The FOSS EDA ecosystem (and GHDL) A short introduction

Tristan Gingold – ISC 2020

FOSS EDA Ecosystem

Free Open Source Software

Any software whose source code is available for any (including commercial) purpose.
Can be:

- Used
- Modified
- Redistributed

EDA

Electronic Design Automation

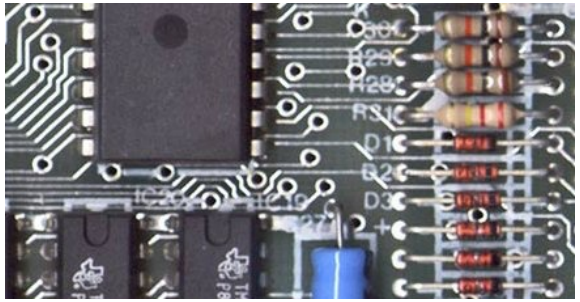
Any software to help (via automation) the electronic hardware designer

This is a vast domain, so let’s clarify

EDA

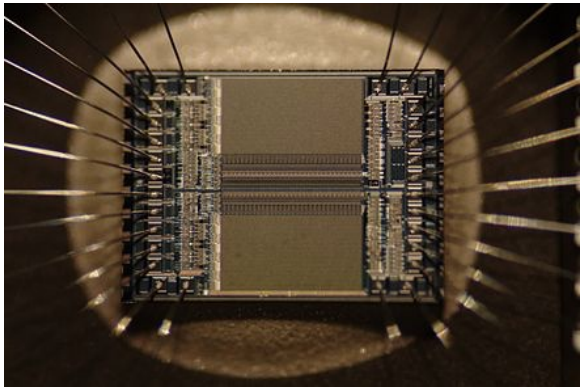
Printed
Circuit
Board

PCBs



Integrated
Circuit

ICs



Analog

Ex: KiCad

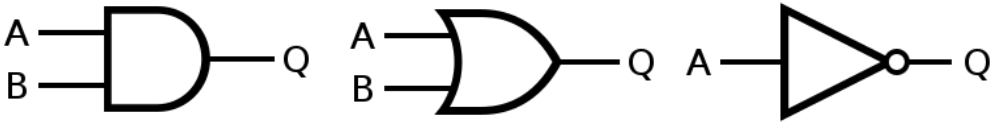
pcb

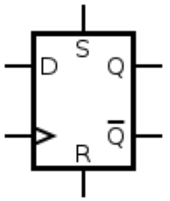
Qucs

NGSPICE

Coriolis
Magic Qflow

What are Integrated Circuits ?

- Logical gates 

- Flip-flop and latches 

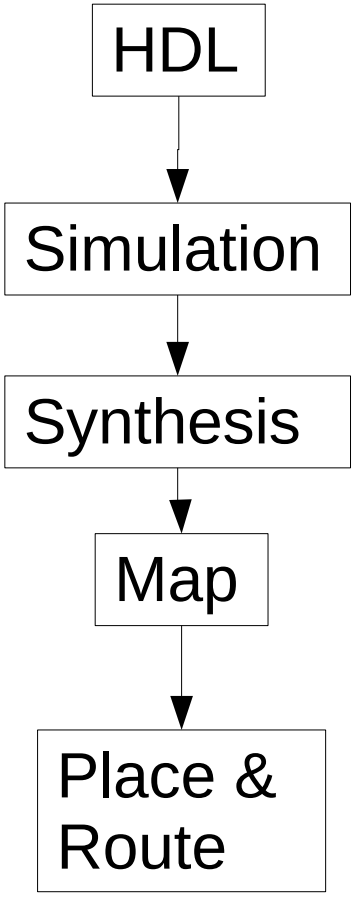
- Memories
- Wires
- Some analog components (PLLs, pads, …)

That’s the netlist!

How are ICs designed ?

- Not anymore by drawings
  - Doesn’t scale
  - Still used for high-level view
- Use of Hardware Description Language
  - Verilog/SystemVerilog
  - VHDL

The design flow

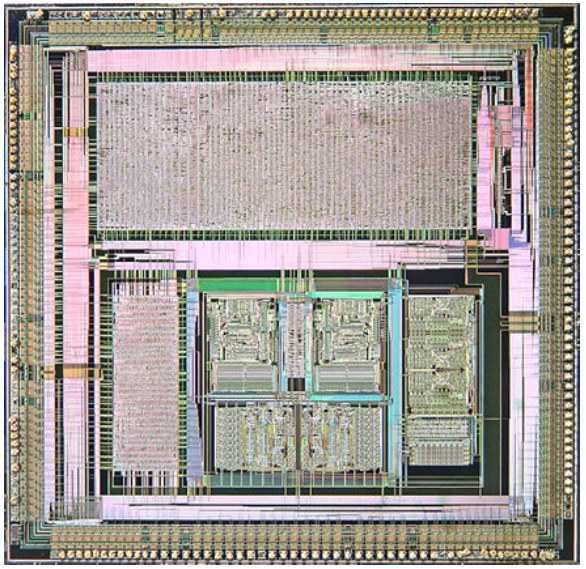


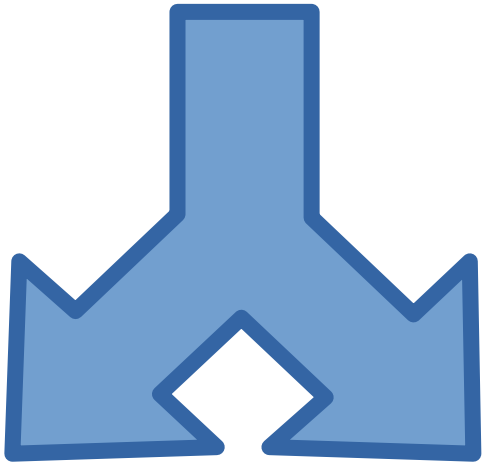
- Write the design
- Check the design
- Generate a generic netlist
- Target specific netlist
- Physical implementation

ICs

Application
Specific IC

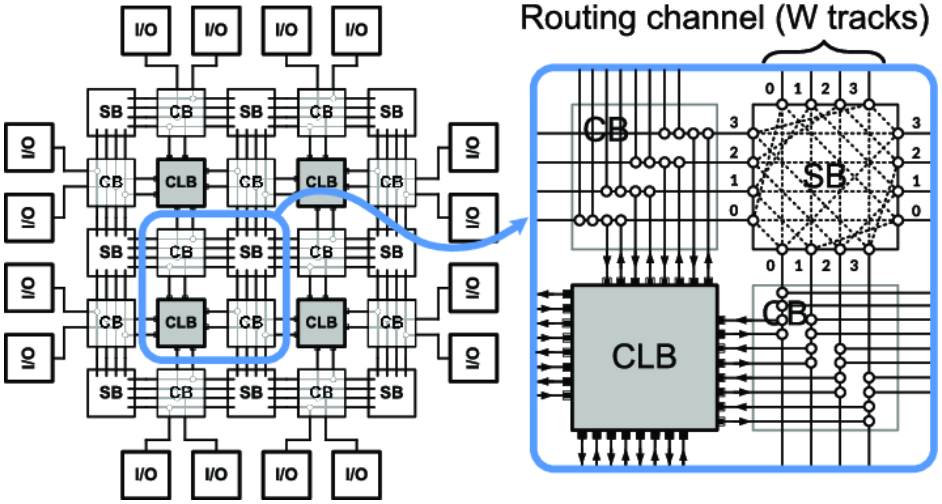
ASIC





Field
Programmable
Gate
Array

FPGA



FOSS for the design flow

New/DSL languages:
- Chisel
- SpinalHDL (Scala)
- Clash (Haskell)
- MyHDL (Python)
- ...

HDL

Simulation

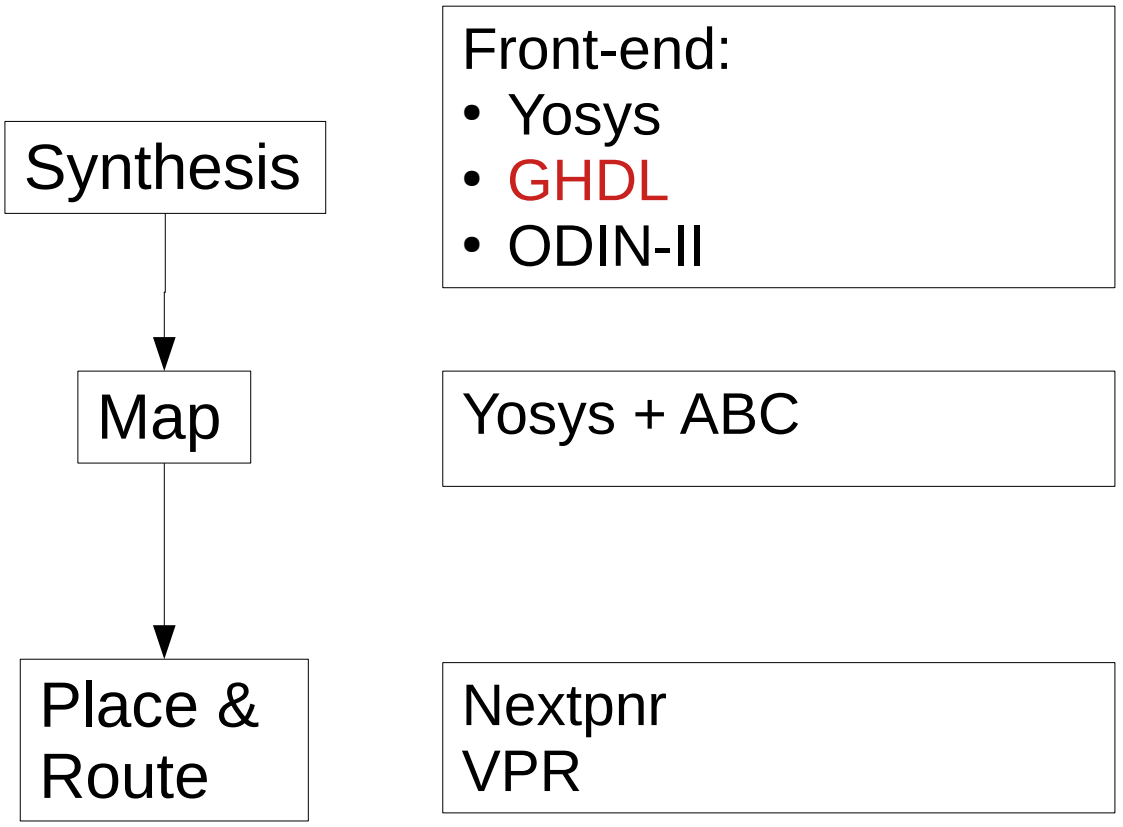
Standard languages:

- Verilog
  - Verilator
  - Icarus Verilog
- VHDL
  - GHDL
  - nvc

Verification frameworks:
- Cocotb (python)
- uvm
- osvvm
- uvvm
- vunit
- ...

Waveform viewer:
- GtkWave

FOSS for the design flow



Why FOSS EDA (1/2) ?

- Prioprietary FPGA tool chains are complete
- They are often free
  - (at least for entry-level FPGA)
- For many flows, they are still required
  - Bitstreams are not documented
- Vendor IPs are often encrypted
- Ecosystem

Why FOSS EDA (2/2) ?

- Coherency if your design is also open-source
- Ideology – contribution
- Price
- Performance (in some cases)
- Avoid vendor-locking
- Features
- Support / Fixes time
- Flexibility