OPEN HARDWARE AND COLLABORATION

J. Serrano, CERN, Geneva Switzerland

Abstract

Open Source Hardware (OSHW) follows the lead of Free and Open Source Software (FOSS) and has similar goals: ensuring developers can share their work without artificial hurdles, improving quality through peer review, avoiding vendor lock-in and providing for a fair playground in which projects can thrive and accommodate contributions without compromising their long-term future. The paper introduces OSHW and then attempts to answer a number of questions: (i) what are the perceived benefits and issues of OSHW, in general and in the context of public research facilities? (ii) what is new with respect to FOSS? (iii) what makes OSHW projects succeed or fail? The paper uses real examples of OSHW projects and practice throughout – mostly CERN-related because they are as good as any other and well known by the author – and concludes with some thoughts about what the future holds in this domain.

INTRODUCTION

What is Open Source Hardware? The word *"open"* is often abused. Some people consider it good marketing to assign this label to a product they have designed, without much concern to what is actually meant by it. Having identified this issue quite early on, the Open Source Hardware Association (OSHWA) published an official definition [1] of Open Source Hardware (OSHW) in 2010. It is inspired by the definition of Open Source Software [2] and the four freedoms of Free Software [3]. As such, it focuses on the freedoms granted to users of OSHW designs, which include the right to study the design documents, modify them, share the modifications, build hardware based on the designs, and commercialize the results. The definition is important insofar as it enables efficient communication by clarifying what OSHW actually means.

With the definition under our belts, let's move on and discuss the OSHW phenomenon. More and more designers share their design documents on the Web, using licenses which comply with the OSHW definition. Why? Because designing non-trivial hardware has become easier in the last years, opening this domain to a larger subset of the population. Things which are easy and useful to do have a special relationship with freedom. If a world power decided to forbid travel to Jupiter, few people would complain. If, on the other hand, Spanish citizens were suddenly forbidden to cross the French border, there would understandably be strong protests. This is why it is important to guarantee that people who want to share their designs can do so within a robust legal framework which guarantees a number of important freedoms.

As we will see throughout this article, much of what is happening in OSHW and many of the issues we confront in its practice have a direct counterpart in the Free and Open Source Software (FOSS) world. The democratization of access to computing in the late 80's and early 90's explains to a large extent the emergence of Free Software. We see here again the need to provide a solid conceptual and legal setting for a freedom when it becomes easy to exercise by many. FOSS is a reference and a source of inspiration for many OSHW practitioners.

There are many examples to illustrate the revolution in empowerment that has happened in hardware design in the last few years. Consider the design of a GPS watch. It used to require the efforts and talents of a small or medium-size enterprise. Today it can be done by a few friends staying after work over a few months [4], even if they restrict themselves to using only FOSS tools in the development. The availability of simple, powerful, ready-made, modular electronic boards has enabled an even bigger community of integrators. They use hardware from companies such as Adafruit, Sparkfun and Arduino – which have themselves grown to a considerable size – and add simple customizations to generate a variety of innovative products.

For reasons of legitimacy and focus, this article deals mostly with the particular case of electronics development in the framework of big public scientific institutions, but most of its reasoning and conclusions should apply directly elsewhere. The existence of very successful commercial and non-commercial OSHW organizations and projects tells us that this paradigm is useful. We try to see what contexts favor these developments and take a critical look at perceived advantages and disadvantages of OSHW.

PERCEIVED BENEFITS

As one would expect, the perceived benefits of FOSS apply quite clearly to OSHW:

Reuse. People don't spend their time and money reinventing the wheel, and those resources then become available to add quality to a project and innovate. Reusing existing designs is also a very powerful risk management technique. Of course, proprietary design can also be reused, but in a necessarily reduced scope. Proprietary designs benefit from fewer pairs of eyes scrutinizing them and finding bugs. They also tend to be more geared to solving a particular problem and are therefore harder to use in a new setting. Finally, in a design reuse scenario, proprietary designs often need dedicated and costly legal efforts to guarantee the licensee is granted a sufficient degree of freedom.

Avoidance of vendor lock-in. All other things being equal, vendor lock-in is never in the interest of the user of a given technology. OSHW makes lock-in virtually impossible. On the other hand, vendor lock-in is often part of the commercial strategy of proprietary design

ISBN 978-3-95450-189-2

companies. For some institutions, such as the military, lock-in can even result in concerns about security.

Dissemination. Some public institutions, such as CERN, have it in their official mandate to maximize their impact on society. One important component of this impact is the use of the technologies developed at CERN in different settings, beyond the realm of High Energy Physics research. The invention and dissemination of the HTTP protocol is a paradigmatic case. Inspired by FOSS and by the standard scientific research paradigm, CERN has actually applied these ideals in the domain of experimental data [5] and scientific publications [6]. The extension to OSHW is natural for institutions that want to maximize the dissemination of hardware designs.

Motivation and recognition. This works for individuals as well as private and public institutions. It is often more interesting and motivating to work on technology which others will find useful. Developers normally prefer to work in teams than to work in isolation. OSHW naturally induces collaboration. It is also a great way to gain a reputation based on one's work. For companies, recognition and prestige translate almost immediately into monetary rewards, so they often consider them an important part of their overall commercial strategy.

As we will see later, the materialization of the benefits above does not follow automatically and necessarily from the practice of OSHW. The latter is rather a catalyst for these benefits to happen in an otherwise well-planned, sensible project.

Benefits of OSHW as seen from the perspective of commercial companies depart slightly from those seen in the FOSS world. The standard arguments apply quite well. OSHW provides a level playing field in which big and small commercial actors can compete freely. Proprietary technologies tend to favor winner-takes-all situations because small players trying to innovate cannot capitalize on a rich basis of pre-existing designs and are often confronted with a patent minefield. Patents of course can affect OSHW companies too, but the situation can at least be partially mitigated by the use of OSHW licenses which include a patent license for all downstream licensees, such as the CERN Open Hardware License [7] and the TAPR Open Hardware License [8]. These licenses are also persistent, in the sense that they require that publication of modified design files happens under the terms of the original license. This perpetuates the virtuous circle of sharing.

OSHW is different from FOSS in that hardware manufacturing needs to happen at some point, and that typically requires a bigger investment than that needed to publish software. There are many ways of making money that closely mimic those for FOSS, such as providing support or design services. But ultimately, most OSHW companies will try to manufacture and sell hardware as well. This means that there is often a higher risk because the necessary initial investment is higher. We will look at this issue in more detail in the next section.

One aspect of OSHW that makes it particularly compelling for small companies is the ability to easily bring in extra help in a given domain for which the company does not have a lot of expertise. The author was once confronted with a situation in which a company was late delivering a newly-designed VME-based ADC board. This company excelled in the ADC side but was struggling with the VME interface part, a domain in which the CERN group had lots of experience. It would have been in the interest of all parties to have the client help the provider in this particular part of the design. After a few months of delay, the company replied that they would actually have accepted to open up their design to the customer in order to receive help. But this was not possible because the VME core they were using was also a proprietary black box for them. So things were doubly locked up and there was no way for the CERN team to help the company fix the bug. In total, 8 months were spent on a problem for which a week should have been enough.

If FOSS can serve as a reference, OSHW should also make hiring and retaining talent easier. It is a recurrent problem in physics laboratories that new recruits take a long time to learn the very specific proprietary technologies used to solve a particular problem. When they finally do, they sometimes need to leave because of the end of a contract, and the person replacing them needs to go through a similar cycle. The use of more standard technologies would make it easier to integrate new people and would also make a more compelling case for candidates to apply. Their learning in the lab would be a worthwhile addition to the set of talents they would carry forward to their next job. Now, a standard solution does not *need* to be open source, but if we relate to experience in the software world, we see that FOSS implementations of standard technologies typically gain more traction than proprietary alternatives, for the usual reasons. In addition, these solutions often need to be slightly adapted to serve the particular needs of a laboratory, and that customization is much more natural in an open source scenario.

PERCEIVED ISSUES

Looking at the possible disadvantages of an OSHW paradigm for hardware development and procurement, it is useful to treat the case of users and providers separately. It is difficult to find any disadvantage for users, all other things being equal. However, quick user polls sometimes reveal some degree of reluctance to the concept, which can be broadly categorized as:

- Where does this stop? We are advocating OSHW PCBs, but what about the chips on those PCBs? Are we happy with those being proprietary? The easiness argument applies in this situation. It is not within the reach of many people to design an Application Specific Integrated Circuit (ASIC), so the loss for not making its design accessible is not that great, today. As ASIC design becomes democratized, we will see more open source designs like the lowRISC System-on-Chip (SoC) [9].

ISBN 978-3-95450-189-2

Then there is the notion of *commodity*. Some chips have direct pin-compatible replacements. If a chip is a commodity in that sense, the risk of lock-in is greatly reduced and the case for open-sourcing is not as strong. The same can be said of PCBs which are standardized to a large extent at the I/O level, such as PC motherboards.

- Will there be providers for a piece of hardware if I specify I want it to be OSHW? Experience reveals that this fear is largely unfounded. For better or worse, we live in a world of fairly active competition among many small, medium and large enterprises. This means that the chances of not finding a commercial partner, if a user has money to spend, are close to zero. As for many of the issues raised in this article, the explanation is not necessarily linked with OSHW: it's just plain old supply and demand.

Then there are a number of arguments which belong in the expectations management category. It is unreasonable to expect that publishing a design under an OSHW license will allow the designer to control all manufacturing and distribution of hardware based on that design. This is because of the way licenses work: they grant a right to the licensees that they would otherwise *not* have, in exchange for the promise of the licensee to accept a set of conditions from the licensor. The famous GNU Public License (GPL) works that way. Licensees do not have, in principle, the right to distribute a piece of code, or a modified version of it, for which they do not hold the copyright. The licensor grants this right in exchange for a promise by the licensee to distribute the original or modified source under the same licensing terms. There is no law that forbids a person to build hardware based on a published design, and to distribute that hardware later on, at least not in the general case. That means that our possibilities of controlling the distribution are necessarily reduced. OSHW licensing is mainly the licensing of the *design documents*. This is a powerful mechanism for ensuring freedom, but not as powerful as some would expect it to be.

It is also unreasonable to expect the same performance from systems which have received wildly varying amounts of resources for their development. This happens quite often in the software world. An incumbent proprietary technology on which the user has spent millions in license fees gets compared with a FOSS alternative on which the user has not invested at all.

There is no magic in FOSS as there is no magic in proprietary development. Features and quality are introduced by people who should be paid for their work. The real question is what would be the state of the FOSS (or OSHW) alternative had all those millions gone into its development, and whether it makes sense to switch at some point in time and start financing that option. This can represent a prohibitive initial investment in some cases, especially if big amounts of the work of the user have gone into adding features to the proprietary solution and "enriching" its ecosystem, thereby worsening the lock-in case.

The last big family of unreasonable expectations refers to frustration induced by the realization that bug fixes and improvements to a design don't start automatically flowing into the inboxes of the designers even if they "put it on the Web." The promise of OSHW not upheld! But, was the design of sufficient interest and usefulness to others? Was it properly documented? Was it easy to extend and contribute back? OSHW in itself does not help answer any of these questions.

To be complete on the issues chapter, we should also describe common concerns from companies which have evolved from a more traditional proprietary paradigm in which the search for big margins is linked with a certain amount of lock-in. The paradigmatic example which proves it is possible to survive and thrive while publishing the complete designs is Arduino. The potential user is offered a great variety of cheaper clones available in online commercial platforms. Many buy them. After all, these are functionally equivalent copies, compatible with the whole software ecosystem that surrounds Arduino. But many people also buy Arduinos designed, manufactured and sold by the original company. The same is true for products from Sparkfun, Adafruit and other vendors. Why is that?

Here we reach an important point of divergence between FOSS and OSHW. It is easy to download a copy of Debian, Ubuntu, Firefox or other successful FOSS products and obviate the donations button in the Web interface. On the other hand, it is impossible to receive an Arduino or one of its clones without paying any money. Pulling out one's wallet is a high-inertia move. That explains why people often choose not to donate to FOSS tools they download. But in OSHW the high inertia must be overcome in any case. So once the wallet is out, what should we do? Choosing the option which will support the OSHW paradigm which gave us this great product to begin with seems easy at that point, provided of course that the extra expense is reasonable. With that extra expense comes peace of mind for things as varied as PCB manufacturing quality, support, nurturing a community or the potential abuse of one's credit card number. The success of these OSHW companies and others proves that this mechanism works. Experience also shows that clones of successful hardware designs are a fact of life, irrespective of whether the design is OSHW. So it would look like business models based on the impossibility of cheap knock-offs are only viable for very high-end devices and companies with a large legal department.

From the perspective of a design group in a lab, it has also been suggested that there is a higher initial investment in OSHW projects, related to the fact that these projects are also a showcase for a given group and so special care is taken to make them high-quality. This typically means a good organization, documentation and quality assurance practices such as the design of one or more test fixtures, the development of test software and various design reviews. In fact, these are all good things to aim for and the fact that OSHW somehow pushes designers in that direction should not be an issue. But in certain situations the extra initial effort may not be justifiable, even in view of larger future benefits. OSHW is not for everyone all the time.

ISBN 978-3-95450-189-2

WHAT'S SPECIAL ABOUT PUBLIC INSTITUTIONS?

Public institutions are different from private companies in many aspects. One fundamental difference stems from the underlying interests they are supposed to serve and manifests in their rational economic behaviors. While private companies represent the interests of a selected set of individuals who own the company, a public institution represents the interests of the much larger set of people who form a society, most of whom finance these institutions by paying taxes.

It is expected of public institutions to purchase goods and services from private companies in the course of fulfilling their mission. Doing so in a way that will maximize their positive impact on the society they serve is an extremely important part of their work, and often far from trivial. What may look like a straightforward option from an engineering perspective in terms of cost and schedule might not look so compelling from the vantage point of a tax payer, and vice-versa.

For better or worse, economics is not a hard science, and the maximization of positive impact on society even less so. Should a public laboratory strike a private deal with one or a few companies for the exclusive transfer of a technology? Or should it open it completely, therefore enabling competition from companies both from member and non-member states of the institution? The optimal solution is not always clear-cut, and it is easy to fall pray of a kind of Stockholm syndrome and mistake the interests of individual companies with those of the whole society. Sometimes they are aligned and sometimes not. Keeping this in mind is crucial if designers and technology officers in public institutions are to fulfill their roles efficiently.

The fact that OSHW developed in a public institution is released for the whole world while development cost is only born by a subset of countries or regions can also complicate matters. The fear that companies in these regions are not sufficiently favored is often unfounded. It comes from an over-estimation of the actual value of the design files within a complete project. A close collaboration between the designers in the public center and those in a company can give the latter a considerable competitive advantage, even in a complete OSHW scenario. Many of the most important aspects of a project are not part of the set of released files. These include design intent, ideas that did not work, possible new applications of the technology and an assessment on how best to move forward, to name only a few.

The case of small and medium-size enterprises deserves special mention. Initial investment is often an issue for them. As we have seen, OSHW can lower the barrier for these companies to access a market. In a common scenario, one or more public "tractor" institutions make the necessary initial investment, thereby enabling many commercial actors to develop a technology further. It is important for public institutions to realize the benefits these actions bring about.

Another important aspect is procurement. Similar arguments apply. Should a public institution require that hardware it purchases have its design published as OSHW? If that incurred extra cost, would it be justified by the benefits it would generate for the laboratory, other similar centers and elsewhere in society?

As is often the case, FOSS leads the way. More and more countries are specifying that software purchased by public institutions using a public budget adheres to standard file formats, and sometimes will even specify that the software itself be FOSS. Specifying FOSS for purchases of software development work is even more commonplace. The whole chain of reasoning for these decisions maps entirely onto the hardware development case. It is therefore to be expected that public institutions will specify OSHW more and more for hardware they have developed by external contractors, as a means of maximizing the positive impact of their spending.

One important challenge remains in the domain of publicly-funded scientific research. A modicum of healthy competition among research groups is deemed an important ingredient to maximize the chances of scientific discovery and the development of useful technologies. There is however a risk that these technologies, which are often just useful infrastructure on the way to discoveries, do not get published by a given group out of fear that their competitors will out-perform them in the scientific part, thanks in part to their publicly-released infrastructure. This is effectively how a private company would (legitimately) behave, but with public money. From the perspective of a tax payer, the re-development of basic infrastructure in different research groups is inefficient. But the very survival of a group may in fact depend on such factors. It therefore would seem important to identify counter-productive incentives provided by the funding agencies where applicable and suggest changes in policy to avoid this waste of resources.

RECIPES FOR SUCCESS AND FAILURE

Most of what makes OSHW projects succeed or fail is basically common sense that one could apply to proprietary projects as well: knowing what the problem is one is supposed to be solving, preparing a plan, informing relevant stakeholders about it, gathering a good team of developers, following up progress, documenting, etc. There are however some factors which play a particularly important role in the case of OSHW projects.

When a design team goes open, there is an expectation that other designers will find the project interesting and contribute in different ways. It is important to bear in mind that these developers are offering the project their most valuable asset: their time. In exchange, they will expect to be treated as first-class citizens within the project, and in particular they will expect that their opinions are taken seriously. A team releasing an OSHW project will therefore help the project be successful by keeping an open mind and allowing ideas coming from elsewhere to make their way into the product. This means that there are chances that the product will in the end not be exactly what the original team had imagined, and that should be for the better. Keeping this flexibility is easier

ISBN 978-3-95450-189-2

said than done. It is a fact of life that competent developers often have quite a clear idea of how things should be done. But how to react to a situation where two competent developers have different clear ideas? Losing talent unnecessarily is a tragedy for an OSHW project. Flexibility is key in order to find a good balance.

But how will competent developers be attracted to a project to begin with? Often they will have a problem to solve themselves, and if an OSHW project comes close enough and shows flexibility, the case will be compelling enough for them to join. This means that an OSHW project will have more chances of succeeding if the original designers anticipate the fact that it will be used elsewhere. The use of standards and consequent enlargement of the initial scope is a useful way to render a project more neutral as far as applications are concerned. This is for example what the White Rabbit [10] project did. Its original scope was to replace the aging General Machine Timing system in the accelerators at CERN. But basing its infrastructure on standards like Ethernet and the Precise Time Protocol, it has evolved into a general solution to deal with all sorts of problems related to hard real-time distributed controls and data acquisition. A large community of users [11] and developers have found ways of using and improving this technology in many different contexts, inside and outside CERN. The overall savings are phenomenal, and developers enjoy collaborating with a network of very competent and helpful peers.

Another common pattern in successful OSHW (and FOSS) projects is walking a part of the path together with partners whose ultimate interests might not be fully aligned with yours. The perfect example is the Linux kernel. Big companies such as Intel, Google and IBM contribute great amounts of code to the kernel, because they consider it basic infrastructure on which they do not compete. This infrastructure enhances the value of proprietary offerings they support on top of it, on which they compete. Taking this point of view to the extreme, even commercial companies which sell Linux distributions could be seen as adding a proprietary layer (e.g. their internal processes) on top of a GNU/Linux system. Although their ultimate goals and sources of income are not necessarily aligned with those of other kernel developers, their contribution is invaluable to the success of the project. Similarly, in OSHW, a piece of hardware could be an end product for some or a piece of basic infrastructure for others. And the same applies to FOSS tools for hardware design, about which more in the following section.

THE FUTURE: A CALL FOR ACTION

OSHW promises better hardware, more accessible to bigger numbers, an educational experience for developers and a fun way to collaborate and get work done. It also revisits the traditional roles of provider and user, allowing for an array of intermediate profiles and opening up new ways of collaborating. So what will it take for it to be more successful and pervasive, following the lead of FOSS? In the opinion of this author, two main things: tools and organization.

It is not surprising that the first developments the FOSS community tackled in the 90’s were for the basic infrastructure one needs to develop more software: editors, compilers, the C library… FOSS is about sharing, and sharing is severely handicapped when your colleagues cannot open your source code, modify it, compile, test and send it back. In a landscape of incompatible proprietary compilers, the choice of one of them as a standard is arbitrary and sharing of code is highly inefficient. Fortunately for the FOSS people, the skill they needed to solve the problem was what they were best at: software development. And so they proceeded to write these tools and FOSS promptly took off with the great results we know today.

The situation for OSHW is more complicated. The dominant PCB design tools are proprietary. Even the proprietary market is highly fragmented, so the chances are relatively low that the recipient of design files holds a valid license for the tool with which the design was made. FOSS tools exist (KiCad, GEDA and others) but they are still not up to the level of the best proprietary tools in terms of features and quality. Things are complicated by the fact that the skills needed to improve the situation (software development) do not match the skills of most OSHW practitioners (hardware design). These hybrid profiles are difficult to come across, so the evolution to a hardware design scene dominated by FOSS tools is harder than in the case of FOSS development. The role of the big OSHW companies as trend setters will be important in this respect. Olimex [12], with their recent migration to KiCad, shows the way.

Halfway between hardware and software development we find the realm of Field Programmable Gate Array (FPGA) design using Hardware Description Languages (HDL). There is great potential for development in this domain too. A collection of high-quality HDL cores and a set of FOSS simulation and synthesis tools would bring great benefits to society. There, as well, an important amount of work remains to be done. There are very complete VHDL and Verilog simulators (GHDL, Icarus Verilog) but no high-quality mixed-language FOSS simulator. Synthesis efforts have not benefited from the support of the FPGA vendors so far, but the Yosys project [13] has shown the way and more exciting developments are to be expected in the coming years, hopefully with the involvement of FPGA vendors. Effort is also needed on developing appropriate strong and weak copyleft licensing regimes for HDL design. This will allow this domain to benefit fully from the multiplicative effect of copyleft.

How can all this be achieved? The role of the Free Software Foundation and other organizations in the rise of FOSS is well known and acknowledged. We have seen the positive effect of organizing and pooling resources in many cases. Take the example of the SCOAP3 initiative at CERN. This Open Access action has effectively turned the market of scientific publications in the field of High Energy Physics upside down. Institutes hosting researchers put together all

ISBN 978-3-95450-189-2

their articles and ask publishers to quote prices for their editing and publishing. In this way, readers have free access to all papers without paying. This maximizes the dissemination of these research results. The overall cost for tax payers is reduced by pooling all articles together and having publishers compete on editing and publication costs. This revolutionary change in the paradigm for scientific publications was only made possible by institutes getting together and by creating a layer of organization and coordination.

The same should work for OSHW. Organizations like OSHWA and FOSSi [14] have already started taking these roles and working on important aspects such as OSHW certification and a repository for HDL cores [15]. GOSH [16] focuses on the particular case of OSHW for science. Other problems like HDL licensing and PCB development tools could also benefit from the help of these or similar organizations.

OSHW – as FOSS – suffers from difficulties in quantitatively evaluating its impact. The complete freedom given to users automatically implies that tracking their use of a technology can only be done indirectly. Modern research on economics acknowledges the fact that many actions are ultimately beneficial even if the quantitatively formal case is difficult to make for them, and warns against decisions based solely on measurable outcomes. We have already seen how OSHW can make attracting and retaining talent easier for organizations. This is one example of an outcome whose positive impact is difficult to quantify. There are however efforts underway to provide funding agencies with more compelling cases regarding FOSS and OSHW, and a certain degree of organization would also help in this regard.

Ultimately, once all the infrastructure is in place, the flow of hardware designs will be unrestricted. For the reasons explained above, it is quite clear that designers working in public institutions can play a key role in the adoption of OSHW and the production and distribution of vast amounts of OSHW designs. How this will be organized is up to us.

CONCLUSION

The border between hardware and software is becoming ever more blurred. Design entry for hardware happens through CAD programs or text editors in the case of HDL. These "sources" can then take different paths. If they are fed to a simulator along with testbench code, they effectively behave as software. If they are fed to a production line, the end result is a tangible product. The legal world still uses different formalisms for hardware designs and software source code, but the conceptual similarities are undeniable.

Since the advent of FOSS, software developers have benefited from a privileged environment. They can share their work and bring new talent into a project easily. They can learn from each other while working in local or distributed teams, much more easily than they could in the past. The result is better software, more efficiency and more freedom for the users. OSHW brings that to the world of hardware development. However, it needs to overcome the inertia of a legal, social and economic paradigm based on proprietary design.

In this article we have looked at the pros and cons of transitioning to OSHW, with a special emphasis on design and procurement work carried out in public research laboratories. The aim has been to be more fair than neutral, and to trigger further discussion in laboratories and elsewhere.

ACKNOWLEDGMENTS

The preceding analysis and opinions are necessarily the result of countless discussions with many colleagues and friends, and also considerable amounts of reading, watching and thinking. There is also a fair dose of practical experience at work. Trying to thank everybody here would be futile, but a few poor souls deserve special mention because they read this long article and helped me make it better: David Cobas, Mick Draper, Pietari Kauttu, Rafael Pezzi, Eduardo Ros, Daniel Tavares and Federico Vaga. Thank you all.

REFERENCES

[1] Open Source Hardware Definition, http://www.oshwa.org/definition/

[2] Open Source Definition, https://opensource.org/osd

[3] The four freedoms of Free Software, https://www.gnu.org/philosophy/free-sw.en.html

[4] The F-Watch, http://www.ohwr.org/projects/f-watch/wiki

[5] Open Data at CERN, http://opendata.cern.ch

[6] Open Access at CERN, http://library.web.cern.ch/OpenAccess

[7] CERN Open Hardware License, http://www.ohwr.org/projects/cernohl/wiki

[8] TAPR Open Hardware License, https://www.tapr.org/ohl.html

[9] lowRISC, a fully open-source, Linux-capable RISC-V-based SoC, http://www.lowrisc.org/

[10] The White Rabbit project, http://www.ohwr.org/projects/white-rabbit/wiki/

[11] White Rabbit users that the WR development team knows of, http://www.ohwr.org/projects/white-rabbit/wiki/WRUsers

[12] Olimex and KiCad, https://olimex.wordpress.com/tag/kicad/

[13] Yosys Open SYnthesis Suite, http://www.clifford.at/yosys/

[14] The Free and Open Source Silicon Foundation, http://fossi-foundation.org/

[15] LibreCores, https://www.librecores.org/

[16] Gathering for Open Science Hardware, http://openhardware.science/

ISBN 978-3-95450-189-2