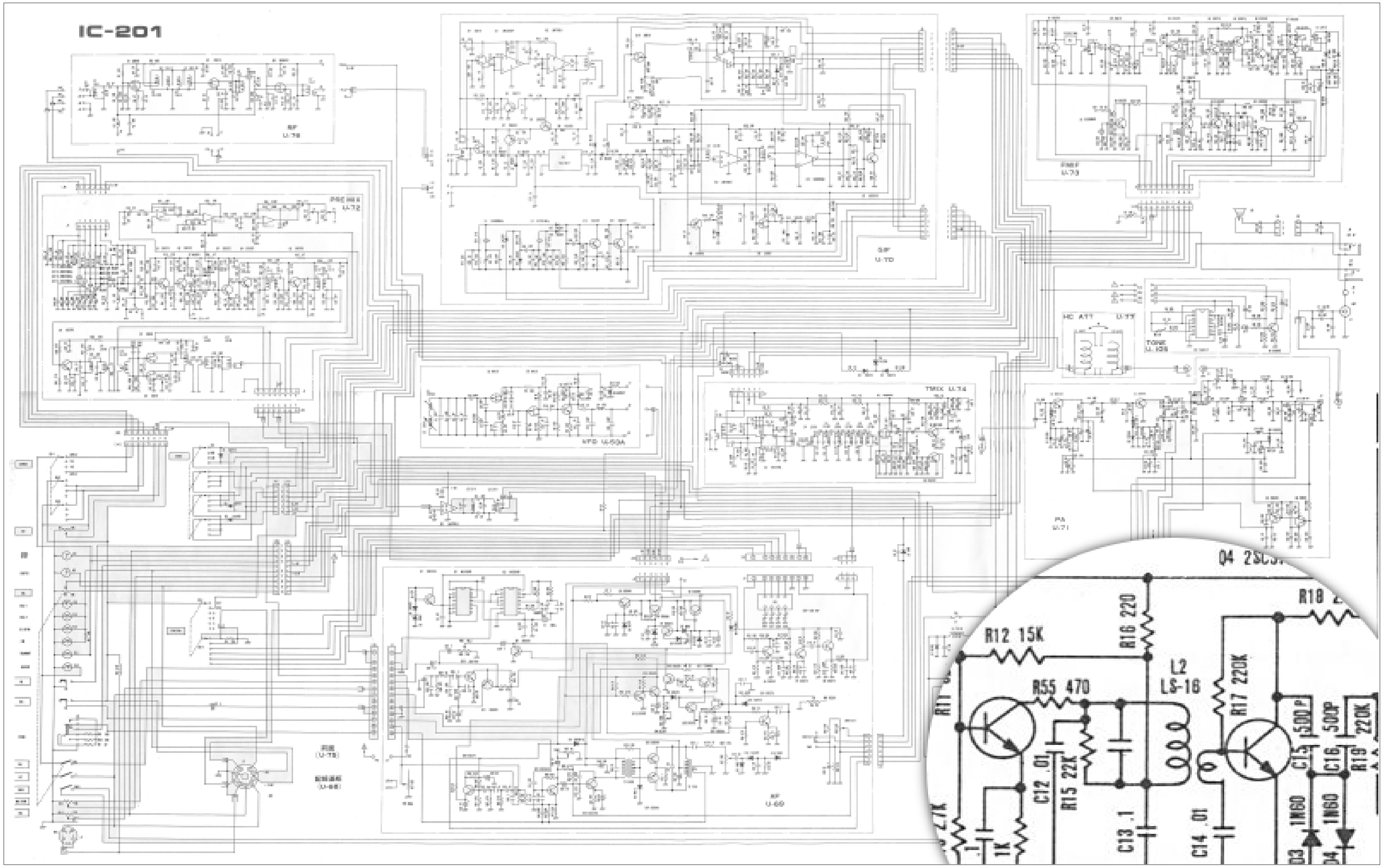


Speed

Size

Power

Concrete designs: very complex



Why is verification hard?

Brute-force method:
Try out every possible input sequence

Example:
Multiplying 32-bit integers

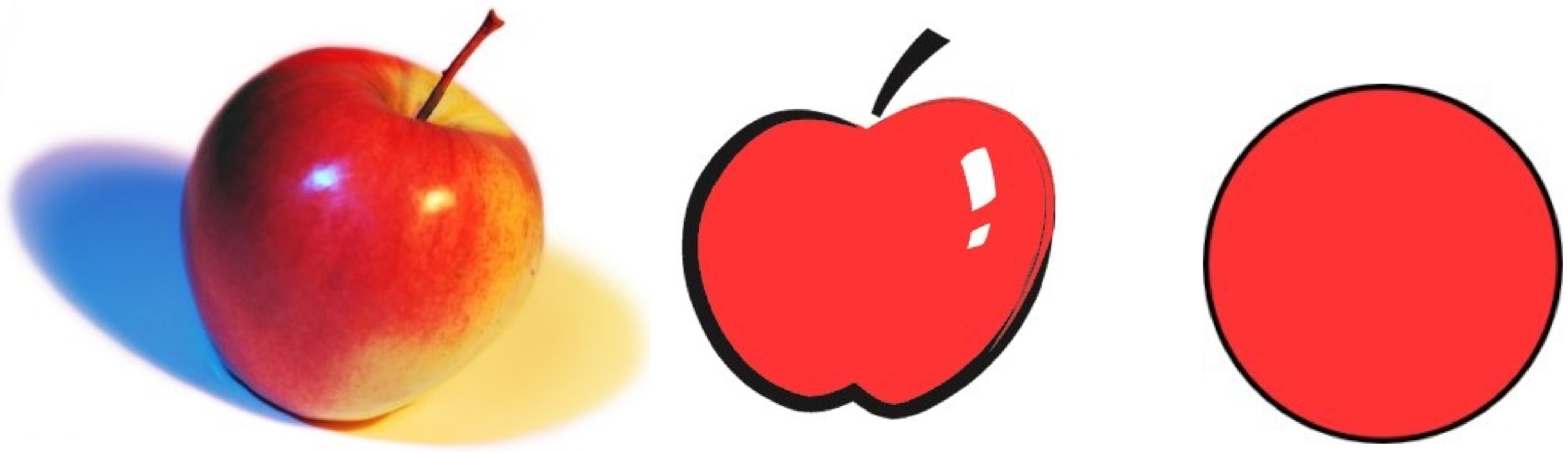
over 18 quintillion test runs
1m runs a second → **over half a million years**

“Not practical”

Abstraction

Work on a simplified version

Problems:
What is a good abstraction?
And how do we get it?



My D.Phil.

Be calculating, Be ignorant, Exploit everything

concrete model: everything is true or false

calculating: allow variables as values
ignorant: allow indeterminates

exploit everything: use spec to abstract