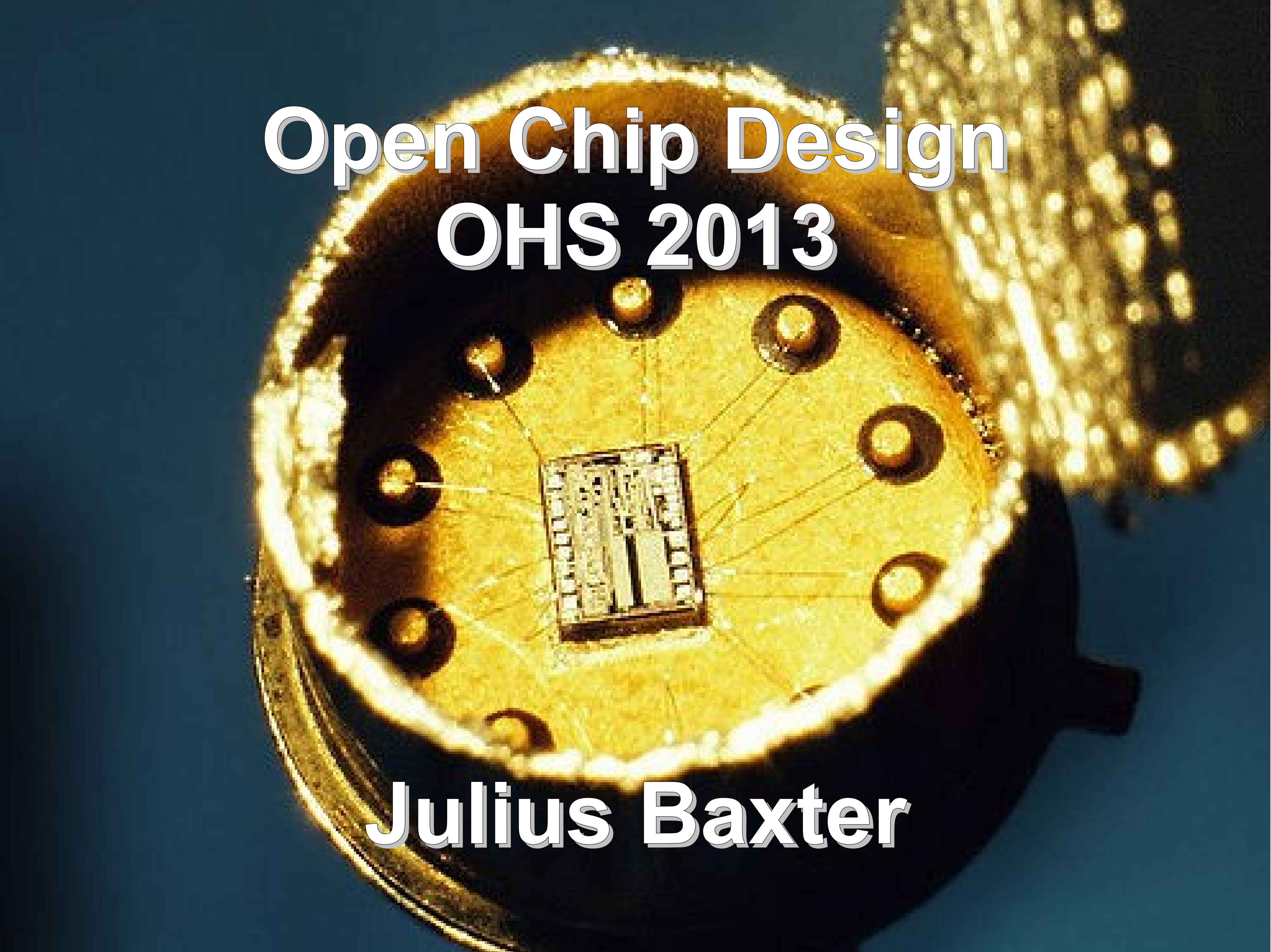
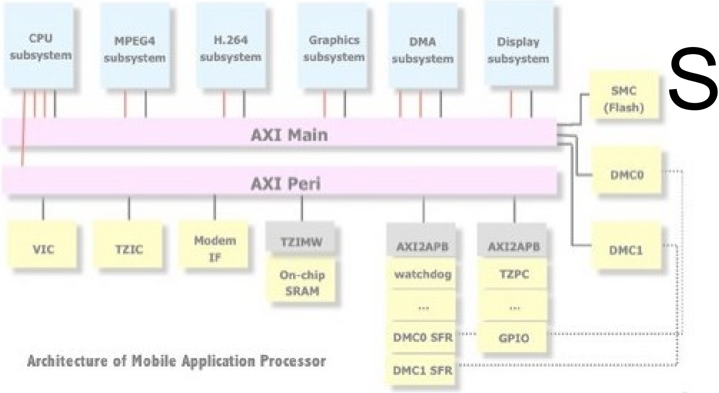
Open Chip Design
OHS 2013



Julius Baxter

“Chip Design” is very broad...

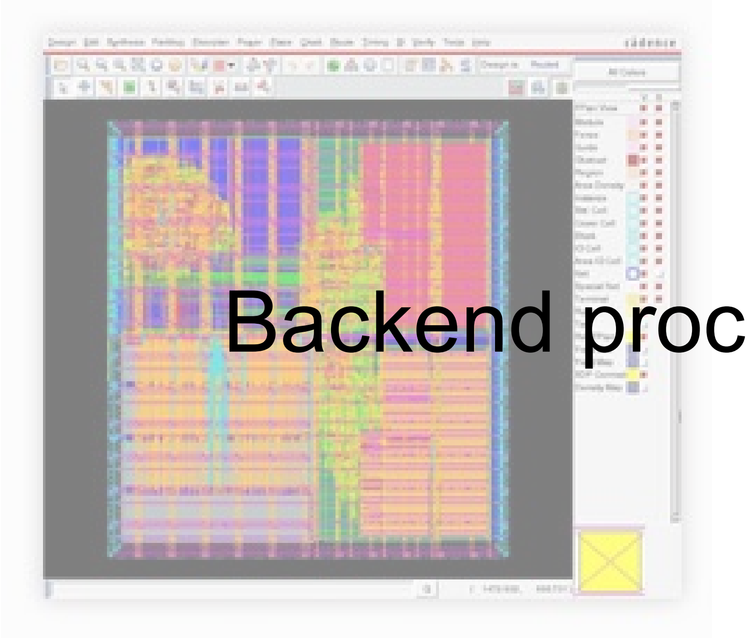
A few steps to make a chip:



System Design

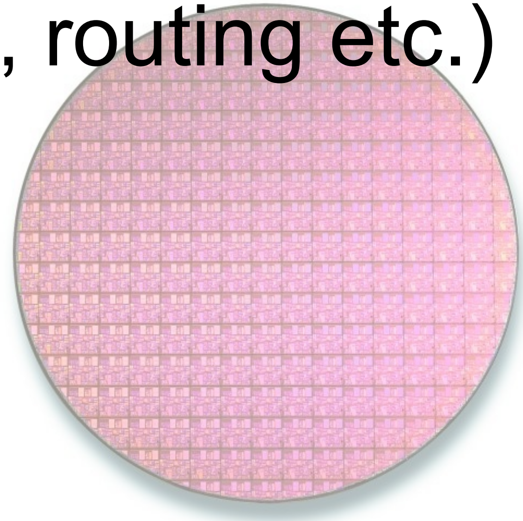
Core Design

Verification (simulation, testing)



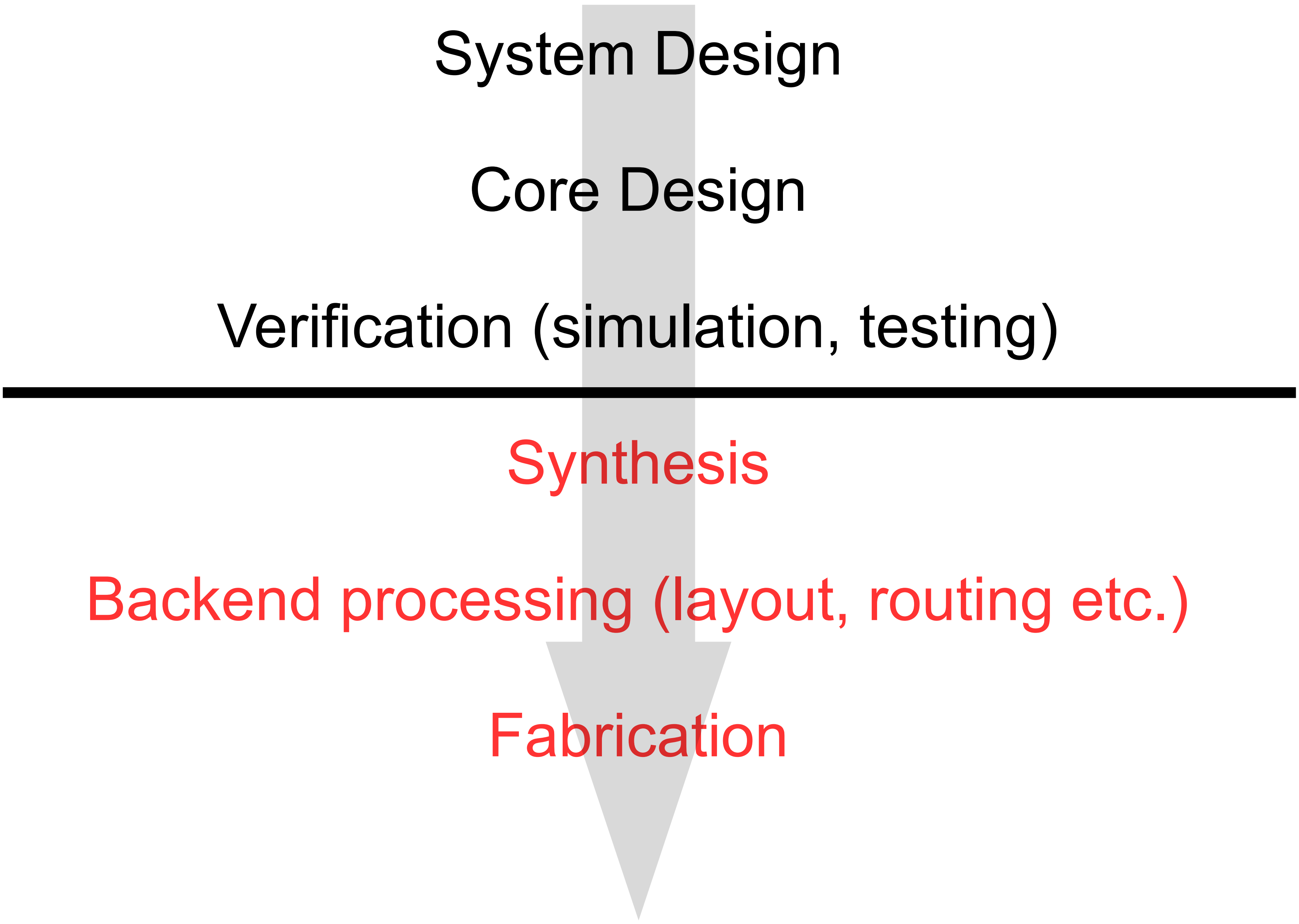
Synthesis

Backend processing (layout, routing etc.)



Fabrication

A few steps to make a chip:



System Design

Core Design

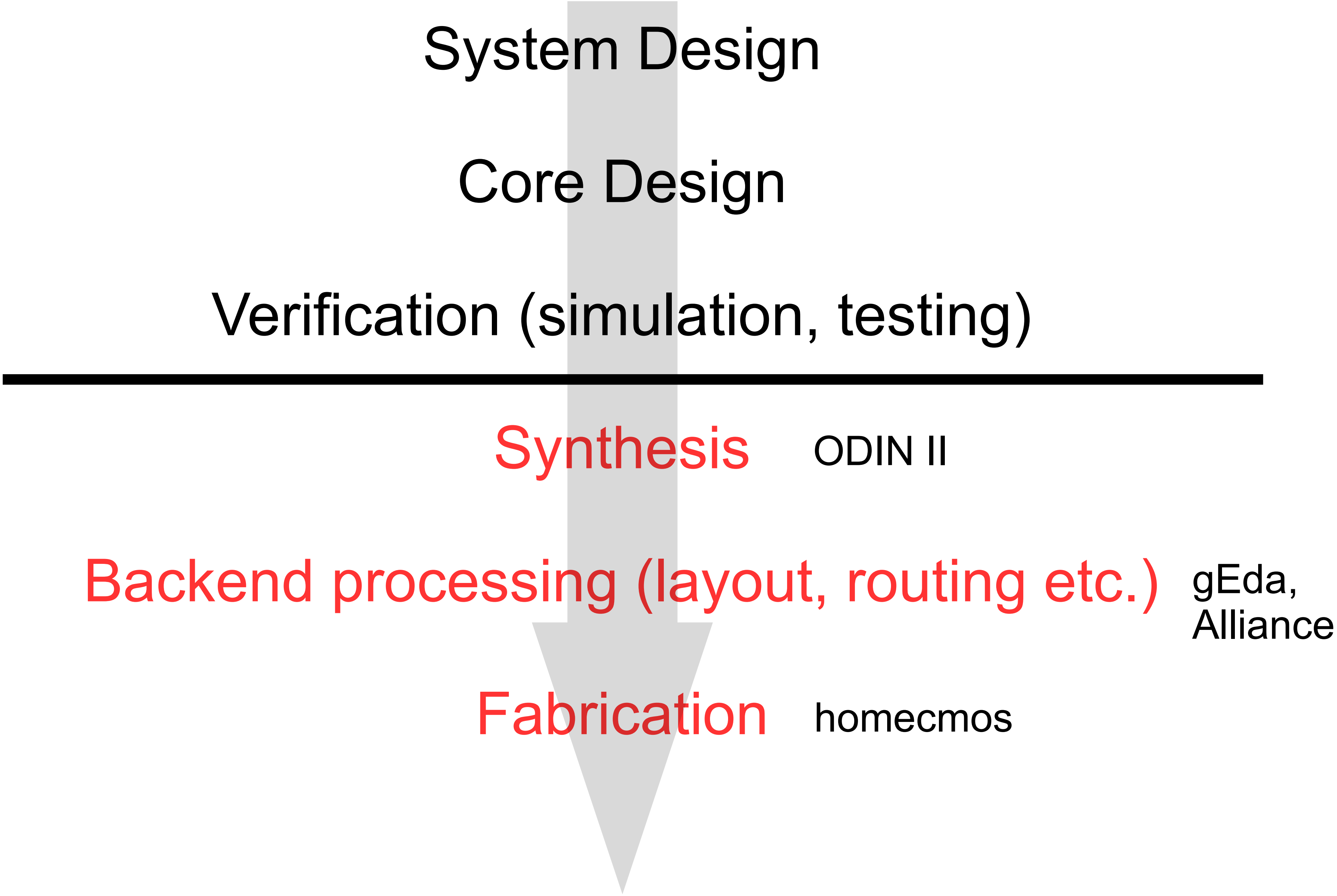
Verification (simulation, testing)

Synthesis

Backend processing (layout, routing etc.)

Fabrication

A few steps to make a chip:



System Design

Core Design

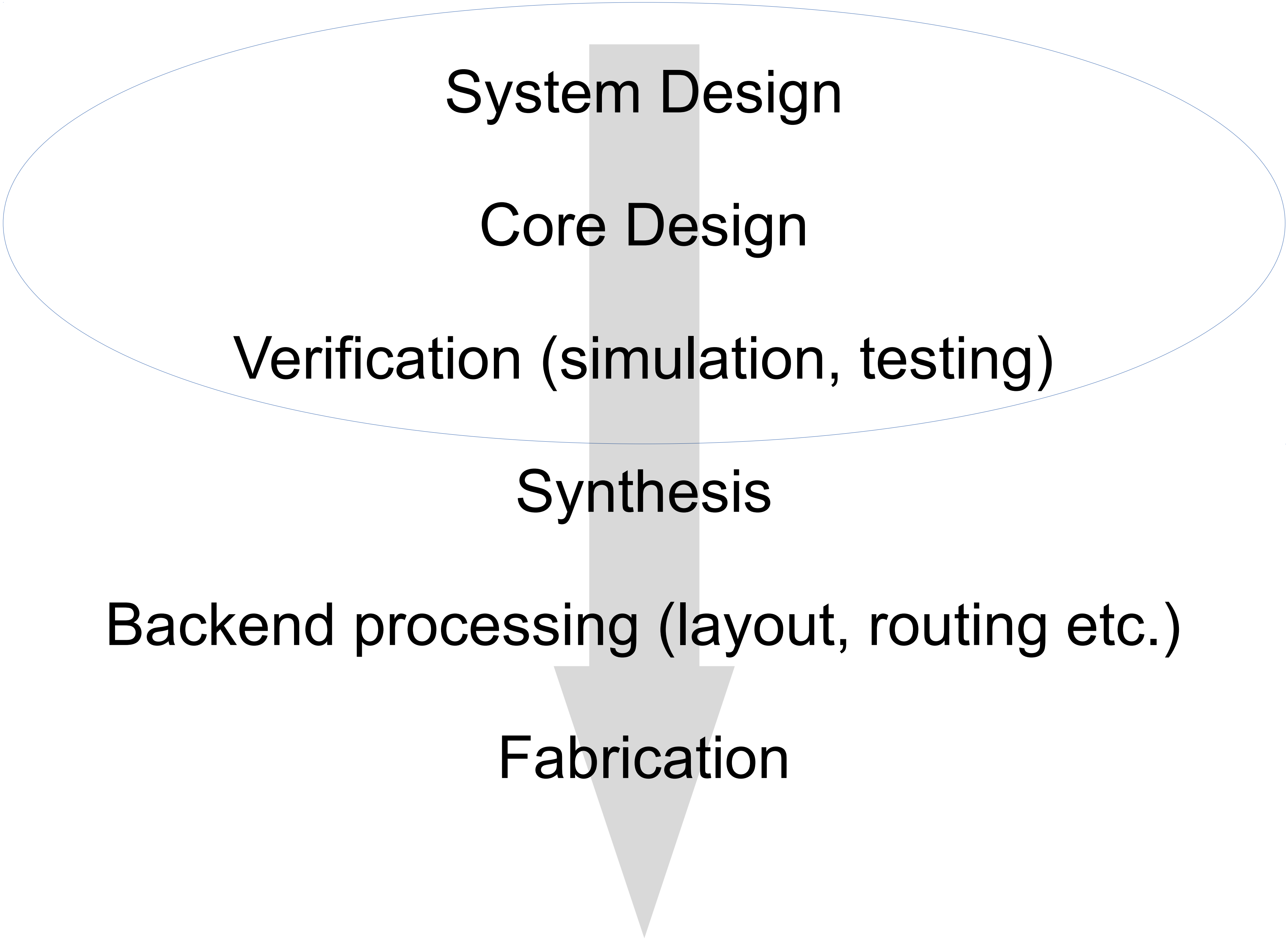
Verification (simulation, testing)

Synthesis ODIN II

Backend processing (layout, routing etc.) gEda, Alliance

Fabrication homecmos

A few steps to make a chip:



System Design

Core Design

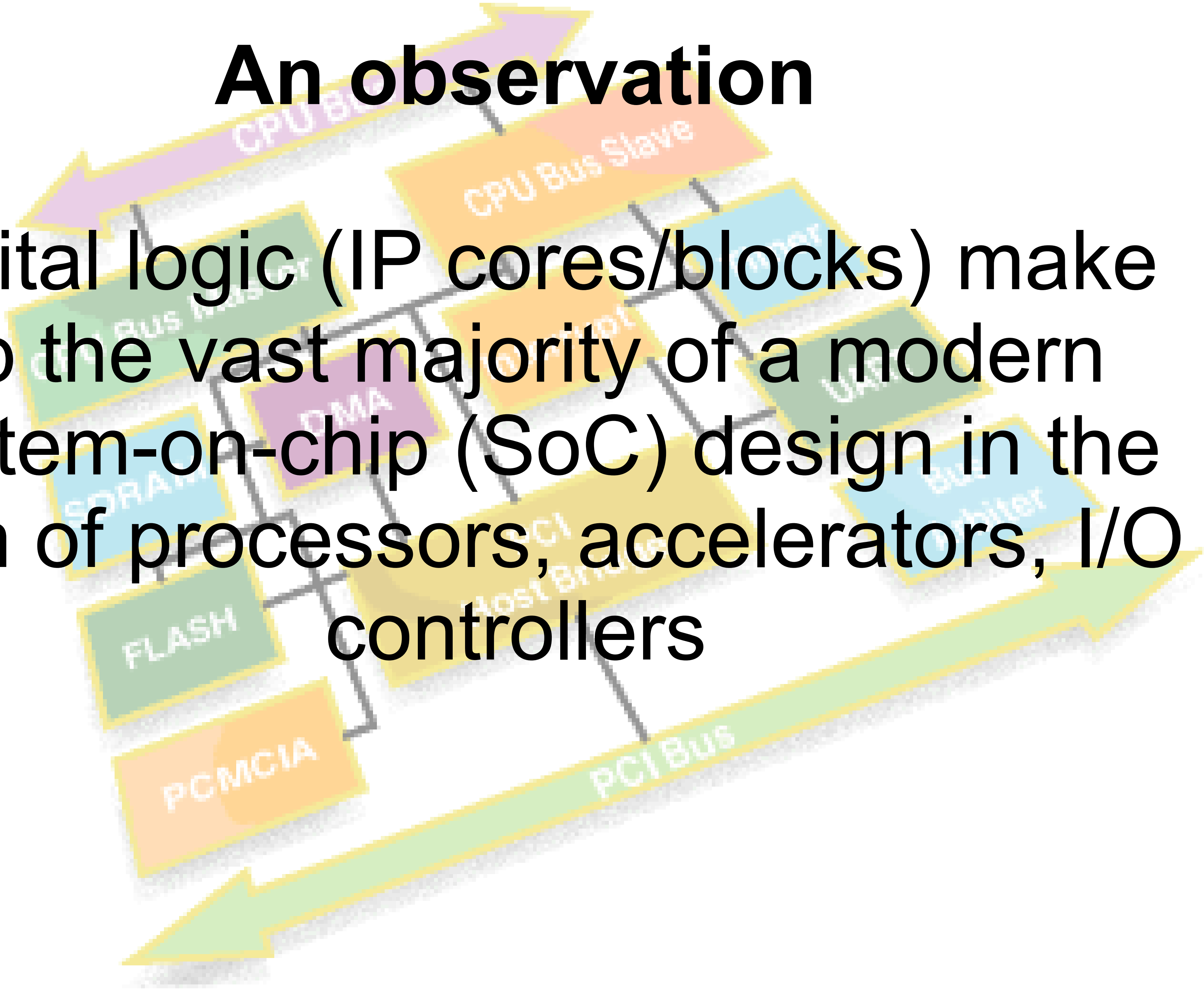
Verification (simulation, testing)

Synthesis

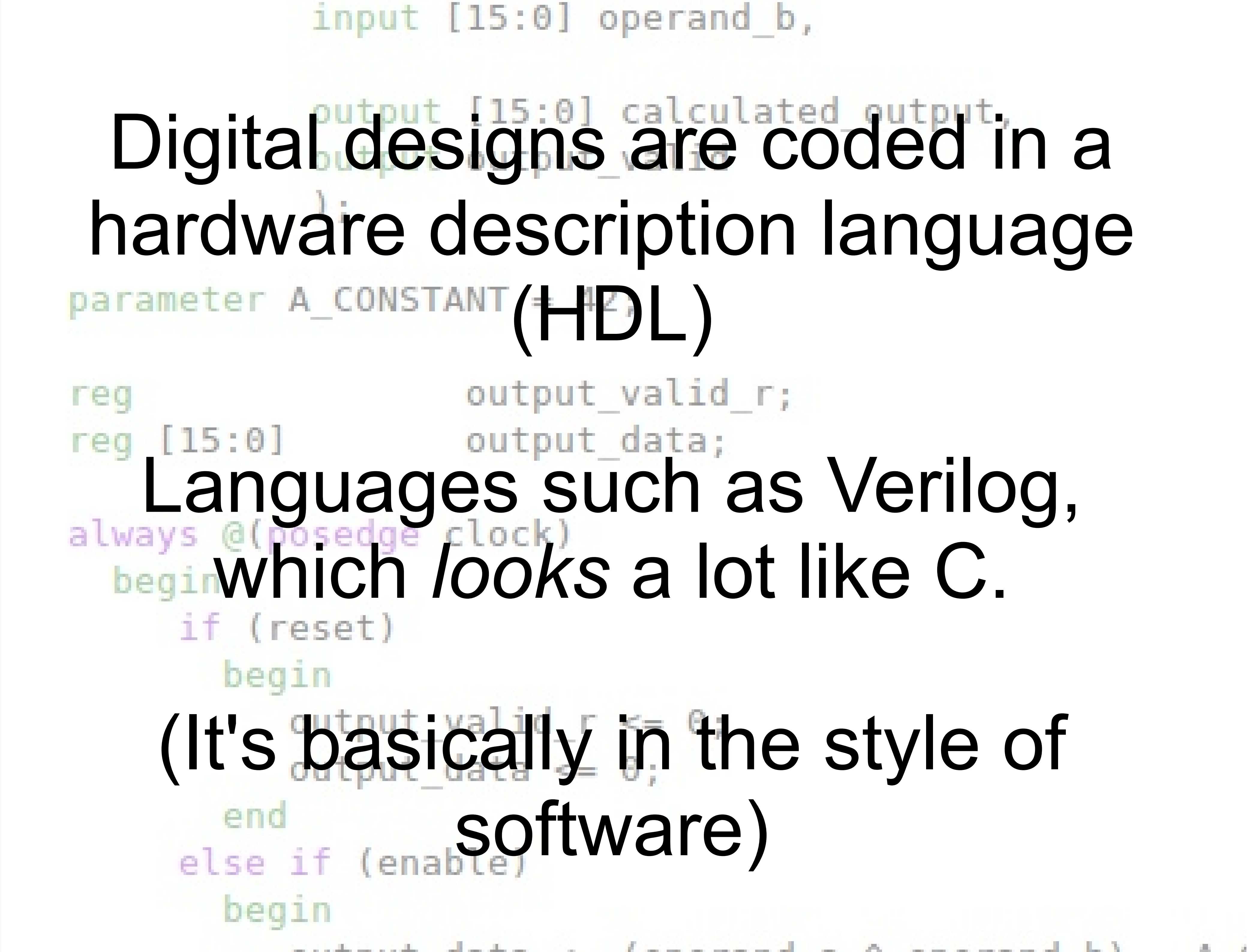
Backend processing (layout, routing etc.)

Fabrication

An observation



Digital logic (IP cores/blocks) make up the vast majority of a modern system-on-chip (SoC) design in the form of processors, accelerators, I/O controllers



```
        input [15:0] operand_b,

        output [15:0] calculated_output,
        output output_valid
        );

parameter A_CONSTANT = 42;

reg             output_valid_r;
reg [15:0]      output_data;

always @(posedge clock)
  begin
    if (reset)
      begin
        output_valid_r <= 0;
        output_data <= 0;
      end
    else if (enable)
      begin
```

Digital designs are coded in a hardware description language (HDL)

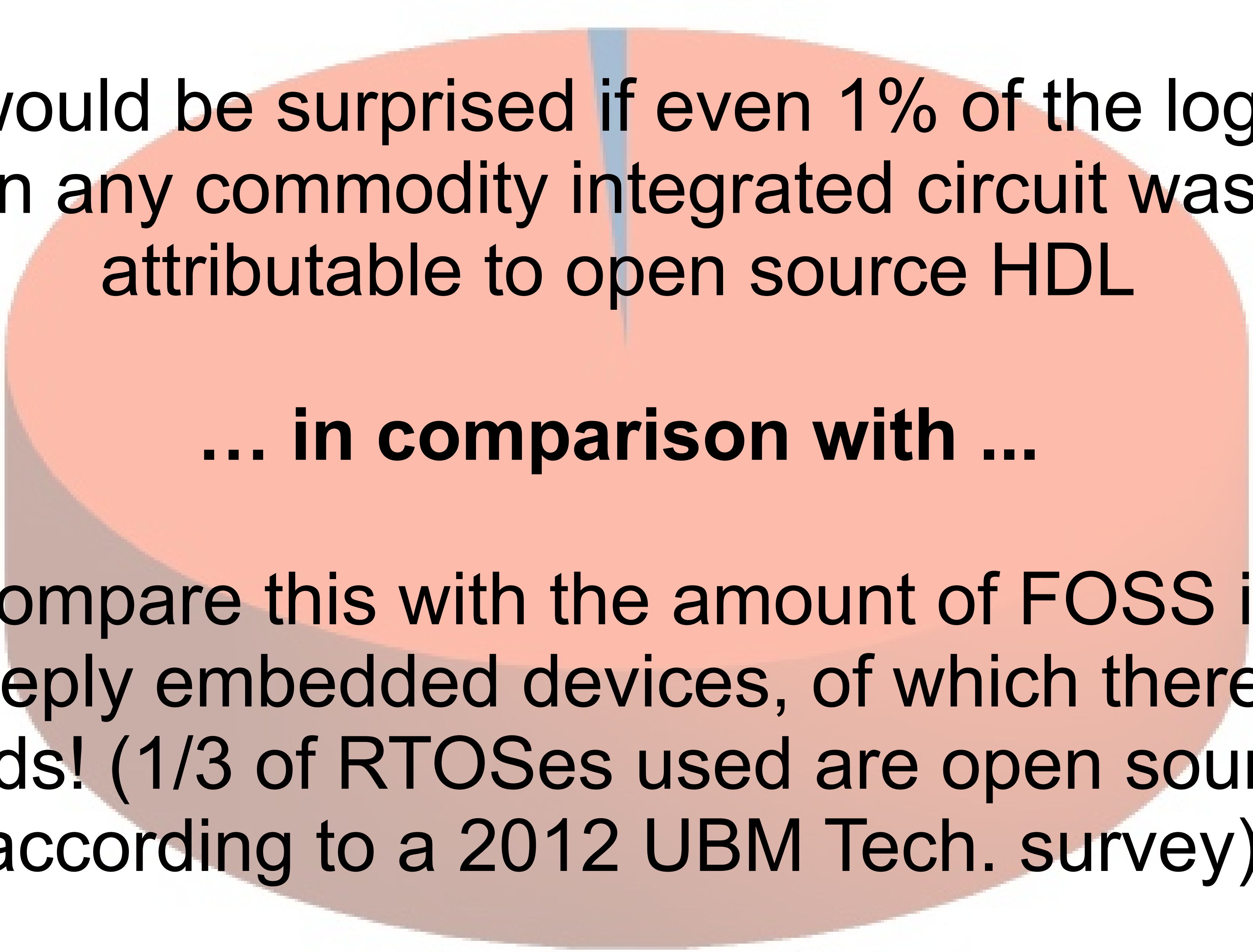
Languages such as Verilog, which *looks* a lot like C.

(It's basically in the style of software)

Q:

Given that this stuff resembles software, and is used to code all of the digital circuitry in our modern lives, why is there *so little* development and use of open source HDL?

***So** little?*



I would be surprised if even 1% of the logic in any commodity integrated circuit was attributable to open source HDL

… in comparison with ...

Compare this with the amount of FOSS in deeply embedded devices, of which there's loads! (1/3 of RTOSes used are open source according to a 2012 UBM Tech. survey)

Some reasons

Culture of secrecy and competition among ASIC design companies, no open collaboration

Given the costs of manufacture, chip designers are very risk adverse

Current state of OS HW projects not brilliant

Far more software than digital engineers

Licensing

What can be done?

Improve the overall quality of the open source IP in terms of...

- **Code quality** – adopting industry-level standards
- **Testing** – easier said than done as all good verification tools are proprietary, but some options exist (Verilator, Icarus)
- **Documentation** – this is important to users of the IP

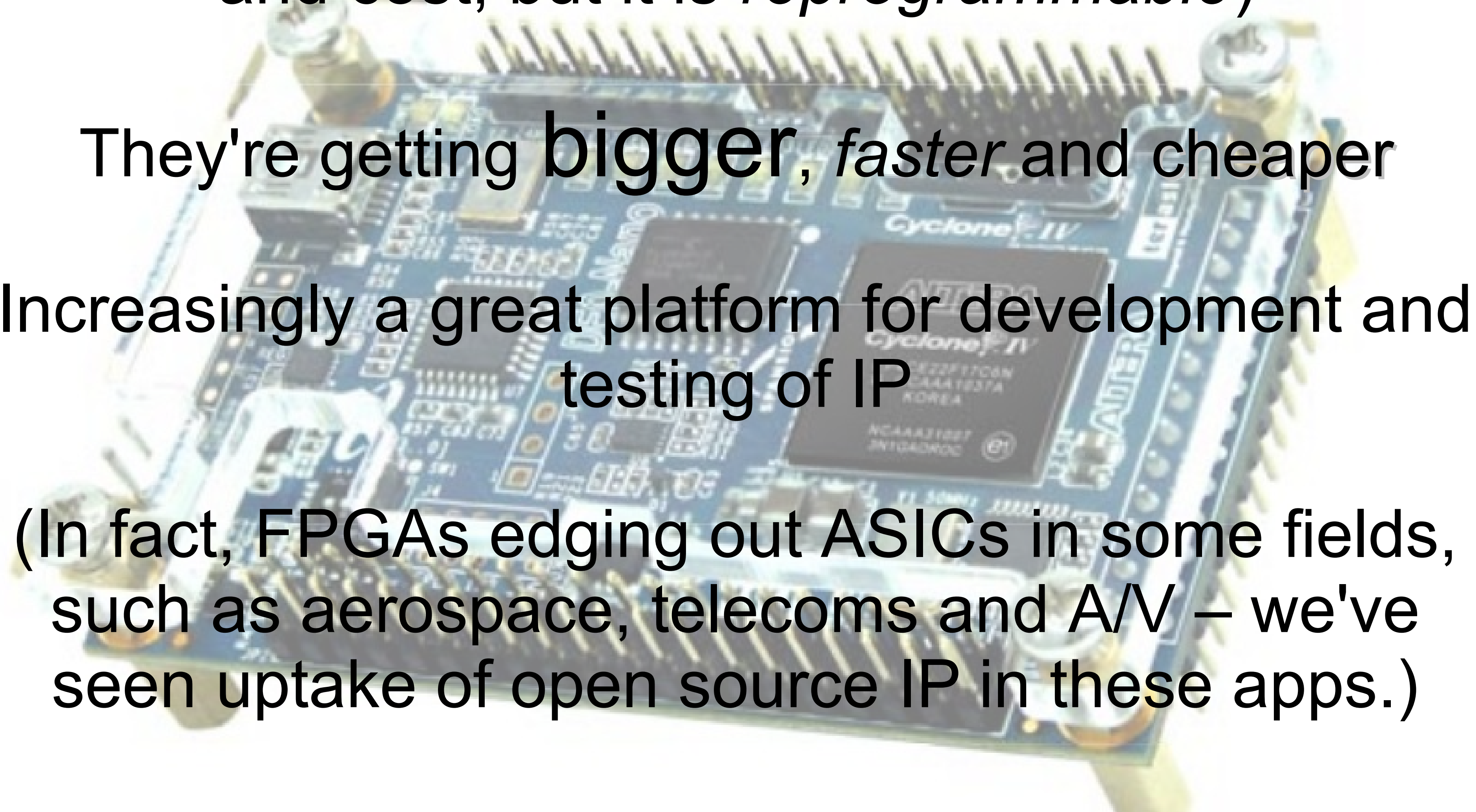
FPGAs are our friend

(An FPGA allows us to put our digital design into a chip, although with restrictions, higher power, area and cost, but it is *reprogrammable*)

They're getting bigger, *faster* and cheaper

Increasingly a great platform for development and testing of IP

(In fact, FPGAs edging out ASICs in some fields, such as aerospace, telecoms and A/V – we've seen uptake of open source IP in these apps.)



Some great OS HW projects:

- OpenRISC – focusing on CPU core development, verification, but encompassing entire SoC development, SW, tools
http://opencores.org/or1k

- Milkymist – video DJ (VJ) SoC on FPGA, also manufactures product based on this!
http://milkymist.org

- CERN-backed Open Hardware Repository
http://ohwr.org
- CERN also leading work on drafting an open source license, one applicable to HDL being discussed

Final observation

Open source software achieved a “critical mass” in the 90s, leading to mass uptake.

Open source IP design is still waiting in the wings, but if the last 5 years are anything to go by, it's on the up.

Exciting times!

That's all from me

http://juliusbaxter.net

#openrisc on freenode

Cheers