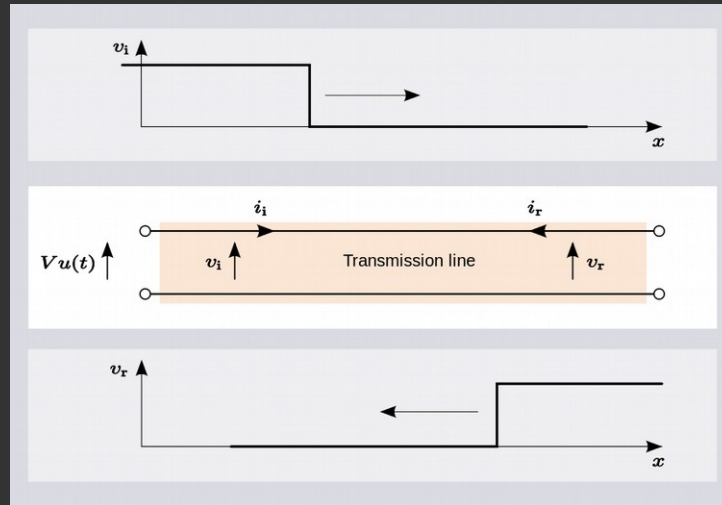


# Impedance Matching Expectations Between RISC-V and the Open Hardware Community

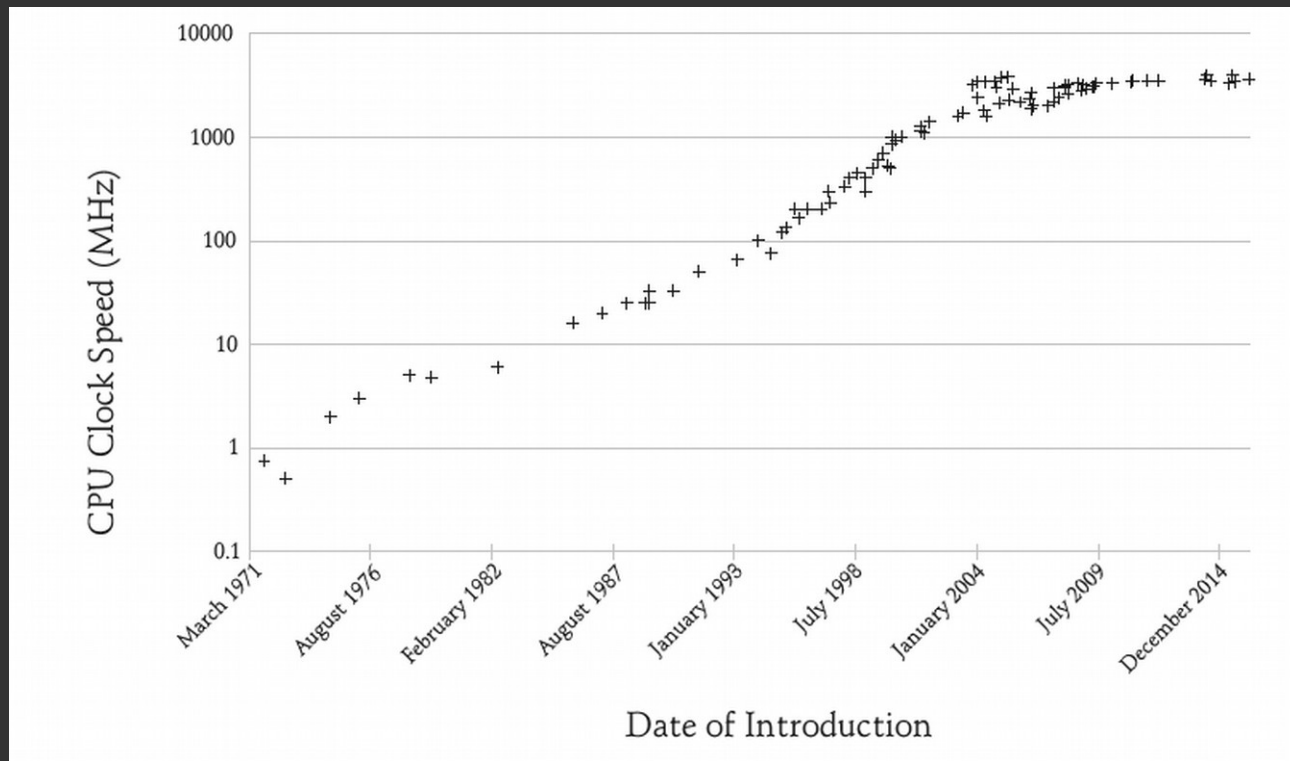
bunnie  
RISC-V Shanghai, 2017

# Why “Impedance Matching?”

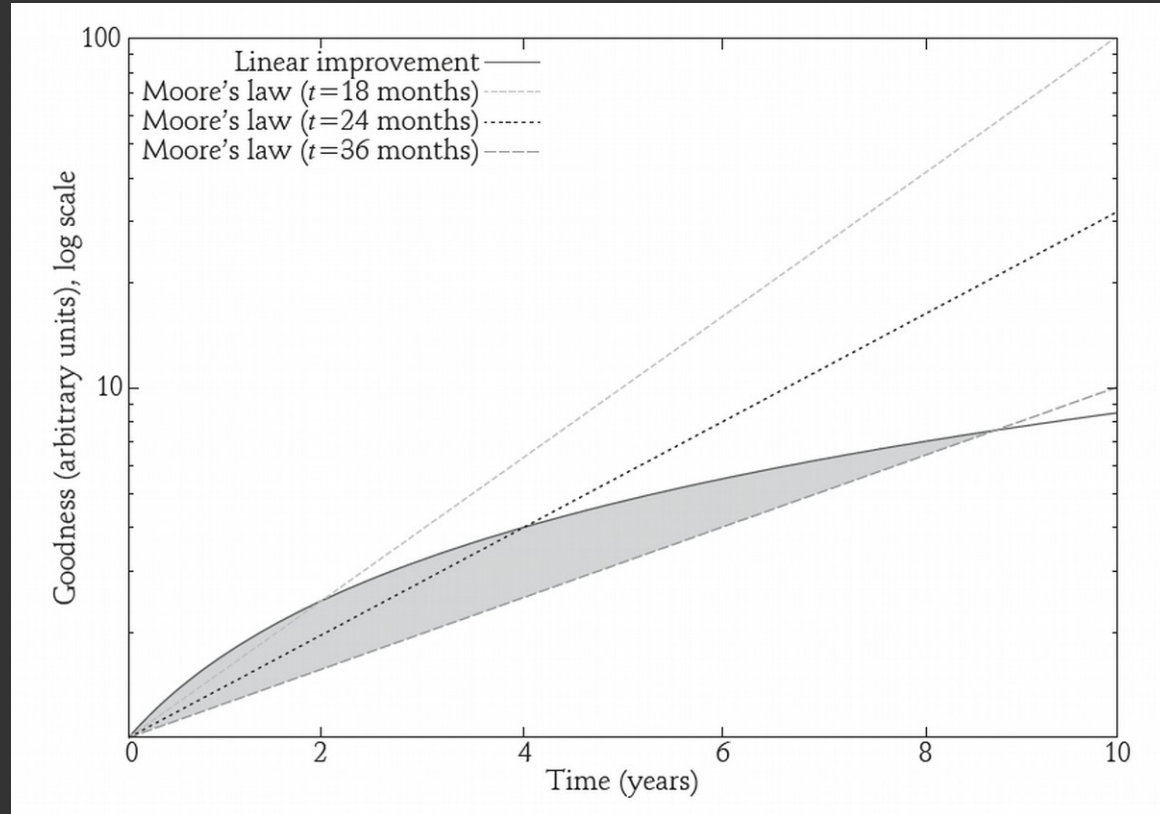
- Open silicon is a massive paradigm shift for open hardware
  - It will come faster than the user base can understand the issues
  - If the rise time is faster than the propagation time, energy gets reflected if the load isn't well-matched



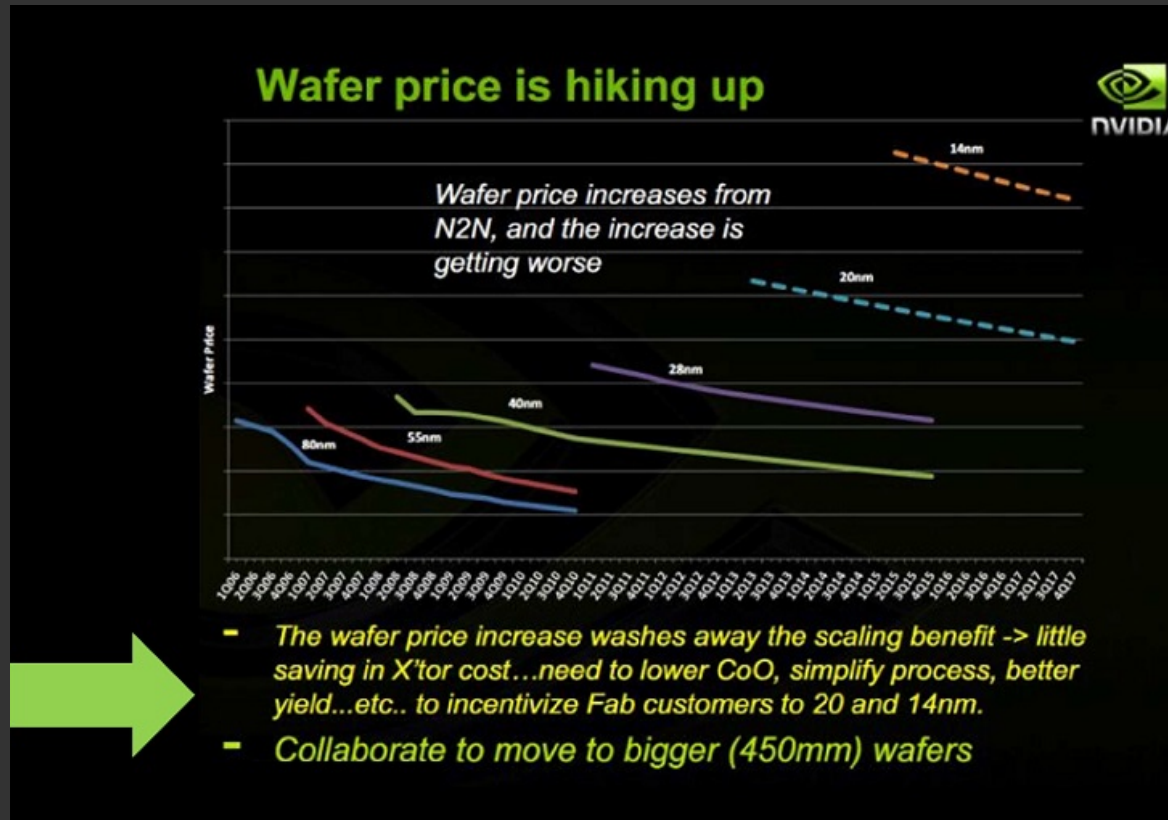
# Review: Why Open Hardware Now?



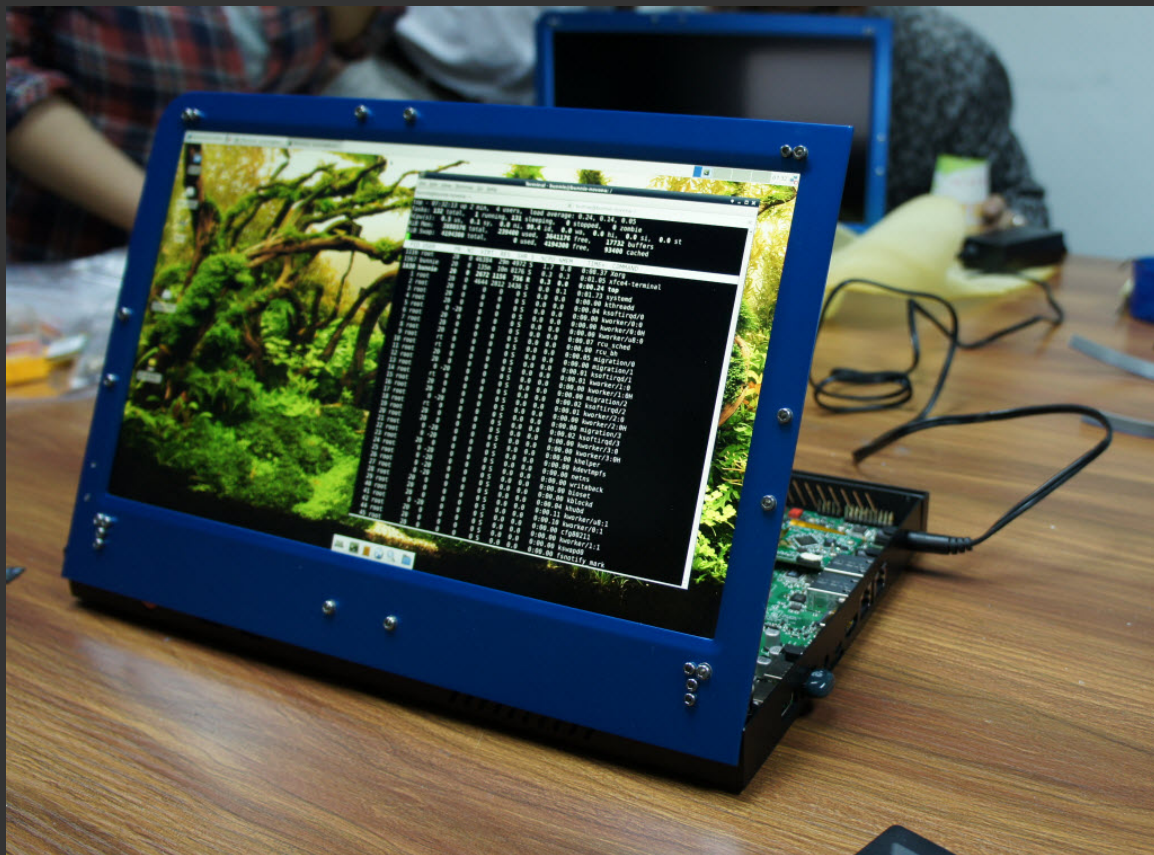
# Slower Iterations = More Time for Refinement in a Given Node



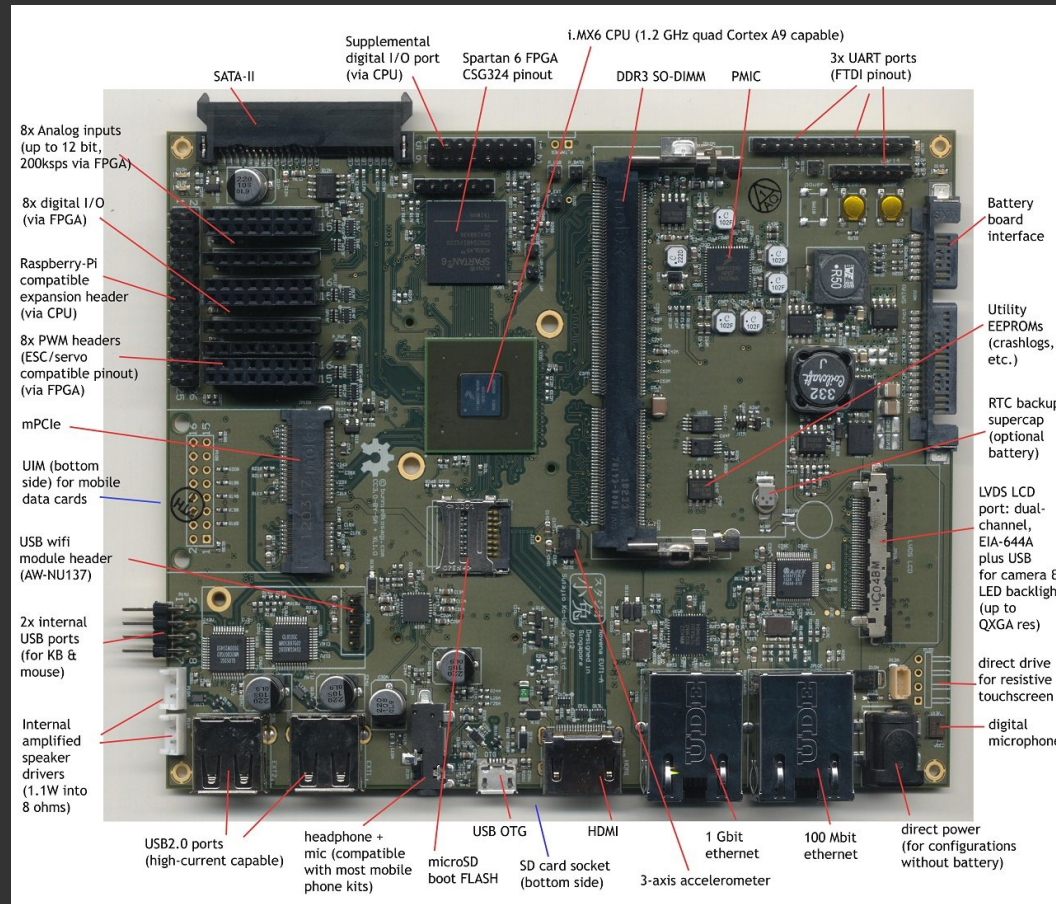
# Put in term of Process Nodes & Cost



# Case Study: Novena









Creators → Sutajio Kosagi

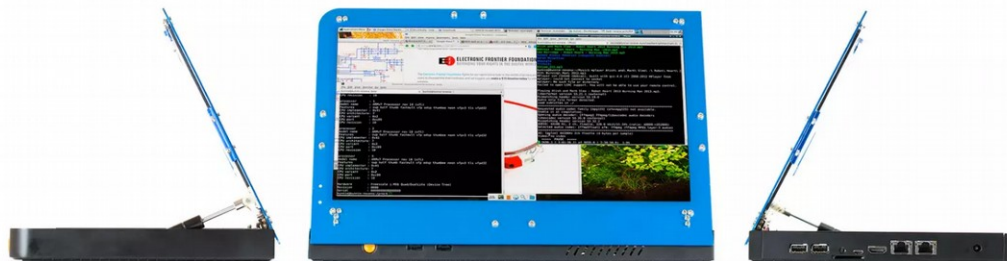
# Novena

[Home](#) [Updates](#) **26** [Backers](#) [History](#)

Singapore

Tools

Open Hardware



me@example.com

Subscribe to Updates

A new open-hardware computing platform, flexible and powerful, designed for use as a desktop, laptop, or standalone board.

As Featured In



Wired

"The project is part of larger movement towards open source hardware."

\$783,382 raised  
of \$250,000 goal

Funded!

Order Now

May 18 2014  
funded on

313%  
funded

1,114  
pledges

Support this project on social media!



Just the Board

\$550

For crafty people who want to build their case and define their own style, we'll deliver to you the main PCBA, stuffed with 4GiB of RAM, 4GiB microSD card, and an Ath9k-based PCIe wifi card. Boots to a Debian desktop over HDMI.

In Stock

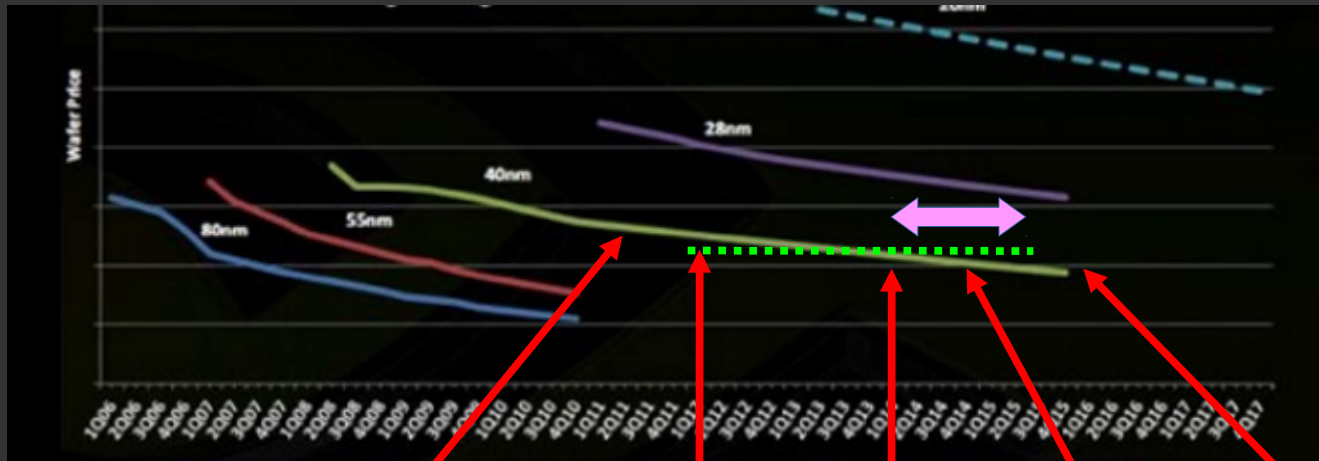
Free US Shipping / \$30 Worldwide



Add to Cart



# New Reality: 30 Months from Concept to Delivery is OK



i.MX6  
fabbed in  
40nm

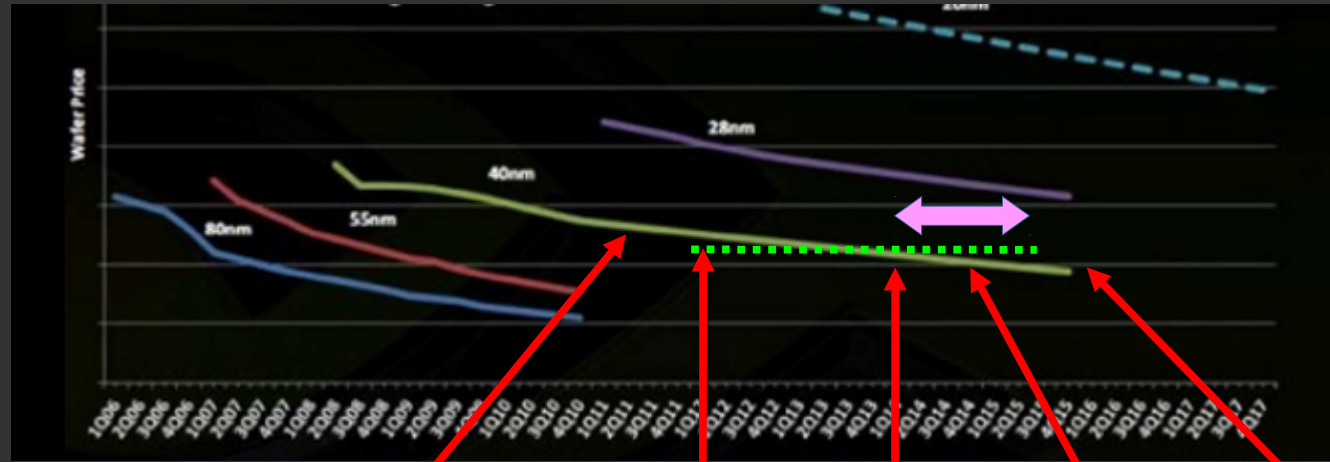
Conception

Launch

Delivery

Obsolescence  
(Pi3 @ 1.2GHz  
Quad-core ARM)

## Post-Moore



i.MX6 fabbed in 40nm

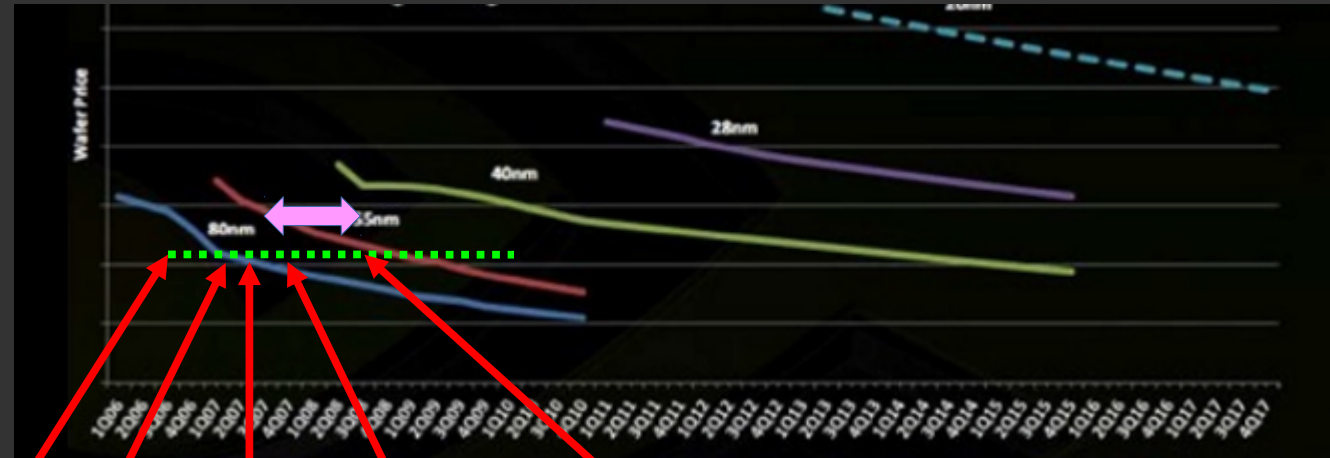
Conception

Launch

Delivery

Obsolescence (Pi3)

## Previous 30 years



Conception

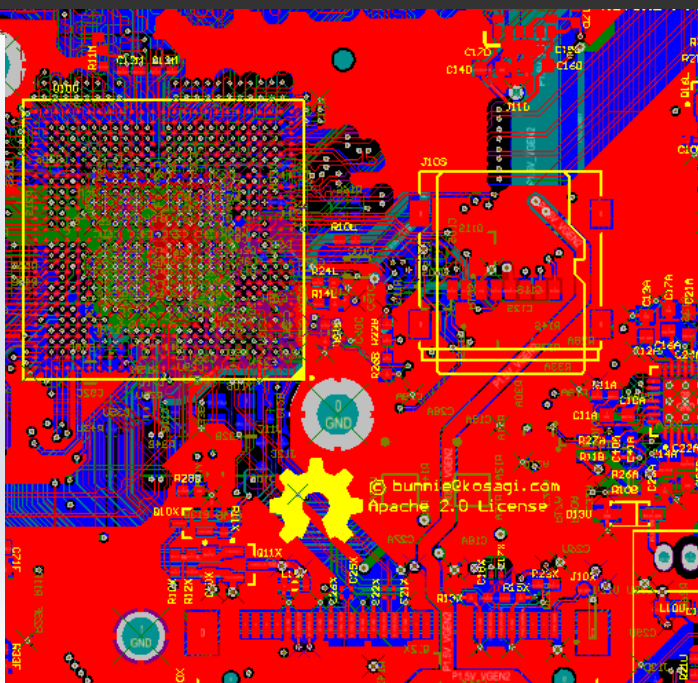
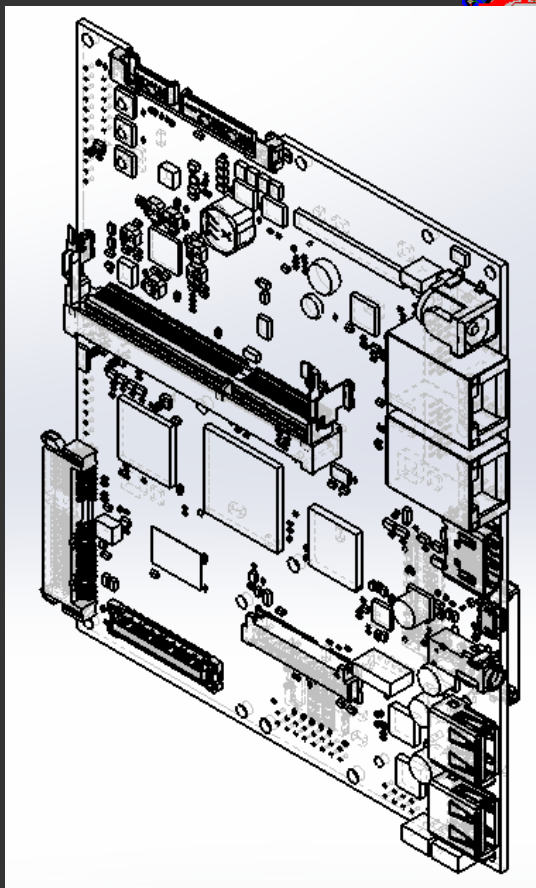
fab

Launch

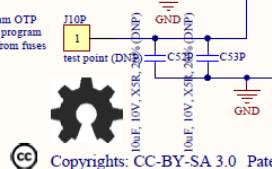
Delivery

Obsolescence

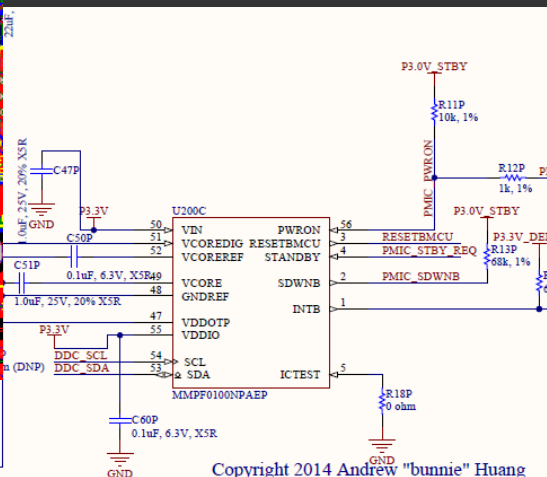
# “Open Hardware”



+5V on J10P to program OTP  
R17B, C52P, C53P to program  
C52P, C53P to boot from fuses



Copyrights: CC-BY-SA 3.0 Patents: Apache 2.0



Copyright 2014 Andrew "bunnie" Huang

Title		Revision	
Novena PVT1-E			
Size	Number		
B			
Date:	3/27/2014	Sheet	of
File:	F:\large\work\104pwr pmic SchDoc	Drawn By:	

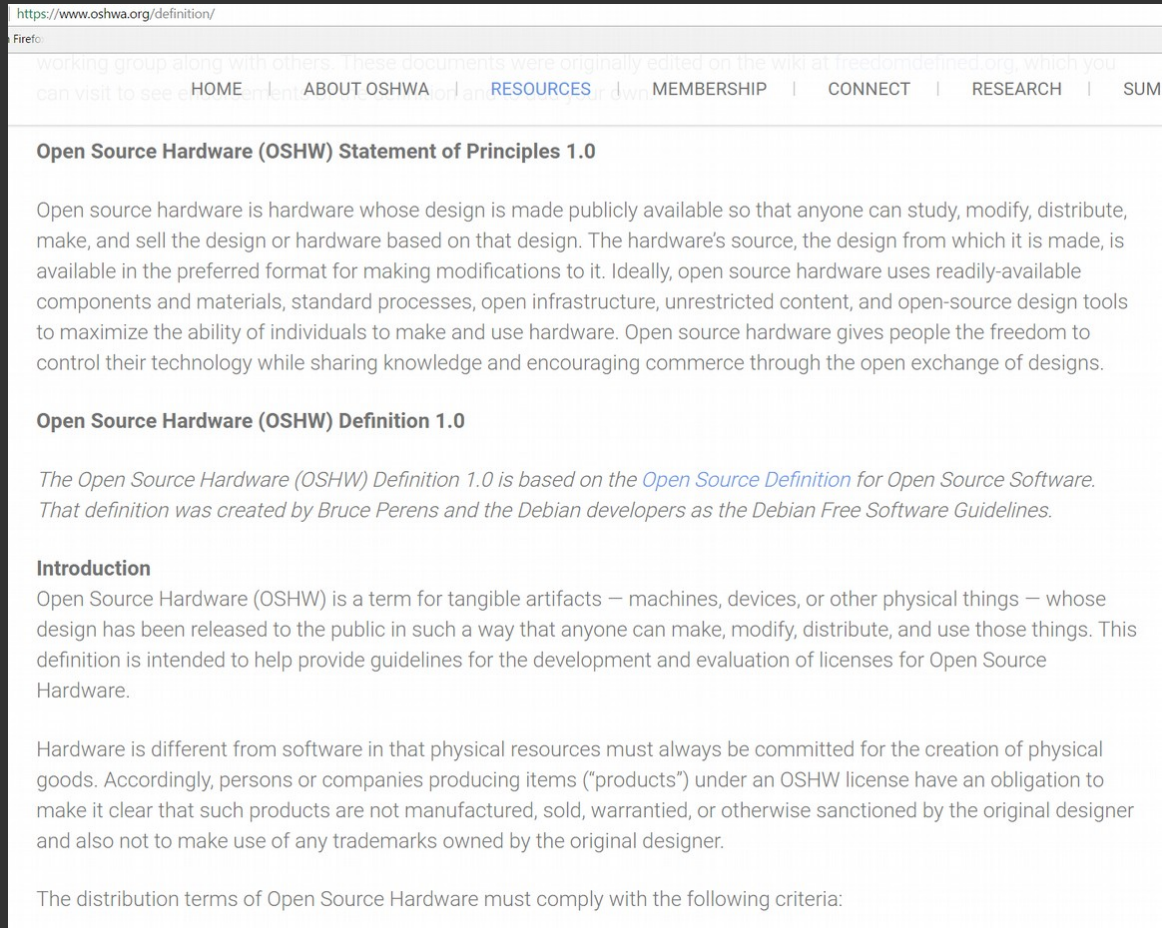


# What the Heck is Open Hardware, Anyways?



James Cridland CC-BY via Flickr

# There's an Official Definition...



The screenshot shows a web browser window with the address bar displaying <https://www.oshwa.org/definition/>. The page has a navigation bar with links: HOME, ABOUT OSHWA, RESOURCES, MEMBERSHIP, CONNECT, RESEARCH, and SUM. The main content area is titled "Open Source Hardware (OSHW) Statement of Principles 1.0". It contains a paragraph defining open source hardware, followed by a section titled "Open Source Hardware (OSHW) Definition 1.0". This section includes a note that the definition is based on the "Open Source Definition for Open Source Software" and was created by Bruce Perens and the Debian developers. Below this is an "Introduction" section that explains that OSHW is a term for tangible artifacts and that the definition is intended to help provide guidelines for the development and evaluation of licenses for Open Source Hardware. The page also mentions that hardware is different from software in that physical resources must always be committed for the creation of physical goods. Finally, it states that the distribution terms of Open Source Hardware must comply with the following criteria:

<https://www.oshwa.org/definition/>

Firefo

working group along with others. These documents were originally edited on the wiki at [freedomdefined.org](http://freedomdefined.org), which you can visit to see e

HOME | ABOUT OSHWA | **RESOURCES** | MEMBERSHIP | CONNECT | RESEARCH | SUM

## Open Source Hardware (OSHW) Statement of Principles 1.0

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. The hardware's source, the design from which it is made, is available in the preferred format for making modifications to it. Ideally, open source hardware uses readily-available components and materials, standard processes, open infrastructure, unrestricted content, and open-source design tools to maximize the ability of individuals to make and use hardware. Open source hardware gives people the freedom to control their technology while sharing knowledge and encouraging commerce through the open exchange of designs.

## Open Source Hardware (OSHW) Definition 1.0

*The Open Source Hardware (OSHW) Definition 1.0 is based on the [Open Source Definition](#) for Open Source Software. That definition was created by Bruce Perens and the Debian developers as the Debian Free Software Guidelines.*

### Introduction

Open Source Hardware (OSHW) is a term for tangible artifacts — machines, devices, or other physical things — whose design has been released to the public in such a way that anyone can make, modify, distribute, and use those things. This definition is intended to help provide guidelines for the development and evaluation of licenses for Open Source Hardware.

Hardware is different from software in that physical resources must always be committed for the creation of physical goods. Accordingly, persons or companies producing items ("products") under an OSHW license have an obligation to make it clear that such products are not manufactured, sold, warranted, or otherwise sanctioned by the original designer and also not to make use of any trademarks owned by the original designer.

The distribution terms of Open Source Hardware must comply with the following criteria:

# Let's Unpack that a Bit

- 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. The hardware's source, the design from which it is made, is available in the preferred format for making modifications to it. Ideally, open source hardware uses readily-available components and materials, standard processes, open infrastructure, unrestricted content, and open-source design tools to maximize the ability of individuals to make and use hardware. Open source hardware gives people the freedom to control their technology while sharing knowledge and encouraging commerce through the open exchange of designs.



# Let's Unpack that a Bit

- 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. The hardware's source, the design from which it is made, is **available in the preferred format for making modifications to it**. Ideally, open source hardware uses readily-available components and materials, standard processes, open infrastructure, unrestricted content, and open-source design tools to maximize the ability of individuals to make and use hardware. Open source hardware gives people the freedom to control their technology while sharing knowledge and encouraging commerce through the open exchange of designs.

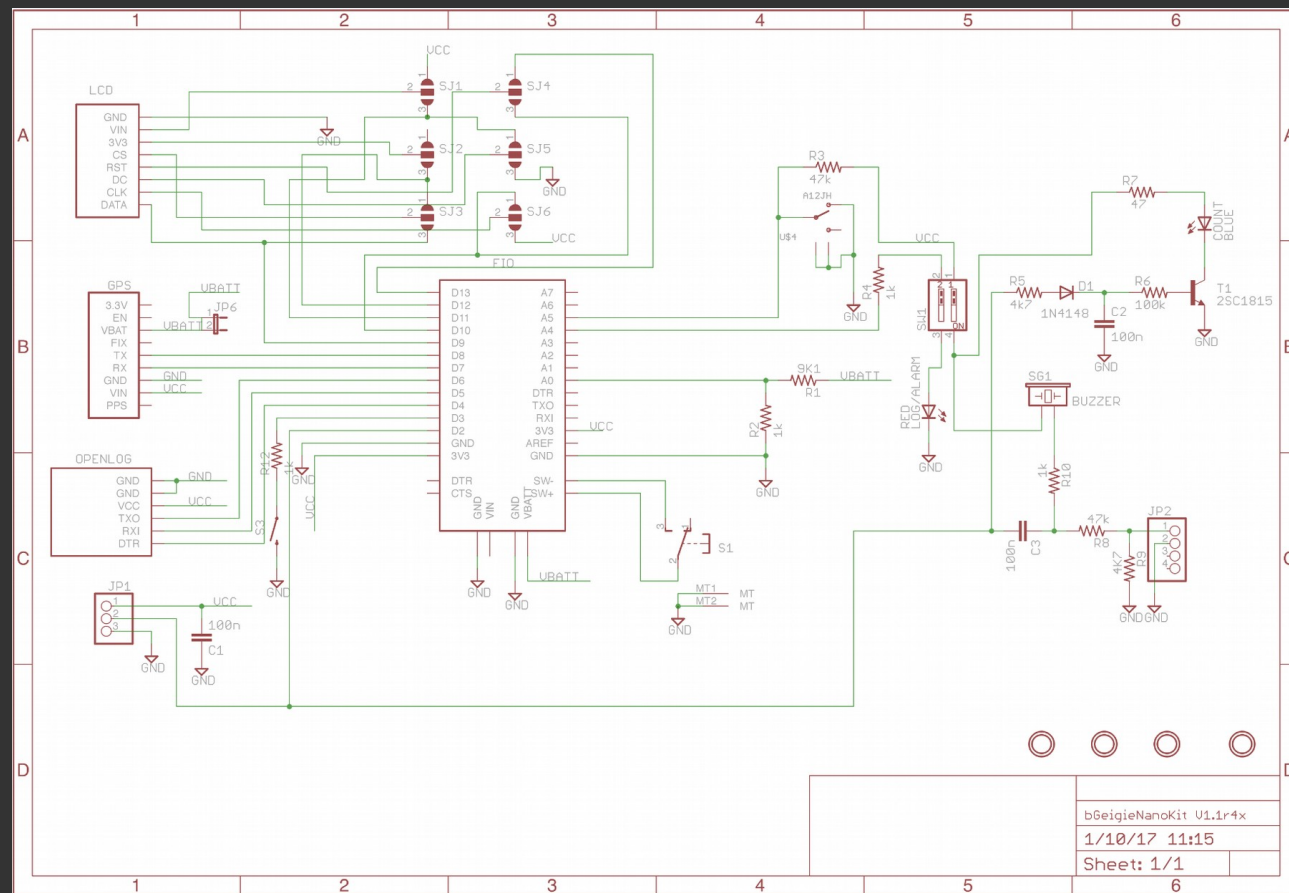
# Let's Unpack that a Bit

- 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. The hardware's source, the design from which it is made, is available in the preferred format for making modifications to it. Ideally, open source hardware uses **readily-available components and materials, standard processes, open infrastructure, unrestricted content, and open-source design tools** to maximize the ability of individuals to make and use hardware. Open source hardware gives people the freedom to control their technology while sharing knowledge and encouraging commerce through the open exchange of designs.

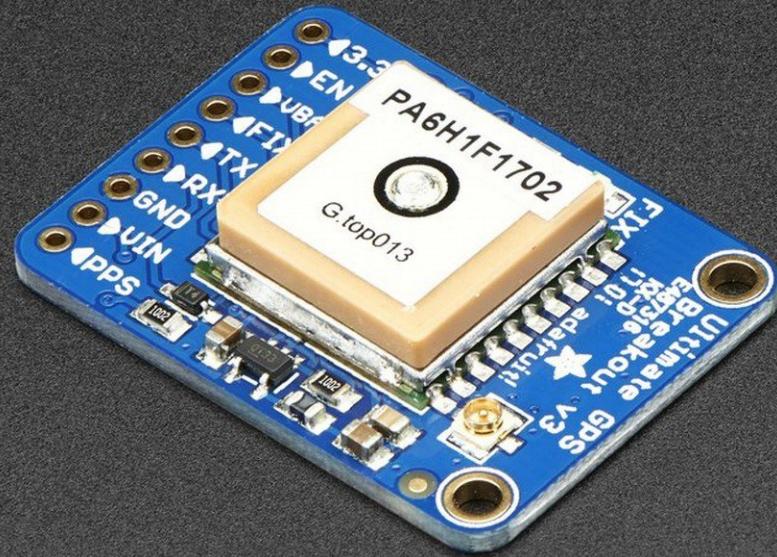
# Let's Unpack that a Bit

- 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. The hardware's source, the design from which it is made, is available in the preferred format for making modifications to it. Ideally, open source hardware uses readily-available components and materials, standard processes, open infrastructure, unrestricted content, and open-source design tools to maximize the ability of individuals to make and use hardware. Open source hardware gives people the **freedom to control their technology while sharing knowledge and encouraging commerce** through the open exchange of designs.

# What it Means in Practice...

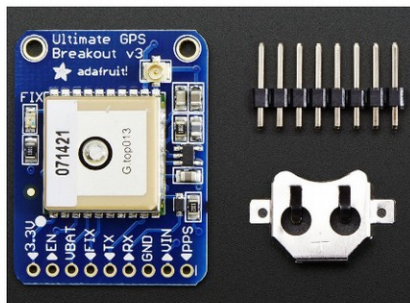












## Adafruit Ultimate GPS

One GPS to rule them all and in the darkness find them!

[Overview](#)

[Pinouts](#)

[Direct Computer Wiring](#)

[Arduino Wiring](#)

[Parsed Data Output](#)

[SENSORS / GPS](#)



## Downloads

by [lady ada](#)

## Files

- [MTK3329/MTK3339 command set sheet](#) for changing the fix data rate, baud rate, sentence outputs, etc!
- [LOCUS \(built-in-datalogging system\) user guide](#)
- [Datasheet for the PA6B \(MTK3329\) GPS module itself - used in version 1 of this module](#)
- [Datasheet for the PA6C \(MTK3339\) GPS module itself - used in version 2 of this module](#)
- [Datasheet for the PA6H \(MTK3339\) GPS module itself - used in version 3 of this module](#)
- [MT3339 GPS PC Tool \(windows only\)](#) and the [PC Tool manual](#)
- [Mini GPS tool \(windows only\)](#)
- [EagleCAD PCB files on GitHub](#)
- [Fritzing object in the Adafruit Fritzing Library](#)

## Ultimate GPS v3 Schematic



# GlobalTop

## PMTK command packet

The document is the exclusive property of GlobalTop Tech Inc. and should not be distributed, reproduced, or any other format without prior permission of GlobalTop Tech Inc. Specifications subject to change without prior notice

### GlobalTop Tech Inc.

No.16 Nan-ke 9th Rd Science-based Ind. Park, Tainan 741-47, Taiwan, R.O.C.  
Tel:+886-6-5051268 Fax:+886-6-5053381 <http://www.gtop-tech.com/> email: sales@gtop-tech.com  
Copyright © 2012 GlobalTop Tech Inc. All right reserved.

- FCC E911 compliance and AGPS support (Offline mode : EPO valid up to 14 days )
- RoHS Compliant

1 Reference to GPS chipset specification

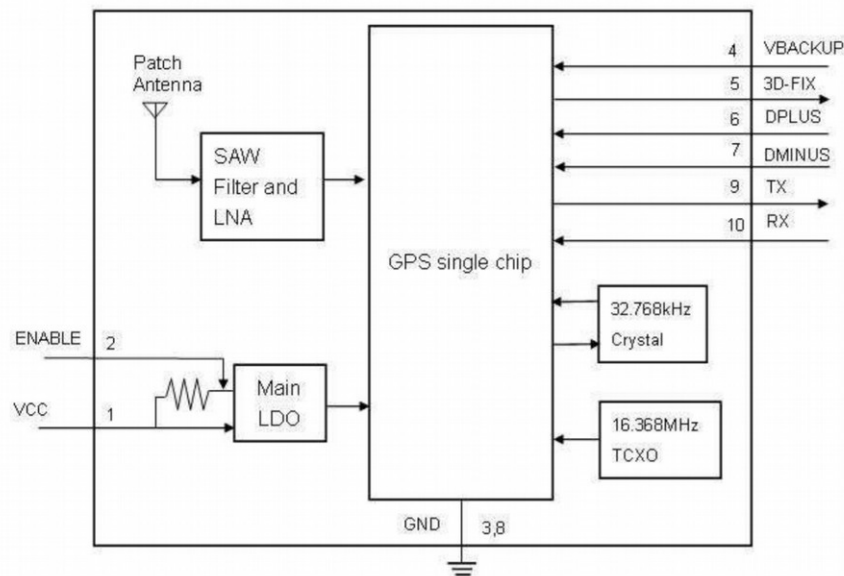
2: SBAS can only be enabled when update rate is less than or equal to 5Hz.

This document is the exclusive property of GlobalTop Tech Inc. and should not be distributed, reproduced, into any other format without prior permission of GlobalTop Tech Inc. Specifications subject to change without prior notice.

**Copyright © 2011 GlobalTop Technology Inc. All Rights Reserved.**



## 1.3 System Block Diagram





## MT3339 GPS All-in-One Solution Data Sheet

Version: 1.05  
Release date: 2011-09-19

© 2011 MediaTek Inc.

This document contains information that is proprietary to MediaTek Inc.

Unauthorized reproduction or disclosure of this information in whole or in part is strictly prohibited.

Specifications are subject to change without notice.

# Case Study: Open Source as a Marketing Term

**CROWD**SUPPLY

BROWSELAUNCHABOUT US

Search

Creators → [Purism](#)

San Francisco, CA  
[Hackable](#)

## Librem 13: A Laptop That Respects Your Rights

HomeUpdates **19**BackersHistory

**\$456,954** raised  
of \$250,000 goal

**Funded!****Order Now**

Sep 17 2015  
funded on

182%  
funded

314  
pledges

Support this project on social media!

[f](#)[t](#)[p](#)

**Librem 13 Laptop****\$1,399**

Orders placed now ship Apr 30, 2017.

Free US Shipping / \$80 Worldwide

# What Appeals to Users

## Philosophy

1. Purism will only use free/libre and open source software in the kernel, OS, and all software.
2. Purism will design and manufacture hardware that respects users' rights to privacy, security, and freedom.
3. Purism will prioritize privacy, security, and freedom for our users.
4. Purism will not discriminate against persons nor groups nor fields of endeavor.
5. Purism will source, and manufacture the highest quality hardware.

Purism will donate a portion of proceeds to free software projects on a quarterly basis, according to Purism's own [Free Software for Freedom Margin Share Program](#).

## Hardware Companies, Freedom, and Privacy

Description	BIOS Freed	Kernel Freed	OS Freed	Software Freed	Promotes Freedom	Promotes Privacy†	Supports Freedom
ACER	No	No	No	No	No	No	No
AAPPLE	No	No	Partial*	No	No	No	No
DELL	No	No	No	No	No	No	No
FUJITSU	No	No	No	No	No	No	No
GOOGLE	Almost‡	No	No	No	Partial*	No	Partial*
HP	No	No	No	No	No	No	No
IBM	No	No	No	No	No	No	No
MICROSOFT	No	No	No	No	No	No	No
LENOVO	No	No	No	No	No	No	No
PURISM	Almost‡	Yes	Yes	Yes	Yes	Yes	Yes
TOSHIBA	No	No	No	No	No	No	No

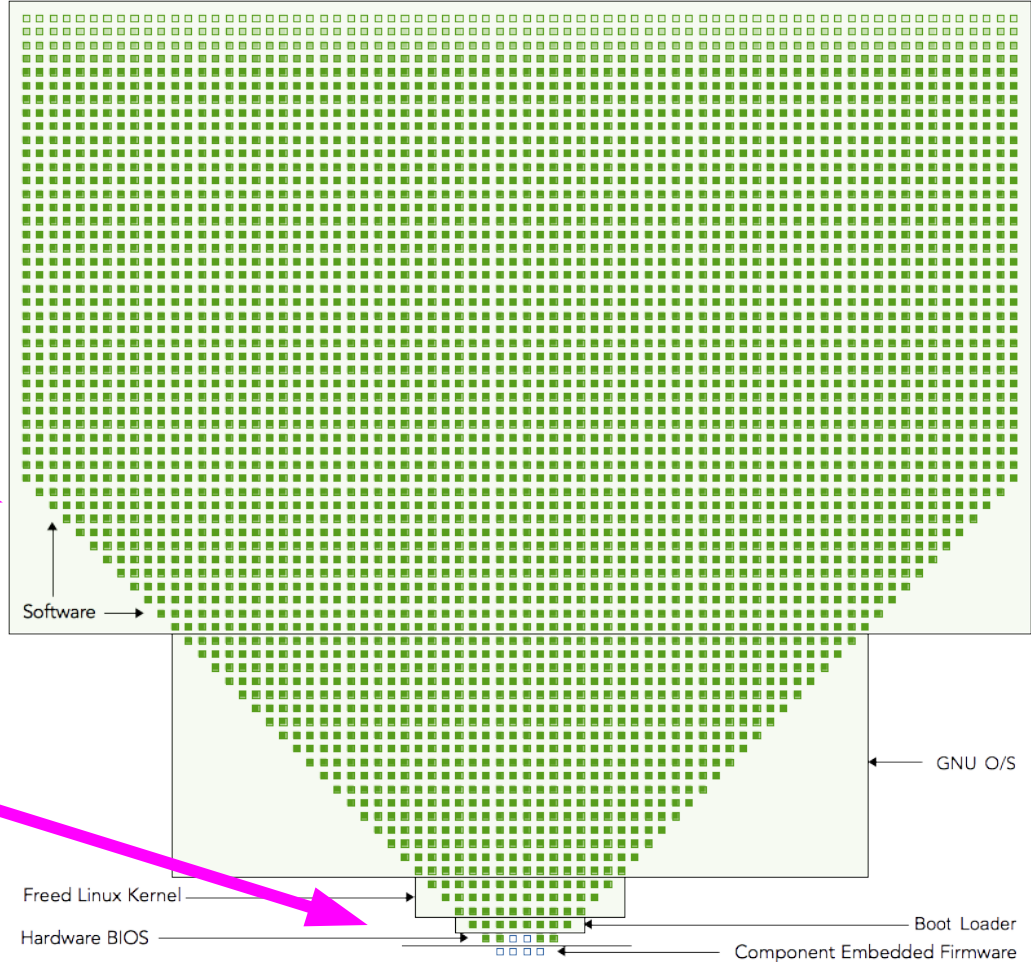


Despite all the efforts of  
the open source community...

Herein lies the problem:  
The trust root is still closed.

Purism Librem 15 Laptop: Graphical representation relative to size of the completely freed kernel,  
operating system, and all software applications.

Completely Freed ■  
Binary Firmware □





# Conspiracy Fears Stoke Demand for Transparency

TechRepublic. SEARCH Q Big Data Security Innovation More Newsletters Forums Resource Library Tech Pro

SECURITY

## Is the Intel Management Engine a backdoor?


Is Intel's Management Engine a backdoor for security groups and hackers, or just a feature created to aid businesses? Jack Wallen dives in and draws his conclusions.


→ ↻ www.eteknix.com/expert-says-nsa-have-backdoors-built-into-intel-and-amd-processors/ Apps Imported From Firefo

## Expert Says NSA Have Backdoors Built Into Intel And AMD Processors

NETWORKWORLD FROM IDG

Home > Security

 **GEARHEAD**  
By Mark Gibbs, Contributing Editor, Network World | JUN 18, 2016 3:41 PM PT

About :  For more than 30 years, Gibbs has advised on and developed product and service marketing for many businesses and he has consulted, lectured, and authored numerous articles and books.

## Intel Management Engine's security through obscurity should scare the \* \* \* \* out of you

Intel's latest x86 chips contain a top secret control subsystem that you can't audit, control, or disable ... what could possibly go wrong?

# Result: Standard for Transparency is Higher for “Computers” than “IoT”

From: Richard Stallman <rms@gnu.org>★  
Subject: **Laptop needs nonfree software**  
To: Me <bunnie@kosagi.com>★

When I read about your plans for an "open laptop", I hoped it would be a platform that we could use without nonfree software.

Unfortunately, it seems that this is not so. The GPU, which is in the same chip as the CPU and can't be removed, requires a firmware blob. That program is not free software (and not open source either).

--  
Dr Richard Stallman  
President, Free Software Foundation  
51 Franklin St  
Boston MA 02110  
USA  
[www.fsf.org](http://www.fsf.org) [www.gnu.org](http://www.gnu.org)  
Skype: No way! That's nonfree (freedom-denying) software.  
Use Ekiga or an ordinary phone call

Dec 2012

From: Richard Stallman <rms@gnu.org>★  
Subject: **Re: Entire OS buildable from source?**  
To: Me <bunnie@kosagi.com>★  
Cc: rms@gnu.org★

[[[ To any NSA and FBI agents reading my email: please consider ]]]  
[[[ whether defending the US Constitution against all enemies, ]]]  
[[[ foreign or domestic, requires you to follow Snowden's example. ]]]

One thing I was wondering about is if you think we should apply for an RYF certification, and if so, how should we go about doing it.

We can't certify it under present circumstances because it has hardware elements (at least the video coprocessor) that can't run without a nonfree program. The reason we chose this criterion is that such a machine will systematically tend to draw people to install the nonfree firmware, even though that is not shipped with the machine.

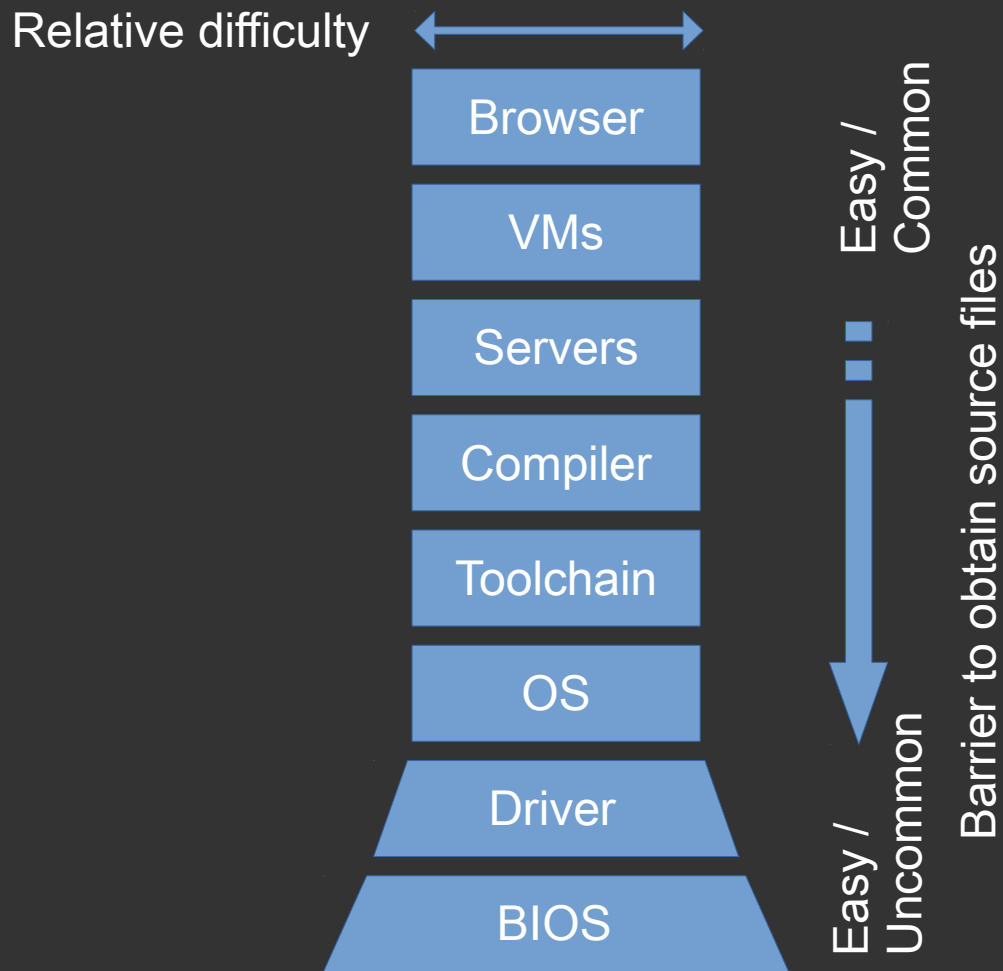
I really want to find someone to do the reverse engineering so we can fix that problem and then certify it.

--  
Dr Richard Stallman  
President, Free Software Foundation  
51 Franklin St  
Boston MA 02110  
USA  
[www.fsf.org](http://www.fsf.org) [www.gnu.org](http://www.gnu.org)  
Skype: No way! That's nonfree (freedom-denying) software.

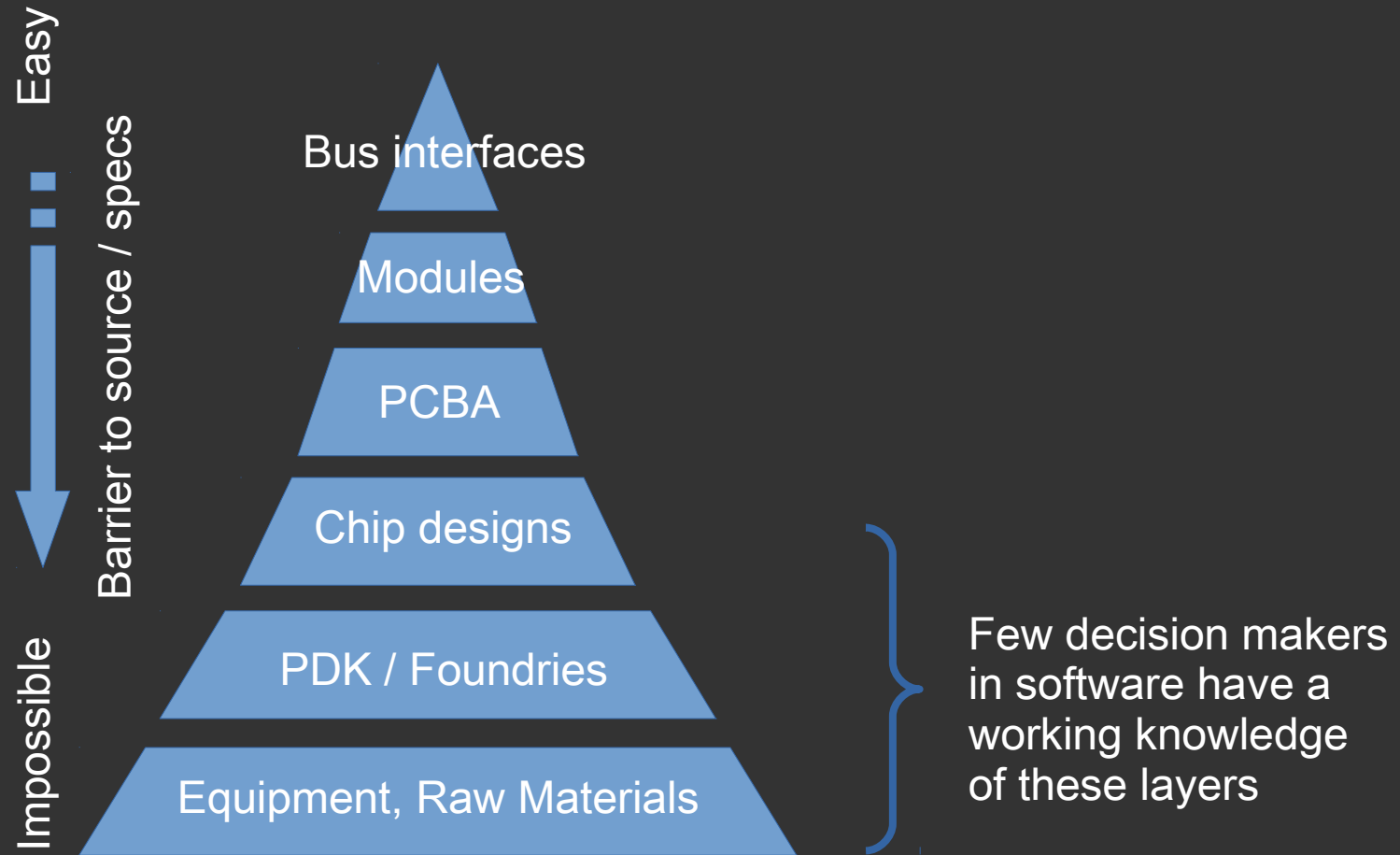
Apr 2014

(dozens of emails later, including an offer to build a custom version that has a fused-out GPU but the VPU was still viable...)

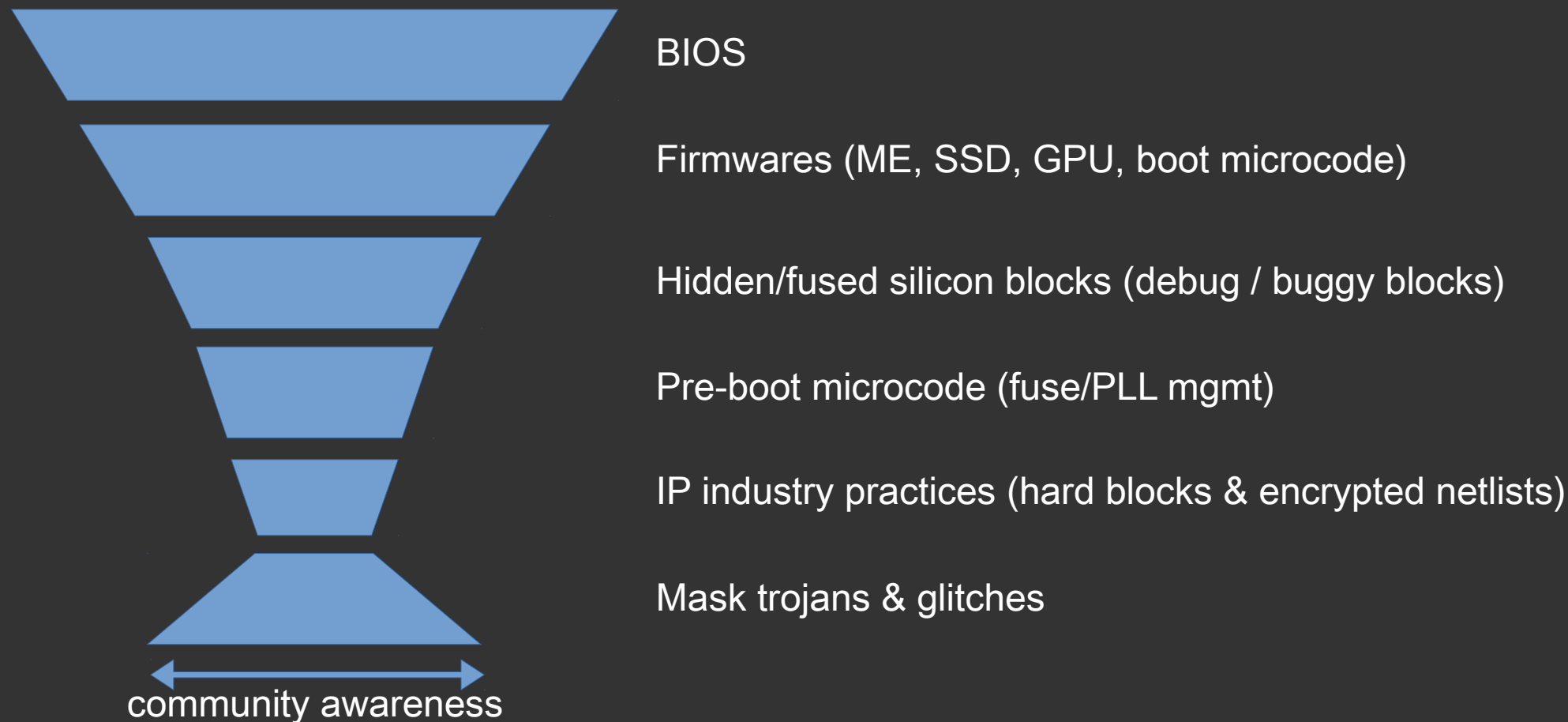
# Transparency is Easy, Right?



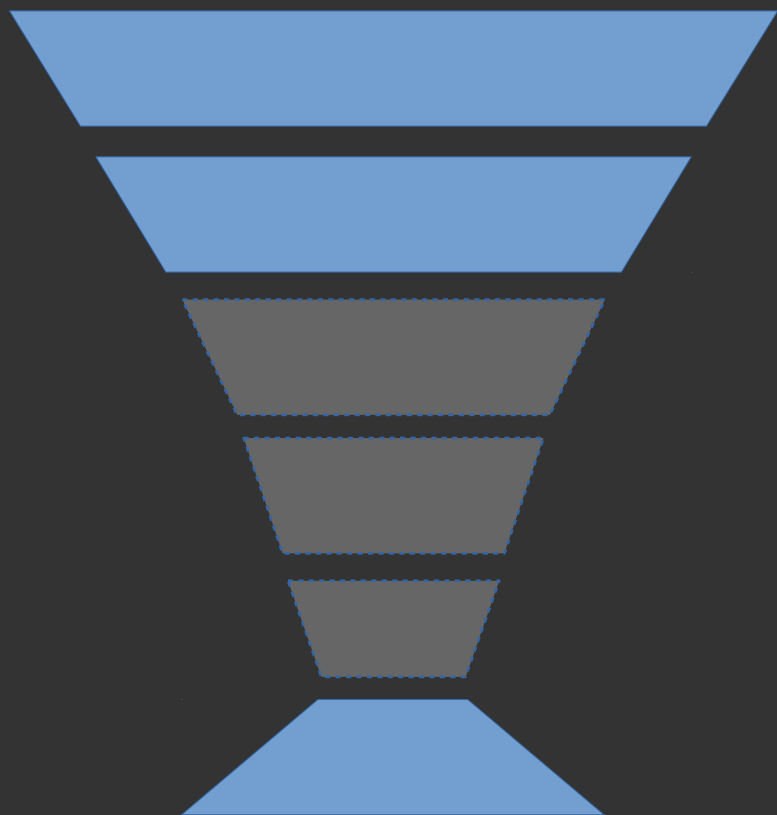
# ...Not So in Silicon



# Community Awareness of Trust Issues



# The Knowledge / Expectation Gap



BIOS

Firmwares (ME, SSD, GPU, boot microcode)

Hidden/fused silicon blocks (debug / buggy blocks)

Pre-boot microcode (fuse/PLL mgmt)

IP industry practices

Mask trojans & glitches

Little awareness  
of imminent threats

Extremely difficult to  
validate / verify, especially  
in cutting-edge processes



# Key Point

- Open Silicon vendors bear a burden to educate system-level decision makers
  - 1) What are the realistic, imminent threats?
  - 2) How does open silicon addresses these issues?
  - 3) What are the practical economic factors that limit transparency?

# The Limits of Transparency in Silicon

- Post-Novena, investigated doing a very simple, 8- or 16-bit CPU using only open source tools
  - Inspired by Visual 6502 project
  - Use something like Magic/XCircuit/IRSIM + Yosys/qrouter for design
  - Fab in MOSIS 0.18um or 0.35um (SCMOS rules)
  - “Totally inspectable” trust root
    - I have a SEM, image layers and confirm construction
- Major problem: no open source FLASH IP
  - Defeats the idea of having a “totally inspectable” trust root when you can’t inspect the code store!

## [The Visual 6502](#)

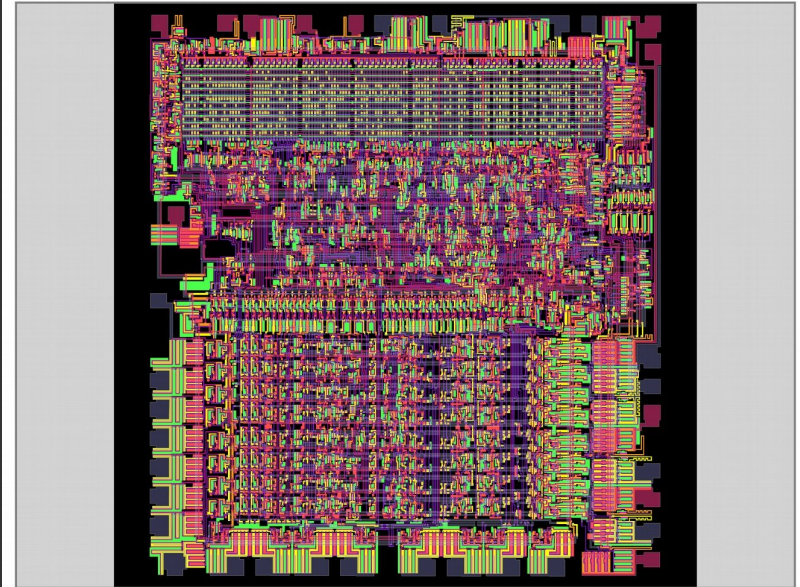
[FAQ](#) [Blog](#) [Links](#)

This simulator uses HTML5 features only found on the latest versions of browsers and needs lots of RAM. If you have trouble, please [check](#)

Keyboard controls: 'z' to zoom in, 'x' to zoom out, 'n' to step the simulation.

Mouse controls: Left-click and drag to scroll around (when you're zoomed in.)

More information in the [User Guide](#).



# IP Industry & Lack of Transparency

- IP blocks & PDKs tend to be opaque or strictly NDA
  - Fab industry is highly competitive
  - PDK elements (including blocks such as SRAM, fuses, FLASH, DRAM) are valuable, difficult to engineer, yet hard to protect
  - High-speed, mixed-signal designs (PLL, CDR, PHY) are valuable, difficult to engineer, also hard to protect
  - Spec-compliance is tough (PCI, USB, ISA (ARM/x86/RISC-V)), yet once the RTL is spec compliant it's easy to copy and compile
- Development barriers are measured in millions to billions of dollars on cutting-edge processes
  - Not remotely comparable to barriers found in software
  - IP licenses are extremely lucrative, often times costing more than the masks

# The Security Nightmare

- Conspiracy: What if key IP providers are compelled to put back doors in IP blocks?
  - A back door in PCI-express, USB cores from Synopsys...
  - A back door in TrustZone, or perhaps even the CPU implementation?
- Realistically: a set of benign escapes, but put together forming a major hardware security breach
  - Trust root in the ISA is great, but worthless if your IP blocks can stomp on data
- How to turn this into an opportunity for RISC-V?

# Compromise:

## Good Fences Make Better Neighbors?

- Recap: a key benefit of openness is the ability to “understand everything inside the hardware”
  - But IP practices within the industry prevent that from being even remotely true
- Proposal: Hardware introspection blocks





# Introspection: “Hardware ASSERT”

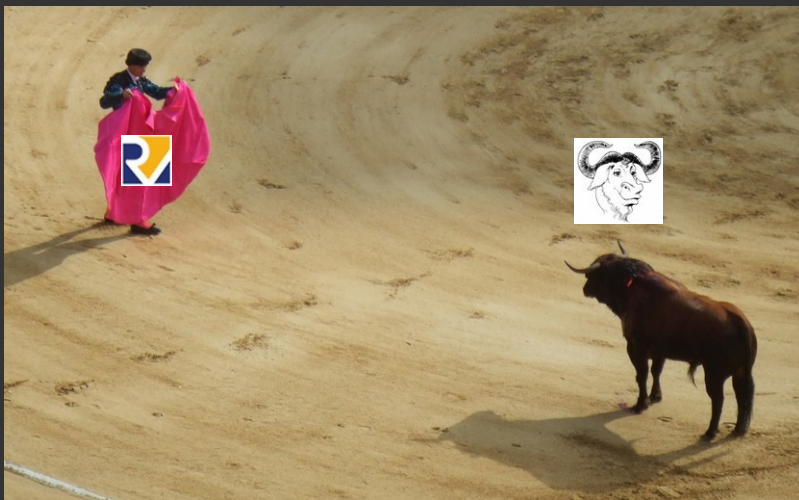
- “Hardware ASSERT statement” IP block
  - “Fence in” opaque IP blocks to certain memory ranges
    - Like a PMP, but not for user process – for 3<sup>rd</sup> party hardware IP
  - Log or trigger on certain transactions
  - Use TAP/BIST infrastructure to configure
  - Primarily protects against
    - Hidden/extra/undocumented registers in opaque IP blocks
    - Monitoring IP blocks which can originate write/read transactions

# Introspection: Storage Validation

- There are some “trivial” mask-edit attacks
  - Masked ROM
  - “Biasing” SRAM/register cells on reset
- Open RTL TAP/BIST readout of fuses, ROMs, SRAM, and security-critical reset values
  - Verification of content & function
  - Done 100% outside of the closed IP blocks
  - Ideally at full clock speed (to avoid reduced clock detection & spoof)

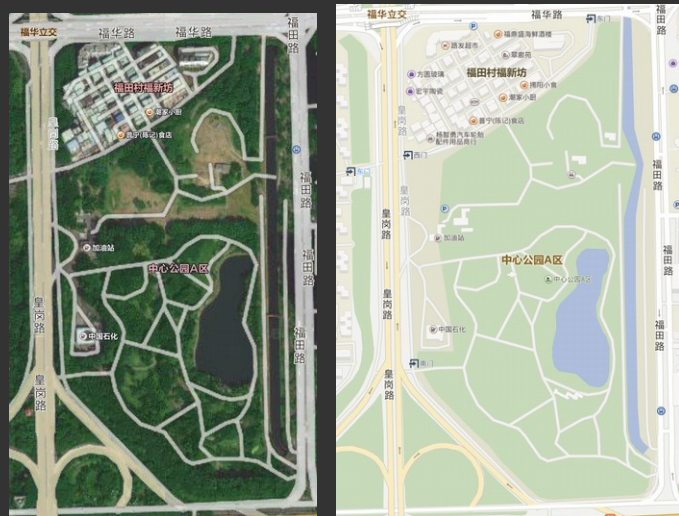
# IC Inspection: It's Hard, Why Even Bother?

- Temptation: tell users “trust me it's too hard, don't bother trying”
  - Pattern: security people have been breaking stuff that's “too hard” for decades
- Solution: make it their problem to solve
  - What's needed is an abstract map of design intent to compare against
  - No need to reveal process-specific details, e.g. phase shift techniques, tiling, etc.



# Mask Inspection Compromise

- Full mask inspection not possible due to PDK confidentiality
- Compromise: share M2 or M3 and up, plus outlines of standard cells within key open-RTL regions?
  - BEOL M3 and up has less secret sauce
    - Share an abstract representation of metal layers (not GDS-II but a list of metal line centroids)
    - Think “map” vs. “satellite image”
  - Don’t reveal standard cell library layouts, or hard macros
    - Difficult to introduce major logic changes at M2 and below
    - Rule out compiler/RTL injection back door
      - Detect extra data pathways for spoofing, copying data; extra instructions in ISA
      - Detect extra RAM/ROM
      - Cannot detect swapping one logic gate for another (e.g. AND→ NAND transformation)
        - Use RTL structural + synthesis techniques + BIST introspection to harden against this?
  - Random sampling SEM validation of e.g. introspection blocks
    - Perhaps M3 or higher pattern comparison is sufficient and reasonably priced
    - Lower metal can be done at a premium for high-value silicon
    - Random sampling N needs to be higher if multiple copies of chip are in reticle
    - Validation focuses primarily on trust root/introspection blocks



# Introspection: Recap

- Assuming RISC-V implementations are willing to be 100% open on self-generated RTL, including disclosing pre-boot config & fusing...
- Three hardware introspection/inspection techniques to work around IP/transparency issues in silicon industry:
  - 1) ASSERT blocks – fence & log
  - 2) Open TAP verification of black-box memories
  - 3) BEOL / M2 or M3+ abstract “street map” availability



# Mismatched Impedances: Risk of Backfire

- Big fan of what SiFive is doing for open hardware
- Worried that expansive claims risk drawing criticism from the open hardware community

**CROWD**SUPPLY

BROWSE

LAUNCH

ABOUT US

## Open-source RTL!

The FE310 is the first open-source RISC-V SoC available in industry. SiFive has contributed the FE310 RTL code to the open source community. That means you can see what's inside the chip and completely understand how the hardware works.

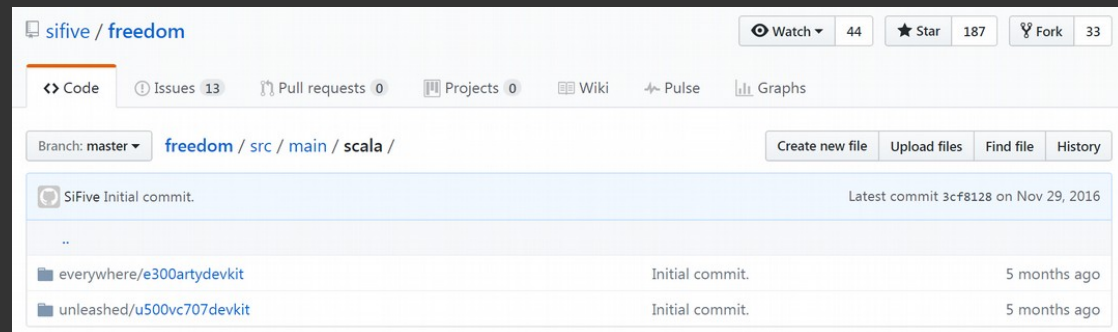
Take a look: [SiFive GitHub](#)

By releasing the RTL code, SiFive wants to encourage open source development of both software support for RISC-V as well as promote open hardware development.

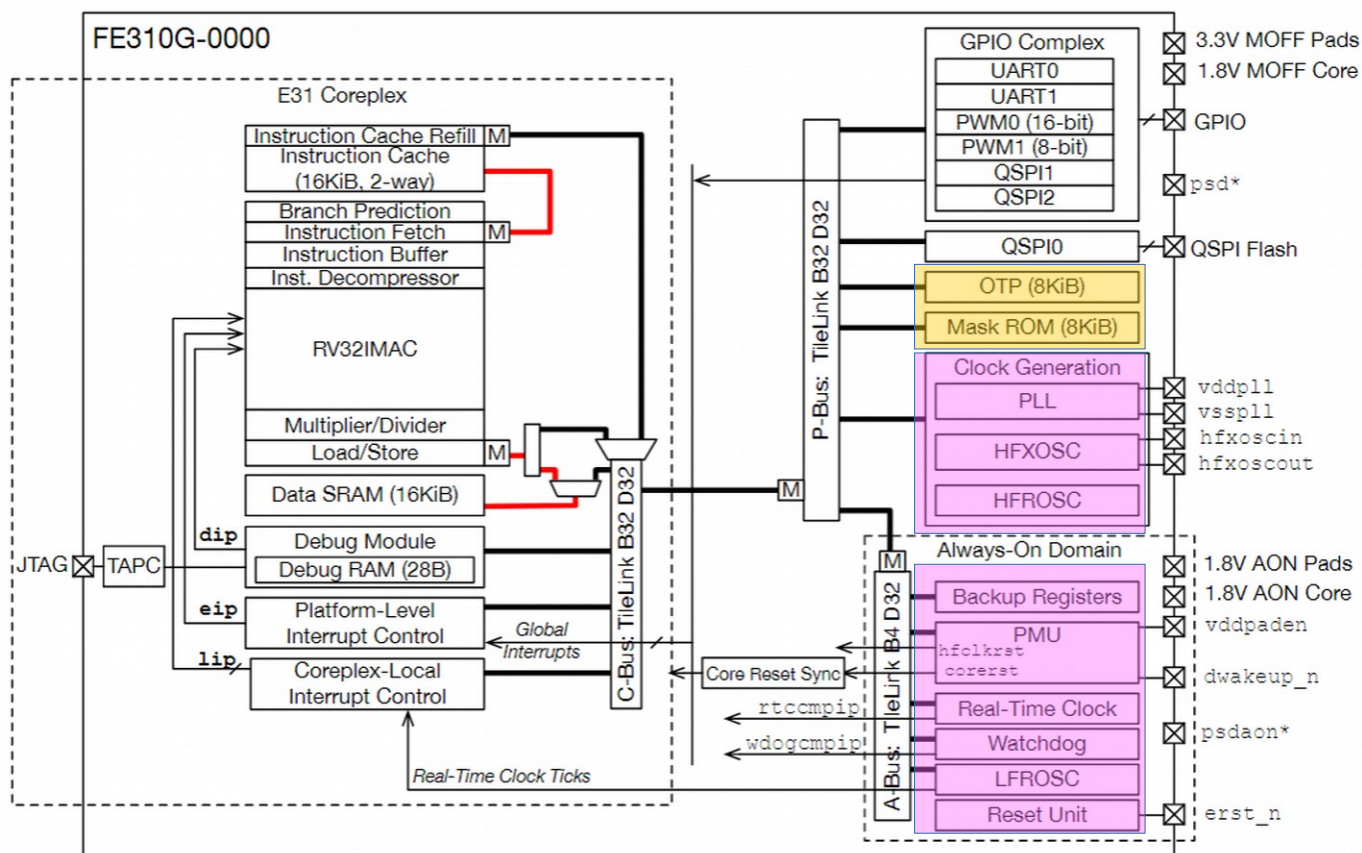
The RTL code also empowers chip designers with the ability to customize their own SoC on top of the base FE310. For system architects, developers, or companies without chip design capabilities, SiFive's "chips-as-a-service" offering can customize the FE310 to meet their unique needs.

# Impedance Mismatches

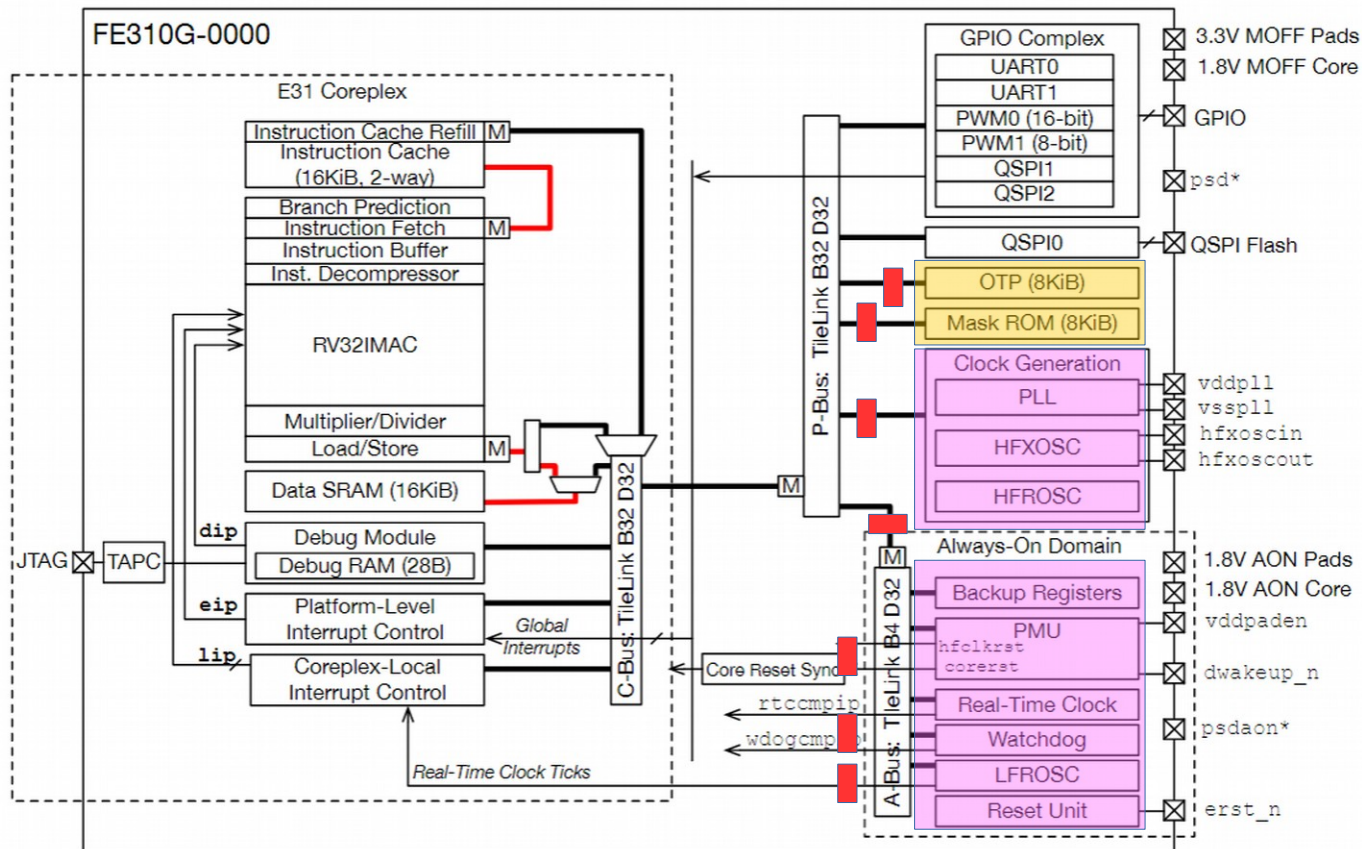
- “see what’s inside the chip and completely understand how the hardware works”
  - PLL and fuse blocks are black boxes
    - Unfortunately, these are two *very* interesting black boxes from a hardware security standpoint
- “SiFive has contributed the FE310 RTL code to the open source community...Take a look: [SiFive at GitHub]”
  - Github repo link doesn’t yet contain the FE310 code?



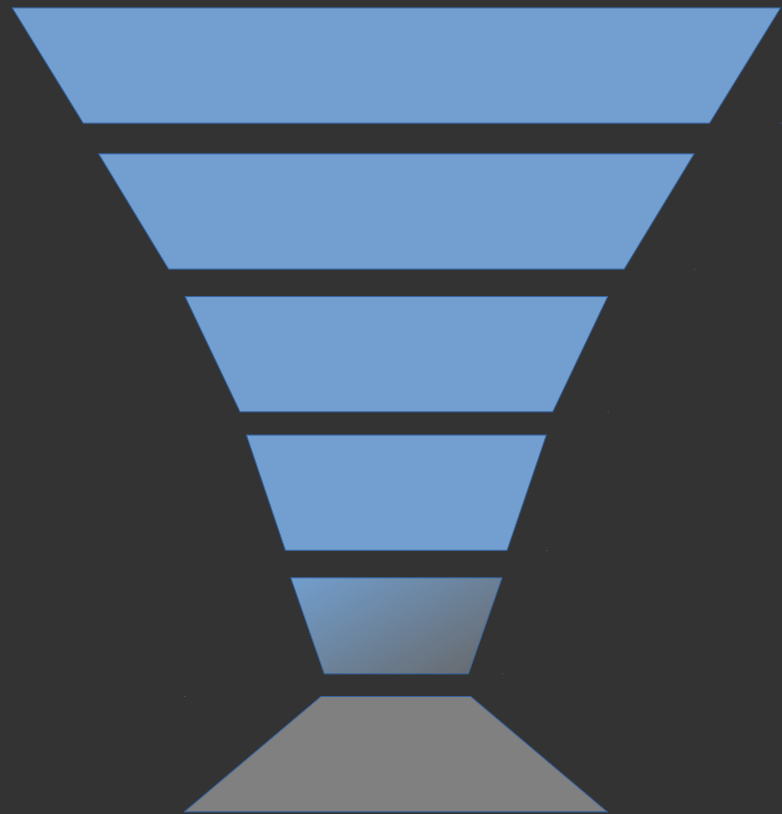
# Better to Be Explicit and Clear



# ...and Add Fences?



# Spell it Out for Non-Silicon Designers



Everyone  
else



RISC-V



BIOS

Firmwares (ME, boot microcode)

Hidden/fused silicon blocks

Pre-boot microcode (fuse/PLL mgmt)

IP industry practices

Mask trojans & glitches



# RISC-V Market Targets in Open Source Hardware

- Security / Trust Roots
  - High-value segment
  - High standard of scrutiny
    - Push audit costs to users, e.g. BEOL metal inspection as a cost-adder, services for toolchain/probe setup
  - Protectable advantage vs. rest of industry – closed vendors can't compete
  - Relatively low-performance, low IO requirement
- Open Source / Libre Movement Zealots
  - Premium laptops and servers – RYF/FSF certified
    - Basic certification is no blobs, and a promotion of user freedoms
  - Scrutiny proportional to security claims
    - Transparent disclosure of closed IP blocks + hardware introspection probably acceptable
    - BEOL inspection probably not necessary
  - High performance, IO requirements

# Features for the Performance Segment

- Wide, fast ECC memory, and lots of it
  - $\geq 64\text{GiB}$ ,  $\geq 2$  channels, DDR4
- Privileged architecture w/hypervisor, security, core ISA extensions
- Always-on PMU complex (speed throttling + sleep/standby, clock tree management, thermal sensing)
- Interrupt + systimer complex
- Debug UART
- Transparent boot process & hardware introspection
- BEOL mask set for inspection
- One wide (x16) PCIe bus for graphics card
  - IOMMU to allow large memory apertures
  - Most window managers *require* 3D graphics for acceptable performance
  - The Libre community has already drawn battle lines on acceptable practices for discrete graphics solutions – avoid integrated 3<sup>rd</sup> party IP cores
- A couple narrow PCIe busses for peripheral expansion; # of busses traded off with peripheral integration level
  - USB2.0 (5x for HID features – can stub out via hub)
  - USB3.0 (2x for laptop, more for server)
  - SATA-3 (2x for laptop, more for server)
  - At least 2x PCIe x1 busses available for network connectivity (wifi + ethernet)

} Any combo of PCIe vs. integrated IP OK; discrete peripherals a plus in terms of hackability. Could imagine a bay of M2.NGFF slots inside a laptop.

# Main Point: Performance Segment

- Enthusiasts drive margins & buzz
  - e.g. “overclockers” / “gamers” in the performance segment – \$\$\$ for GHz and FPS
  - Open/Libre enthusiasts have a similar elasticity in price points – \$\$\$ for Transparency and Freedom
    - Has overlap with the system-level decision makers, key developers
- Less 3<sup>rd</sup> party IP in SoC is a “feature” in the open hardware market
  - Fewer black boxes in the silicon
  - Market would bear higher system costs for discrete peripherals
  - System-level modularity is a marketable feature
- Quick path to raise RISC-V awareness among early adopter crowd
  - But to market as an “open” CPU, the openness aspect *must be done right*, or else you may get the opposite effect

# Impedance Matching: Recap

- “Open Hardware” definition means different things to different users
  - Intelligent user base, but only partially educated on core issues
  - Common values:
    - Open hardware means hackable – any user can download, inspect, mod
    - Open hardware means freedom – freedom of speech, freedom to make and use, non-discrimination
    - Open hardware means transparent – “no black boxes”, *but* need to clarify up to which abstraction barrier
- Impedance matching will require finding a common ground and agreeing upon values and terminology
  - Transparency is a potential key selling point, but:
    - Never claim to eliminate all threats – you can’t
    - Define a threat model, and how you mitigate that threat
- Of course, open hardware is a minority market for RISC-V
  - Purely economic, not ideological arguments e.g. displacing ARM cores
  - But: strong overlap between tool developers & open source enthusiasts

Q&A