

The Open Hardware Initiative

J. Serrano

BE-CO Hardware and Timing section
CERN, Geneva, Switzerland

CERN, 22 June 2010



Outline

- 1 Requirements
- 2 Choices
- 3 Open Hardware
 - Introduction
 - Business models
 - Legal issues
 - The Open Hardware Repository



Outline

- 1 Requirements
- 2 Choices
- 3 Open Hardware
 - Introduction
 - Business models
 - Legal issues
 - The Open Hardware Repository



Desirable HW features of a distributed control system

1/2

The good things of custom HW

- Function is exactly what you need.
- Can change easily if you find a bug. Or have it changed!
- Peer review. Potential for really good designs.
- Not tied to a single company (you never know).

The good things of commercial HW

- Designed, built and tested by someone else (resource gain).
- Immediately available.
- Hardware proven by many users in many different applications.



Desirable HW features of a distributed control system

2/2

Modular

- Re-use components easily.
- Have different people in an organization do what they do best.

Interconnect!

- Allows to build distributed systems easily.
- Based on communication standards.
- Good sync capabilities. Transparent common notion of time.



Outline

- 1 Requirements
- 2 Choices
- 3 Open Hardware
 - Introduction
 - Business models
 - Legal issues
 - The Open Hardware Repository



Hardware development

VITA 57 FPGA Mezzanine Cards (FMC)

- The only standard to fit our needs.
- High-bandwidth BGA connector for multi-Gb/s links.
- VME and PCIe carriers in the pipeline.
- Xilinx kits have FMC slots → convenient for testing.

White Rabbit network

- Sub-nanosecond sync for ~ 1000 nodes over typical fiber lengths of 10 km.
- Based on Ethernet.
- Multi-site collaboration.



Gateway and software

Some initial ideas

Gateway

- Internal bus is Wishbone-based (open standard with IP cores available).
- Try to automate repetitive code through scripts.
- Auto-discovery of Wishbone cores by Linux kernel would be nice.

Software (very preliminary ideas!)

- Define Wishbone as a bus in Linux.
- Write Linux modules and interconnect them by a driver representing the whole board.
- Integration into official kernel desirable.



Outline

- 1 Requirements
- 2 Choices
- 3 Open Hardware**
 - Introduction
 - Business models
 - Legal issues
 - The Open Hardware Repository



Open Hardware: our definition

Publish everything needed to review

Specifications, discussions, schematics and layouts in some human-readable format, HDL, etc. Publish universally, no NDAs.

Publish everything needed to modify

Schematics and PCB layout files for your favorite EDA tool. Unfortunately the best ones are neither free nor free. . .

Publish everything needed to produce

Manufacturing files, bill of materials, etc.



Advantages

Peer review

Get your design reviewed by experts all around the world, including companies!

Design re-use

How many people are designing a 100 MS/s ADC independently, making the same – or different – mistakes?

Healthier relationship with companies

No vendor-locked situations. Companies selected solely on the basis of technical excellence, good support and price.



Getting organized

“It’s all about interfaces”, Bob Dalesio

Your piece of HW can speak to others if you can agree on a set of interfaces. Examples (currently in OHR) include Ethernet, VME, PCIe, FMC...

Design compromises

The price to pay for sharing (and saving time and money) is to choose sub-optimal technical solutions from time to time. We did not choose to write an OS more suited to our needs than Linux, did we?



Role of companies

Design partners

Pay a company specialized in a given topic to design a specific card with/for you.

Commercial partners

Buy the cards you designed from a company that will take the charge of manufacturing, testing, managing stocks and providing support.



Some business model examples for commercial partners

IBM-style

Become part of a larger OH team, fully respecting OH practice. Sell full systems based on OH kit.

Red Hat-style (kind of)

Sell manufacture, test and support of individual boards along with a guarantee. Participate in design if needed.

Oracle-style

Support OH kit and build a closed solution on top with added value.



Licensing

A quick landscape tour

Hardware is not like software

- Copyright protects the expression of an idea, not the idea itself.
- For a schematic (and even HDL), GPL is easily bypassed.

Options

- OHL (viral). If you take my design and use it, you promise not to sue me for patent infringement.
- BOHL (viral). Design files are not released.
- MIT/BSD (non-viral). Do what you like, don't blame me in case of problems.



Licensing

Our thoughts so far

LGPL for HDL

- It's very easy to turn a “used in” into a “connected to” situation in HDL, so GPL would not help.
- We do want to be informed and profit if our cores are improved.

MIT/BSD-style for the rest

- Not clear how OHL, BOHL and others would perform in court. And don't want to find out!
- Viral licenses scare some of our potential commercial partners. Could do more harm than good.



What about free riders?

Free riders are fine

People and companies who take open designs and do not contribute anything back do not pose a problem to us.

But what about *mean* free riders?

If somebody takes OH and uses it to build a closed solution for a profit, that is fine as well, but we would not be clients.



Mad patent disease and patent trolls

See <http://www.eetimes.com/showArticle.jhtml?articleID=216600017>

“In this climate, many fear being charged with willfully infringing patents or omitting prior art in patent applications, a charge known as inequitable conduct. So Intel and other companies have put strict procedures in place to control which patents its engineers can read.”

Opening up your designs does make you more vulnerable to this disease.



Open Hardware Repository: <http://www.ohwr.org>

A very useful tool

A web-based collaborative tool for electronics designers.

Made itself of open software

- Redmine for wiki and task/issue management.
- Sympa mailing list manager.
- SVN/GIT for version management (integrated in Redmine).

Other possible uses

- Traceability for Technology Transfer departments.
- Prove prior art with UTC time stamps in SVN, GIT, wiki...



Conclusions

Open Hardware looks like a good idea so far

- We can get the best of the custom and COTS worlds.
- We are learning a lot, even electronics! ;)
- Definitely more fun than closed HW.

Some things not completely clear yet

- Legal framework, work in progress with CERN's KTT group.
- We still need a clear collaboration model with companies.

First HW due end of 2010, stay tuned!

