Why Open Source Hardware matters and why you should care

Author[s]:
Gürkaynak, Frank K.

Publication Date:
2017-07

Permanent Link:
https://doi.org/10.3929/ethz-b-000171624

Rights / License:
In Copyright - Non-Commercial Use Permitted
Why Open Source Hardware matters...

... and why you should care

3 July 2017
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

- Released Pulpino
  - Single core system
  - February 2016

- Updates in preparation
  - Probably in August

- We will release more
  - Practically everything
  - SDK
  - Multi-core versions
Already 20+ PULP chips have been sent to fabrication

http://asic.ethz.ch
Based on our experience on Open Hardware, I will

- Present the current situation on Open Hardware
  - Why is open hardware different than open software
- Discuss why Open Hardware matters
  - It is not just "ideology", there are practical reasons
- Clarify the challenges facing Open Hardware
  - At the moment only limited to HDL description
- Explain how open access to manufacturing data will help us
  - To be able to have a completely open IC, manufacturing companies need to reveal their “secret” information on their technologies
What is Open Hardware?

- **From Open Source Hardware Association**
  Open source hardware is hardware whose design is made publicly available so that anyone can **study, modify, distribute, make**, and **sell** the design or hardware based on that design.

- **OSHWA** is mainly concerned with Printed Circuit Boards (PCBs).

- But the idea is also valid for FPGA and even ASIC designs
Open Source HW is where SW was one generation ago

- There is a visible momentum
  - Berkeley RISC-V foundation
  - Cambridge Low-RISC
  - ETH Zürich PULP
  - FOSSI foundation
  - many more...

- Already several companies are using it
  - Google, Mentor, Micron, Greenwaves
  - Already products with open source hardware

- But this is still the beginning
  - About where the Open SW was 1985-1995

Open Source Software Timeline

- GNU - 1983
- FSF - 1985
- GCC - 1987
- GPL - 1989
- Linux - 1991
  - v1.0 - 1994
  - v1.2 - 1995
  - v2.0 - 1996
Why is Open Hardware different than Open Software

- From gnu.org www site:
  http://www.gnu.org/philosophy/free-hardware-designs.html

- **Software** is the operational part of a device that can be copied and changed in a computer

- **Hardware** is the operational part that can't be.

- You can not produce HW directly, you need
  - manufacturing plants
  - know-how
  - and volume

  to be able to manufacture HW within reasonable cost.
Open Hardware is a necessity, not an ideological crusade

- The way we design ICs has changed, large part is now infrastructure
  - Processors, peripherals, memory subsystems are now considered infrastructure
  - Very few (if any) groups design complete IC from scratch
  - High quality building blocks (IP) needed

- **We need an easy and fast way to collaborate with people**
  - Currently complicated agreements have to be made between all partners
  - In many cases, too difficult for academia and SMEs

- **Hardware is a critical part of security, we need to make sure it is secure**
  - Being able to see what is really inside will improve security
  - Having a way to design open HW, will not prevent people from keeping secrets.
Open Hardware is a necessity, not an ideological crusade

- The way we design ICs has changed, large part is now infrastructure
  - Processors, peripherals, memory subsystems are now considered infrastructure
  - Very few (if any) groups design complete IC from scratch
  - High quality building blocks (IP) needed
The way we do IC design has changed

1987

What We Designed
New

IP (I/O)
IP (pwr)
IP (mem)
IP (interconnect)
IP (ADC)
IP (core)

IP (Clock)

What We Designed
New

2016

Parallel Multiplier 3, Fast CMOS multiplier, Faselec 3µm, 2.5mm x 2.0mm
http://asic.ethz.ch/1986/Parallel_Multiplier3.html

VivoSoC2, Biomedical signal Acquisition SoC, SMIC130, 4.7mm x 4.7mm
http://asic.ethz.ch/2016/Vivosoc2.html
Open Hardware is a necessity, not an ideological crusade

- The way we design ICs has changed, large part is now infrastructure
  - Processors, peripherals, memory subsystems are now considered infrastructure
  - Very few (if any) groups design complete IC from scratch
  - High quality building blocks (IP) needed

- We need an easy and fast way to collaborate with people
  - Currently complicated agreements have to be made between all partners
  - In many cases, too difficult for academia and SMEs
It is not easy to collaborate with all these NDAs

Scenario:
- We (ETH) have access to a technology
  - Zarcovia Semiconductor Company (ZSC)
- We have standard cells from
  - An IP company called LEG Inc.
- We are using tools from
  - A well known EDA Company Syndance

We want an expert from Syndance help us sort the problems we have on ZSC technology using standard cells from LEG

<table>
<thead>
<tr>
<th>Date</th>
<th>Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>22 Oct 15</td>
<td>ETH ➔ Syndance</td>
</tr>
<tr>
<td>30 Oct 15</td>
<td>Syndance NDA?</td>
</tr>
<tr>
<td>30 Oct 15</td>
<td>ETH ➔ ZSC, LEG</td>
</tr>
<tr>
<td>4 Nov 15</td>
<td>ZSC ➔ ETH ok</td>
</tr>
<tr>
<td>17 Nov 15</td>
<td>ETH ➔ LEG ?</td>
</tr>
<tr>
<td>26 Nov 15</td>
<td>ETH ➔ LEG ?</td>
</tr>
<tr>
<td>16 Mar 16</td>
<td>LEG responds</td>
</tr>
<tr>
<td></td>
<td>4 months 23 days !!</td>
</tr>
</tbody>
</table>
Two parties that have individual access can’t share data

- **ETH Zürich and University of Bologna both have signed NDAs to:**
  - Access the Zarcovia Microelectronic Company (ZSC) 57nm technology
  - Use the LEG standard cell libraries and memory compilers and I/O cells

- **ETH Zürich can not send University of Bologna:**
  - a [netlist](https://en.wikipedia.org/wiki/Nets_list) mapped to the ZSC 57nm process using LEG standard cells

- **All parties that want to share data, need to sign a Mutual NDA**
  - This is not automatic, needs approval, takes time

- **... and if a third party wants to join**
  - You need a new Mutual NDA from scratch
Open Hardware is a necessity, not an ideological crusade

- The way we design ICs has changed, large part is now infrastructure
  - Processors, peripherals, memory subsystems are now considered infrastructure
  - Very few (if any) groups design complete IC from scratch
  - High quality building blocks (IP) needed

- We need an easy and fast way to collaborate with people
  - Currently complicated agreements have to be made between all partners
  - In many cases, too difficult for academia and SMEs

- Hardware is a critical part of security, we need to make sure it is secure
  - Being able to see what is really inside will improve security
  - Having a way to design open HW, will not prevent people from keeping secrets.
Current HW only supports security through obscurity

- Systems are built on hardware blocks where you do not know what exactly is inside
  - Open standards have proven themselves in SW
    Why should HW be any different?
  - If you really want, you can still ‘obscure’ HW, but open HW gives you a choice!
  - Many bugs, features with unintentional consequences are hiding inside HW
- Open HW will allow a larger community to verify building blocks
  - Better verification, more reliable hardware
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

ETH Zürich

Frank. K. Gürkaynak - Open Source Hardware
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
- Simul. 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
HLS
HDL
Simulator
Logic Synth.
Netlist
P&R
GDSII
PDK
Technology Provider
Foundry
Chip

IP Providers
Sim. Models
Std. Cells
Other IP
IP Providers

Technology
Provider

ASIC Design Flow and Main Actors

- Only beginning of the pipeline can be open HW
- Later stages contain business interest of various actors

High-Level Code

HLS

EDA Companies

HDL

Simulator

Logic Synth.

Netlist

P&R

GDSII

Foundry

Chip

PDK

Technology Provider

Sim. Models

Std. Cells

Other IP

IP Providers

§ 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

```
module alu

    Input logic [TRANS_ID_BITS-1:0] trans_id_i;
    Input logic alu_valid_i;
    Input logic [63:0] operand_a_i;
    Input logic [63:0] operand_a_rev;
    Input logic [63:0] [63:0] operand_a;
    Output logic [63:0] result_o;
    Output logic alu_valid_o;
    Output logic [TRANS_ID_BITS-1:0] alu_ready_o;

    // ALU is a single cycle instructions, hence it is always ready
    assign alu_ready_o = 1'b1;
    assign alu_valid_o = alu_valid_i;
    assign alu_trans_id_o = trans_id_i;

    logic [63:0] operand_a_rev;
    logic [63:0] operand_a;

    begin
        assign operand_a_rev = operand_a[63:0];
        assign operand_a = transpose([63:0] operand_a_rev);
        // generate
        for(k=0; k <= 64; k++)
            assign operand_a_rev[k] = operand_a[63-k];
        for (k = 0; k <= 64; k++)
            assign operand_a[k] = operand_a[63-k];
        // handles both signed and unsigned forms
        always_comb begin
            logic sgn;
            sgn = 1'b0;
            if (operand_a_i == SLT5)
                sgn = 1'b1;
            end
        endmodule
```

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)

  SOLDER Pad
  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 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
- SMIC
- ...

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

- Honestly, I have no idea...
- Historically it has been this way
  - At some point foundries did everything
  - Not only manufacturing, but services, IPs
  - Leftover from the time that foundries had IPs?
- Companies afraid to give up things for free
  - This is something they kept secret for so long
  - What if they give away too much (lose money?)
  - Safer to keep status quo
- Lawyers...

---

This Agreement governs any Party’s disclosure and/or receipt of Confidential Information (as defined below) and/or from one or more of the other Parties in connection with the evaluation, support and/or execution of a business relationship involving semiconductor products, processes or services (“Purpose”).

1. Confidential Information. “Confidential Information” means information a Party does not wish to be publicly known, provided such information is (a) in writing and marked “confidential” or with another similar legend; (b) in the form of a device, product, materials sample, or benchmark results derived from Confidential Information; (c) disclosed in any other manner and identified as confidential at the time of disclosure; (d) shared through ZSC’s password protected electronic portal; (e) learned as a result of a visit to any of The Companies’ or ZSC’s manufacturing facilities; or (f) ZSC’s PDK Information. “PDK Information” shall mean all information pertaining to DRC, ERC, LVS, PEX runsets, Cadence tech files, ESD and fuse documentation kits, SRAM cells, technology design manuals, Spice models and related information. For purposes of this Agreement, Zarkovia Semiconductor Company, Silicon Manufacturing Partners Pte. Ltd. and Zarkovia Semiconductor Company, Inc. and its Subsidiaries shall be referred to herein collectively as “ZSC”.

---

Frank. K. Gürkaynak - Open Source Hardware
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 verified 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 better verification, more reliable.
- Silicon verified is paramount
  - Only way of getting silicon-verified Open Hardware
- Potentially more customers
  - The more verified 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 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 verified 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
Conclusions

- Open Hardware is already available
  - Major research groups are committed to it, many companies are working on it
  - Already high quality open hardware components are used in products

- It is more complicated to have open hardware than open software
  - Licensing still borrows from open software
  - Different design flows (PCB/FPGA/ASIC) have different situations
  - There is resistance from different actors (EDA companies, IP suppliers)

- Next step is to convince foundries to make their PDK freely available
  - This is essential to allow physical open hardware (not only RTL)
  - This talk is part of this process
  - It will take time, but we will get there
QUESTIONS?

kgf@ee.ethz.ch  http://pulp.ethz.ch