Challenges and Opportunities of Open Source Licensed Hardware

... based on our experiences from the PULP project

22 September 2018

Frank K. Gürkaynak

Integrated Systems Laboratory
ETH Zürich
At ETH Zürich and UniBo we have been working on PULP

- **Open Source Platform**
  - Based on RISC-V ISA

- **RISC-V cores**
  - 32bit: Micro/Zero risky
  - 32bit: RI5CY
  - 64bit: Ariane

- **Complete Systems**
  - PULPino
  - PULPissimo
  - OpenPULP

- **Multicluster systems**
  - HERO (bigPULP)
Already 25+ PULP chips have been sent to fabrication

http://asic.ethz.ch
From day one we knew we **NEED** open source hardware

- **Processors, Peripherals, Interconnect solutions should be commodity**
  - We need them to design and explore architectures that are novel, without them we can not work on things that are interesting
  - Developing 32-bit processors and peripherals is not really research for us.

- **System on Chip design is getting more complex**
  - Even a large group as ours (UNIBO+ETHZ ~ 60 people) can not afford to design everything alone, we need help.

- **Being able to collaborate with who we want is important**
  - Fighting through all the NDAs to get things approved is a nightmare
  - ... especially if no one really cares about the things that are under NDA
(almost) all we have on PULP is released Open Source

- First release 2016
  - PULPino was a 32-bit single core system

- As of 2018 most of what we have developed is online
  - We continue work on public repos (http://github.com/pulp_platform)
  - 32/64 bit cores, peripherals, multi-core / multi-cluster systems

- Rewarding experience
  - Got some nice press and feedback (I even weaseled myself into OrCONF)
  - Surprised to hear many institutes and companies using our designs (even in products)
  - Started several collaborations with new partners
Where are we now?

OPEN

Emacs
Linux
RISC-V
PULP

Application
Operating System
Instruction Set Arch
Microarchitecture
Components (RAM, ALU)
Gates
Transistors

NOT YET

SW
HW

Frank. K. Gürkaynak - Open Source Hardware
Some (surprising) side effects of open source hardware

- There is a surprising amount of bureaucracy involved
  - Code Ph.D. students/staff develop belongs to the university (they pay us)
  - Master/semester thesis students own the work they produce
  - Need to get proper approval for everyone involved.

- Most agreements with companies are not meant for open source
  - Instead of paying for exclusive IP, we need sponsoring agreements
  - Important to make sure we do not sign anything that binds our open source effort

- There are worries that if you see inside you will be more vulnerable to patent trolls / litigation
  - Some companies do not want it known that they use our designs
Hardware design flows for PCB/FPGA/ASIC are different

- The differences complicate things
  - Not a uniform way to discuss the issue, depends on the design flow
  - This talk about ASIC flow (the most complex one).

- Different actors, different revenue streams
  - Not every actor in the current design flow, earns money the same way
  - EDA companies are long complaining that they are out of the loop
    Their income is not based on the amount of ICs produced.
  - Not everybody will be happy if open hardware will be more common
  - Important to understand the relationships

- Interestingly most intermediate file formats are open, readable
  - Verilog, Liberty, SPICE, EDIF, CIF, LEF, DEF, OA (open access)
FPGA Design Flow and Main Actors

- High-Level Code
- HLS
- HDL
- Simulator
- Logic Synth.
- Netlist
- P&R
- Bitfile
- FPGA
- Sim. Models
- FPGA Resour.
FPGA Design Flow and Main Actors

- High-Level Code
- HLS
- HDL
- Simulator
- Logic Synth.
- Netlist
- P&R
- Bitfile
- FPGA
- Sim. Models
- FPGA Resour.
FPGA Design Flow and Main Actors

- FPGA vendors want to sell the FPGAs
- Little commercial interested in intermediate files

- Most vendors allow bitfiles to be published -> will sell more FPGAs
 ASIC Design Flow and Main Actors

**EDA Companies**
- High-Level Code
- Logic Synth.
- Netlist
- P&R
- GDSII
- Foundry
- Chip

**Technology Provider**
- PDK

**IP Providers**
- Sim. Models
- Std. Cells
- Other IP
- HDL
- HLS

**Other**
- Sim. Simulator
- Logic Synth.
- Netlist
- P&R
- GDSII
- Foundry
- Chip
ASIC Design Flow and Main Actors

- Only beginning of the pipeline can be open HW
- Later stages contain business interest of various actors
At the moment, open HW can (mostly/only) be HDL code

- **The following are ok:**
  - RTL code written in HDL, or a high-level language for HLS flow
  - Testbenches in HDL and associated makefiles, golden models

- **How about support scripts for different tools?**
  - Synthesis scripts, tool startup files, configurations

- **And these are currently no go:**
  - Netlists mapped to standard cell libraries
  - Placement information (DEF)
  - Actual Physical Layout
What is HDL code used for

Simulation/Modeling

FPGA Design

ASIC Design

Virtually Indistinguishable from standard Software

Frank. K. Gürkaynak - Open Source Hardware
Can we use licensing from software also for HDL code?

In principle yes

- For most part HDL code is indistinguishable from other software

- This is what we are doing for the PULP project at the moment
  We use Solderpad hardware license (derived from Apache/BSD license)
  SolderPad
  http://www.solderpad.org/licenses/

But there are some issues

- Not clear if all licensing models work
  For example LGPL v3 (sec. 4.d) discussion:
  “You need to ship/convey everything that is needed to make a new combined work with a modified version of the library”

- Some EDA vendors think it is not possible to develop HDL code without using their tools

- A copyleft license specific for HW that stops at design boundaries is needed
What is not yet FREE in open source hardware

- **Transistors, Technology**
  - Manufacturing ICs is very costly, a lot is possible only if you have volume
    - Printing your own (performant) IC at home is not happening anytime soon
  - We have to rely on big fabs for manufacturing. They 'own' the transistors
    - Actually they own design data needed to build with transistors

- **Physical building blocks (GDS)**
  - If you want to manufacture ICs you need construction plans (GDS files)
  - All ICs need support circuitry that is technology specific
    - Memories, standard cells, PLLs, I/O drivers, USB PHYs, DDR PHYs, ADCs..

- **EDA Tools**
  - For years EDA companies supplied academia with 'cheap licenses'.
  - For years negligible effort was put into developing free EDA software.
What can be done to make hardware more open?

- Manufacturing is done via the foundries, we need access to them
- Foundry gives a Process Design Kit (PDK):
  - Provides the "necessary information" to design in the technology
  - Currently this information is (aggressively) protected by NDAs
  - Sometimes with "strange provisions"

“For as long Customer’s employees and students (to the extent applicable) are actively involved in research, prototyping or manufacturing activities of Customer which grant access to Confidential Information, such employees and students may not engage in any research, prototyping and/or manufacturing activities on the 57nm technology node or smaller with any company with semiconductor wafer manufacturing capability other than ZSC.”
What is the "secret" information that is in the PDK?

Design Rules

Layer Stack

Transistor Models

All information on this slide is publicly available. From left to right: taken layout from http://vlsi.wpi.edu, layer stack from http://xfab.com, transistor models from ptm.asu.edu
This information is not really so secret, or so relevant!

- Delayering and metrology will reveal all
  - Most are trivial, and are already known
    “The gate length of our 65nm process is 65nm”

- The real trade “secret” is
  - How to tune the manufacturing process so that chips can be produced reliably with these parameters.

The information in the PDK does not really reveal that

- The Process Design Kit helps people design circuits for your technology.
  - Why keep it a secret?

Only few companies can manufacture chips
- TSMC
- UMC
- Globalfoundries
- Samsung
- …

All have the capability to ‘extract’ this data from their rivals
So why keep it a secret at all? Some theories

- Safer to keep things as they are.
  - What if they give away too much (lose money?)
- Some believe it is actually a trade secret
  - These are mostly non-technical people (lawyers)
- Not everything may belong to the company
  - May have parts bought (borrowed) from 3rd parties
- Exclusive agreements for support IP
  - These days fabs do not want to deal with simple IP
- They ask 3rd parties to do it for them (saves support)
  - They may have given these 3rd parties incentives

You will be the exclusive design enablement provider

Important to understand why if we are to change anything

Frank. K. Gürkaynak - Open Source Hardware
If you are old enough: It worked differently before

- **Lambda based rules**
  - Technology independent design rules

- **Chips were made with lambda rules**
  - Early manufacturing processes supported this
  - Porting was easy

- **Newer technologies needed stricter rules**
  - Companies developed their own custom rules
  - These were no longer scalable with one parameter (lambda)
  - But they were also no longer freely available
Why should foundries give free access to technology data

- They will not lose anything
- Number of IP for technology will increase
  - Many enthusiasts, SMEs, academic institutions are willing to contribute, quality not behind companies
  - More people will be able to look into the IP, better verification, more reliable.
- Silicon verified IP is paramount
  - Only way of getting silicon-verified Open Hardware
- Potentially more customers
  - The more IP is available, the more attractive a technology becomes for production
Why should foundries give free access to technology data

- They will not lose anything
- Number of IP for technology will increase
  - Many enthusiasts, SMEs, academic institutions are willing to contribute, quality not behind companies
  - More people will be able to look into the IP, better verification, more reliable.
- Silicon verified IP is paramount
  - Only way of getting silicon-verified Open Hardware
- Potentially more customers
  - The more IP is available, the more attractive a technology becomes for production

Frank. K. Gürkaynak - Open Source Hardware
Why should foundries give free access to technology data

- They will not lose anything
- Number of VC for technology will increase
  - Many enthusiasts, SMEs, academic institutions are willing to contribute, quality not behind companies
  - More people will be able to look into the VC, better verification, more reliable.
- Silicon verified VC is paramount
  - Only way of getting silicon-verified Open Hardware
- Potentially more customers
  - The more VC is available, the more attractive a technology becomes for production

VC = Verified Component
Does it matter that this information is not free?

ABSOLUTELY !!!

- As long as this information is kept confidential, we can not
  - Exchange physical layout data
  - Design standard cells, memories, I/O cells (things you need for a basic digital design) and make these freely available for other open source projects
  - Allow completely open hardware from RTL to GDSII
  - Develop/share analog components (ADCs, PLLs, Memory interfaces, Serial I/O)
What’s next

- Convincing any foundry to give access to PDK is not easy
  - Do not expect this to happen short term (> 5 years)
- But it will happen
  - It makes sense
  - It has the potential to increase revenue for the foundry (maybe not for others)
  - It will not cost the foundries extra (no active support needed)
- We need to raise awareness for the need for open hardware
  - We need to have the discussion, why the PDK access is not open
  - Discussions will eventually lead to change