

A Survey of Open Processor Core Licensing

Andrew Katz,^a

(a) Partner, Moorcrofts LLP

DOI: [10.5033/fjossr.v10i1.130](https://doi.org/10.5033/fjossr.v10i1.130)

Abstract

In the Spring of 2018, Western Digital Corporation commissioned Andrew Katz of Moorcrofts LLP to prepare a survey of open processor core licensing. This is an edited version of that report.

Keywords

Law; information technology; Free and Open Source Software; open hardware licensing; processor cores

Foreword by Alan Tse, Western Digital Corporation:

Western Digital's relationship with open source has evolved significantly over the last decade. When I first joined Western Digital, our main focus was on open source compliance. That is because in 2009 we were one of the first major companies sued for our open source use. As a result, the main goal for the next few years was to prevent any litigation happening again and we viewed open source with some trepidation. Over time our view shifted as we had to learn about the importance of the open source community and how to be a good participant in that community. And over the years, we have increased our participation – not to avoid litigation, but because our own business interests have started to align. Over the years we have made multiple contributions to the Linux kernel and other open source projects and we have released internal tools that we thought others could use. Now that it has been almost a decade since that first lawsuit, we would like to think that we have learnt a bit more about the open source community and we are proud to say we are a part of that community.

While we have been public about our support of RISC-V and plans for RISC-V cores since 2017, we also believe the best way to show our commitment to the open source community is by leading from the front. Following our announcement at the December 2018 RISC-V Summit, we recently released our RISC-V SweRV Core under an Apache-2.0 licence on January 24, 2019.¹

In deciding our licensing strategy for our core release, we engaged Andrew Katz to help us understand the community norms for open source hardware. Owing to his involvement in the drafting of two open source hardware licences, we believed he was at the forefront of Legal scholarship on this issue. His report that follows was instrumental as we balanced our goals of community growth and protection in a space with unique constraints. It's unique because open source

¹ For more information see <<https://github.com/westerndigitalcorporation/swerv>>

hardware is a capital-intensive field quite different from software and the fact that the established open source licences were written without open source hardware in mind. We are happy to release this research to the community and hope this research and our journey serves as a beacon to our peers to join use in the open source hardware community.

– Alan Tse, Associate General Counsel, Western Digital.²

Introduction

The research was undertaken by Andrew Katz between March 26 and April 22 2018.³ The methodology was as follows:

1. To identify major open hardware communities using a combination of research and pre-existing knowledge of various open hardware activities that Andrew Katz has been involved in including both specific projects and umbrella organisations.
 2. To undertake research of those organisations and schedule and carry out a range of telephone interviews with identified leading individuals in the field. Given the relatively short time available to undertake the research, a total of eight individuals were identified, of whom six were able to agree to an interview within the time available for the first version of this report. A further two individuals arranged to be interviewed on a date after the original date of submission of the report to Western Digital, and their responses have been taken in to account in the updated version. No one who was approached declined to take part in the research, and all were very open and candid. We are grateful for their time and interest in the project. We also requested further input from the interviewees about community development and involvement, based on the answers to the first round of questions, and two individuals responded comprehensively by email. Their responses were taken into account in the report.
 3. To review the projects listed on LibreCores and OpenCores.org, and the list researched by Mohammad Shahradi⁴ and updated as a result of further desktop research and responses from interviewees.
 4. The results of the research were compiled into this report.
 5. In order to facilitate candour on the part of the interviewees, the interviewees were told that their names would not be linked to specific comments they made in a manner similar to the Chatham House Rule. Subsequently, the individuals kindly consented to their names being released.
 6. To avoid bias in answers provided, the interviewees were told the research was being undertaken on behalf of a major US digital hardware manufacturer, but no further
- 2 Alan Tse is a member of Western Digital's Legal team and responsible for open source compliance across the company and supporting Western Digital's open source strategy. His practice covers product lines both up and down the stack including storage devices firmware, consumer devices, data centre systems, and now even hardware cores. As a former computer engineer who grew up using open source software and anxiously waiting for the year of the Linux desktop, he has watched the evolution of open source throughout the tech industry and occasionally dabbles in various open source communities.
- 3 The data presented in this paper represents information obtained during that research period, unless explicitly stated otherwise. For example, during discussions with Mohammad Shahradi (see footnotes 4, 19 and 20) we agreed to provide him with an updated of the data he presented in the paper referenced at footnote 4, and since this update was provided as at 29 January 2019, we decided to update the relevant text and appendix of this report accordingly. It does not affect the conclusions. The author thanks Heather Stewart for her invaluable assistance in the updating process.
- 4 Balkind, Joseph, et al. (2016) 'OpenPiton: An Open Source Manycore Research Framework'. ASPLOS '16, pp 217 – 232. <<http://parallel.princeton.edu/papers/openpiton-aspl0s16.pdf>> DOI: 10.1145/2872362.2872414

information was provided about the research sponsor. The identity of the sponsor was released to the interviewees some months later when they were asked if they were prepared to waive anonymity.

Open Source Hardware and Licensing

Summary

Broad consensus is that ‘Open Source Hardware’ is hardware whose licensing terms comply with the definition set out by the Open Source Hardware Association. Although the thrust of the definition is relevant to this report, the detail is not.⁵ The OSHWA definition follows the Open Source Initiative’s definition for Open Source software licensing.⁶

Specific licences which have been identified⁷ by OSHWA are:

Copyleft (reciprocal) licences:

- Creative Commons Attribution, Share-Alike (CC-BY-SA)
- GNU General Public License (GPL)
- Hardware-Specific Licenses: TAPR OHL, CERN OHL⁸

Permissive Licences

- Free BSD license (BSD-2-Clause)⁹
- MIT license (MIT)
- Creative Commons Attribution (CC-BY-3.0)¹⁰
- Hardware-Specific License: Solderpad Hardware Licence¹¹

Given that the above licences are specifically referenced by OSHWA we can make the reasonable assumption that they meet the OSHWA definition. OSHWA does not (at the time of writing) have a process for approving licences. It can be assumed that licences (such as Apache) which are approved by the OSI would also meet the OSHWA criteria.

Licences that were identified during the course of this survey as applying to various open source hardware projects are:

5 For more information see <<https://www.oshwa.org/definition/>>

6 With the interesting distinction that in the preamble, OSHWA states that the design must be publicly available so that anyone can make etc. the design. OSI only requires that the licensing terms enable the licensee to make open source software publicly available, but not that public availability itself is necessary.

7 <<https://www.oshwa.org/sharing-best-practices/>>

8 Andrew Katz has been involved in the drafting of CERN OHL <<https://www.ohwr.org/documents/294>>.

9 <<https://opensource.org/licenses/BSD-2-Clause>>

10 <<https://creativecommons.org/licenses/by/3.0/>>

11 Andrew Katz drafted the Solderpad Licence. <<http://solderpad.org/licenses/>>

Licence	Comments
BSD-2-Clause (simple permissive)	Widely used for many types of open source hardware, including processor cores
MIT (simple permissive)	Widely used for open source hardware
ISC ¹² (simple permissive)	Sometimes used for open source hardware
Apache-2.0 (permissive with patent clauses)	Widely used for open source hardware
GPLv3 (strong copyleft with patent licence)	Frequently used for open source hardware
GPLv2 (strong copyleft without patent licence)	Frequently used for open source hardware
LGPL (various versions)	Frequently used for open source hardware
MPL-2.0 (weak copyleft with patent grant)	Rarely used for open source hardware

Table 1: Open Source Software Licences

Licence	Comments
Creative Commons Attribution (various versions)	Widely used for open hardware designs
Creative Commons Share-Alike (various versions)	Widely used for open hardware designs
Creative Commons Public Domain Dedication (CC0)	Widely used for open hardware designs

Table 2: Open Content Licences

Licence	Comments
TAPR (Tucson Amateur Packet Radio) Open Hardware License	Mainly used for RF circuit boards. Has interesting copyleft mechanism, based on patents
CERN OHL (various versions)	Used for a wide variety of open hardware but originally designed mainly for applicability to circuit boards
Solderpad Licence (Versions 0.51 and 2) (an Apache-based Open Hardware License)	Used for a wide variety of hardware, including cores
Open Hardware Description Licence (Mozilla Public License-based open software licence)	Designed specifically for semiconductor cores. Rarely used.
NVIDIA Open NVDLA License and Agreement v1 (an Apache-based Open Hardware License)	Designed specifically for NVDLA (Nvidia Deep Learning Accelerator)

Table 3: Hardware Specific Licences

¹² <<https://opensource.org/licenses/ISC>>

Licence	Comments
Public Domain Dedication	Public domain dedication is not recognised in many jurisdictions, although it may take effect as a broad licence. CC-0 (see above) seeks to remedy this by providing an explicit fallback licence
Creative Commons NC variants	Non-commercial licences contain a restriction against a field of endeavour (commerce) contrary to paragraph 8 of the OSHWA criteria
Open Compute Project Licences (passive and copyleft)	Designed for hardware for use in OCP-compatible databases. The licences only really work when the various participants are patent holders, and are better regarded as standards-coupled licences

Table 4: Licences which are not compliant with OSHWA/ODI criteria

Note that both the Solderpad licence and the CERN OHL are in the process of revision. Version 2 of the Solderpad licence remains very similar to the Apache licence it is based on, but has been amended so that is now expressed to be a ‘wraparound’ of the Apache licence, rather than expressed as a different license. The advantages are that it is much easier for a practitioner familiar with Apache 2.0 to immediately see what the differences are between Solderpad and Apache 2.0.

As of January 2019, the CERN OHL is in the process of being modified significantly to produce version 2. Under current proposals this will be published in three variants: a permissive version which has an Apache-like effect, and two reciprocal versions – lesser and strong (strong reciprocal being the default for those who have already published hardware designs under current versions of the CERN-OHL with the ability to select a later version). Care has been taken to consult with developers of FPGAs and ASICs to try to meet their concerns, particularly around the use of proprietary tools and libraries that are all but unavoidable in practice, while retaining the copyleft nature of the reciprocal versions of the licence.

Desktop Analysis of Licence Adoption

The OSHWA Surveys

Across open hardware as a whole probably the most in-depth survey of open source hardware use and attitudes was undertaken by the OSHWA in 2012¹³ and 2013.¹⁴ This contained a small section on licence adoption. It should be noted that this survey covered open hardware in general, from mechanical items through to electronics, but there is no indication that any of those responding were involved in development at sub-component (i.e. chip design) level. Therefore, the results are both fairly out of date and of dubious relevance to chip design. One section of the survey related to licence adoption, and like the annual Black Duck licence adoption survey¹⁵ counts all projects of equal weight. For example, in the Black Duck survey the Linux Kernel counts as a project with equal weighting to a tiny driver project which appears on GitHub but has never been used in commercial deployment). The results are therefore a dubious reflection of reality though it is interesting to note that very nearly 50% of the respondents had released projects with no explicit licence. It is difficult to interpret the results, as each respondent was permitted to respond with multiple answers to the

13 <<https://www.oshwa.org/oshw-community-survey-2012/>>

14 <<https://www.oshwa.org/oshw-community-survey-2013/>>

15 <<https://www.blackducksoftware.com/top-open-source-licenses/>>

question of which licence they had used, but the thrust for open hardware in general (covering everything from mechanical devices and casings through to circuit boards) is that there is rough equality of deployment of copyleft licences (e.g. Creative Commons Share-Alike, GPL) and permissive (e.g. MIT, BSD, Creative Commons attribution-only) licences.

GitHub Search

Many open hardware projects are hosted on GitHub. CERN carried out some basic research on how many projects have adopted the CERN OHL by carrying out a Google search for “site:github.com CERN-OHL” that as of March 2018 produced¹⁶ 657 results. It is misleading to assume these are all projects. However, undertaking a random sample of 10 pages from the complete Google results shows that around 80% of the results **are** projects. It is not easy to tell if these are unique results, but if they are, it suggests that something over 500 CERN-OHL licensed projects exist on GitHub. By comparison, TAPR OHL only generates 39 results of which 15 appear to be projects.¹⁷ Solderpad shows 434,¹⁸ almost all of which appear to be legitimate projects. It should be noted that it is more difficult to use this sort of search to find hardware projects licensed under Apache, MIT or BSD for the simple reason that the search will generate, overwhelmingly, software projects.

*The OpenPiton Survey*¹⁹

As part of a 2016 paper, Mohammed Shahrads, a member of the Princeton OpenPiton team, researched active processor core projects.²⁰ We have updated, corrected and verified the information presented and a summary in the table in appendix 2 under the section ‘OpenPiton’.²¹

Of particular interest is that, when the projects are listed in order of the date of last active contribution, it is clear that the more recent projects are more heavily weighted towards permissive, rather than copyleft licensing. There is a total of 28 processor core products listed. There is a gap between October 2015 and February 2017, and if we take the projects that have been active since February 2017 (of which there are 15), 5 of them are copyleft. For projects prior to this date (of which there are 13), 12 are copyleft.

To summarise: recently active projects are split 33% copyleft, 67% permissive, as against the non-active projects, which are 92% copyleft, 8% permissive. This indicates a clear shift to permissive licensing for currently active projects.

16 As of 29 January 2019, this has increased to ‘about 1500’, but the search results are somewhat noisier, so it’s not clear if this is a valid comparison.

17 We tried to rerun the search on 29 January, but the results were so much noisier that it’s impossible to make a valid comparison.

18 We tried to rerun the search on 29 January, but the results were so much noisier that it’s impossible to make a valid comparison.

19 Balkind, Joseph, et al. (2016) ‘OpenPiton: An Open Source Manycore Research Framework’, ASPLOS ‘16, pp 217 – 232. <<http://parallel.princeton.edu/papers/openpiton-asplos16.pdf>> DOI: 10.1145/2872362.2872414

20 <http://parallel.princeton.edu/openpiton/open_source_processors.php>

21 The results in the appendix have been updated to 29 January 2019, and therefore differ slightly from the version of the table provided to WD in the original version of the report. The figures above have been updated accordingly. For comparison, the text in the original report read: “There is a gap between October 2015 and February 2017, and if we take the projects that have been active since February 2017 (of which there are 12), 5 of them are copyleft. For projects prior to this date (of which there are 14), 12 are copyleft.”

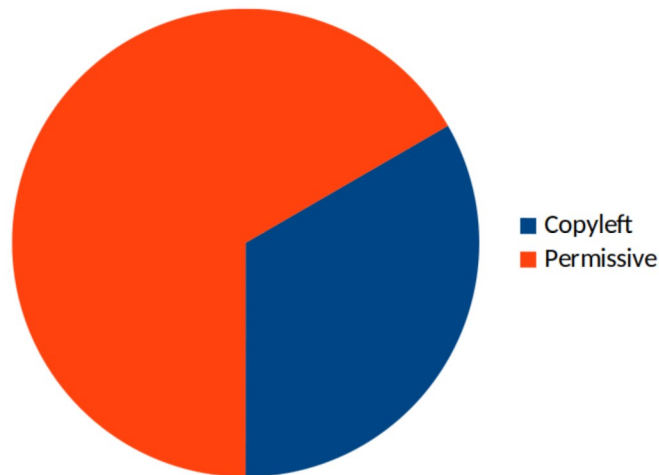


Fig. 1: Summary of licences chosen for recently active projects, data from the OpenPiton Survey

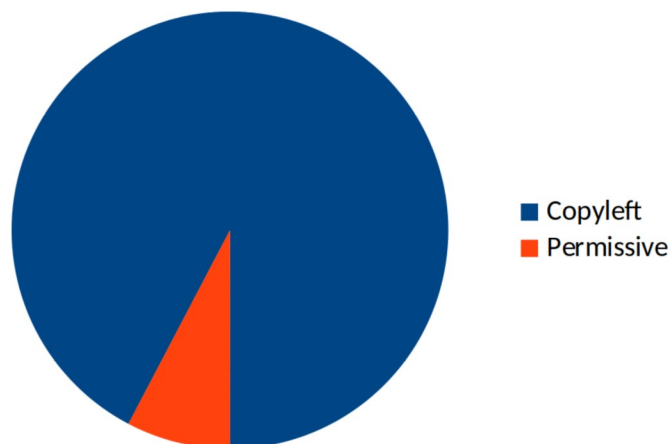


Fig. 2: Summary of licences chosen for non-active projects, data from the OpenPiton Survey

OpenCores and LibreCores

Two websites, opencores.org and librecores.org, host core designs and related materials such as tools and interfaces ('interfaces' are materials for other components which would typically appear on silicon alongside a core, such as UARTs and memory controllers). Opencores is run by a commercial entity, a situation which led to dissatisfaction from members of the FOSSi foundation regarding how Opencores operated, and their subsequent creation of Librecores as an alternative. Librecores has fewer projects, but they tend to be more active than Opencores (possibly because they have had less time to become obsolete).

There is a total of 1190 entries on the Opencores website, including software, toolchains, utilities and interfaces, as well as cores, of which 30 are marked verified. The Librecores site contains 90 entries but does not have any form of verification mechanism. We examined 24 entries in the Opencores website which are marked as 'verified' and 40 entries on Librecores. We selected entries which most

clearly relate directly to cores and interfaces, details of which are contained in appendices 3 and 4. There is also a thriving ecosystem of associated software tools, test suites and build and utility scripts, analysis of which is outside the scope of this report. Whilst we have undertaken a statistical analysis of this data, it is important to note that should be interpreted in the light of the following constraints:

- (1) there is no easy way to weight each entry in terms of how pervasive and active the project is, so a barely-functional and rarely-adopted project would rank the same as a more mature and active one;
- (2) there are significant projects which are not represented on either database;
- (3) the selection of entries is largely subjective, and whilst the intention is to select projects which instantiate hardware (as opposed to toolchain or utility components), the selection was undertaken by a lawyer and not a microelectronics engineer, so mistakes are inevitable.

Various analyses of the licensing in both Opencores and Librecores for various categories of project are set out on Appendices 3 and 4.

Outcomes of the telephone interviews

Licensing – copyleft vs. permissive

All but one interviewee noted that a permissive model was the most likely to succeed from a commercial perspective. All acknowledged that a particular issue with copyleft licensing was that existing licences, including GPL and LGPL, and even CERN OHL did not provide sufficient certainty as regards boundaries delineating where the copyleft effect occurs. For example, if a component whose design is released under LGPL is combined with another component on the same silicon, does that mean that both components then have to be released under the LGPL? How about if the components are on separate chips? One interviewee specifically referred to the little-understood requirement in LGPL for sufficient interface information to be made available (together with the right to reverse engineer), for the LGPL component to be modified and re-linked to the ‘work’ as a whole. It is not clear how that would work with electronics especially since the works could be combined on static silicon (as masks). One interviewee noted that OpenSPARC (which was licensed under GPLv2) had in the past proved to a successful design (used for devices as diverse as digital cameras and network interfaces), thus demonstrating that GPL-based designs are capable of being commercially successful. There is little publicly available information on OpenSPARC (which is a relatively old project, having been released in 2006), and the interviewee suggested that separate research should be undertaken by locating some of the individuals who had been involved in the project initially, and in particular, the decision to open the technology, and to interview them.

Horizontal and Vertical Boundaries

Another interviewee made the explicit distinction between ‘horizontal’ boundary problems (as mentioned above), and ‘vertical’ boundary problems where it is not clear whether a requirement to release design documentation for a circuit design (or similar) also requires releasing the designs of the components themselves. It was noted that the CERN OHL explicitly deals with this via the requirement to release information for modifications at a similar ‘level of abstraction’ to the original design.²² The current version of CERN OHL does not deal with the horizontal boundary problem

²² At of January 2019, proposals for CERN OHL v2 take a different approach and have introduced the concept of an ‘Available Component’. Designs do not have to provide the design documentation for components that qualify as ‘Available Components’, which include items like readily available electronic components, provided that enough information about their specification, characteristics and interfaces is available to enable them to be sourced or used in the design. Thus a 555 timer when provided along with its datasheet would qualify as an ‘Available Component’.

(although this is to be addressed in the upcoming version 2 as mentioned above). The Open Hardware Description Licence (based on Mozilla Public License 2.0) does address this problem but is not frequently used.

One interviewee suggested the horizontal boundary problem might be fixed by saying that a weak copyleft licence could be drafted in a manner that the licensor provided an interface definition alongside the code. Provided that any third party complied with the interface definition, their code linking to the original licensor's code would be free of the reciprocal effect. The next version of the CERN OHL – see above – is likely to adopt an optional mechanism similar to this.

What drives licence choice (copyleft vs. permissive)?

Most interviewees expressed a preference for permissive licensing on the basis that existing copyleft licences left too much uncertainty, and that this uncertainty would inhibit adoption. It would also make it more difficult to deal with companies which provide proprietary libraries as those companies would be uncomfortable having their proprietary library used in a design which covered by copyleft of uncertain scope. One interviewee noted the value in copyleft licensing and noted that the Open Hardware Description licence expressly addressed the scope problems, but that it had not been widely adopted.

When it was suggested in each case that the next version of the CERN OHL would likely incorporate additional optional exceptions which expressly limited the reciprocal effect (as noted above) respondents suggested that this would cause them to potentially reconsider their licensing choices and consider its adoption. However, that there was little point in examining the issue in greater depth until such a licence was more widely accepted in the wild.

It was generally accepted that licence choice was ideological, and that some projects would be more inclined to wish to maximise use of their designs by providing them under a permissive licence while accepting the danger that the designs may become incorporated into proprietary hardware, while others wished to maximise freedom by making them available under a copyleft licence which ensured modifications would be made available under the same licence. However, all parties were all uncomfortable with existing copyleft licences, and regarded the issue, at this stage, as being largely hypothetical.

One interviewee noted that the use of components under copyleft licences in their current state would potentially cause difficulties with fundraising. One interviewee noted that in a due diligence exercise it was not unusual to run a code-scanning tool such as Black Duck against HDL files, although it is not immediately clear what the benefits of such an activity would be and whether Black Duck holds any HDL in its codebase, other than potentially to scan the code for licence texts such as the GPL which are frequently regarded as 'risky' by funders.

'Selling exceptions to the GPL'

One interviewee did note that it was possible that a design could be licensed under a restrictive copyleft licence of uncertain scope with respect to hardware such as the GPLv2 with a view to the licensor making a parallel proprietary licence providing certainty available for a fee. Clearly, this model tends to cause the licensor to use more restrictive licences in an effort to drive adoptees to the proprietary licence, whilst still permitting the licensor to describe the designs as 'Open Source [hardware]'. Richard Stallman, founder of the Free Software Foundation, has described this practice disparagingly as 'selling exceptions to the GPL'.²³

²³ Stallman, Richard 'Selling Exceptions to the GNU GPL' <<https://www.gnu.org/philosophy/selling-exceptions.en.html>>

One interviewee provided, as an example, the Leon core provided by Gaisler and based on SuperSPARC that is available both under LGPL/GPL and a proprietary licence. This was simply an illustration of dual licensing and does not suggest any particular motivation on the part of Gaisler for choosing that licensing model.

Open Hardware Communities

The consensus among interviewees was that the lack of open source or low-cost toolchains was an inhibiting factor in the growth of open hardware communities focusing on cores.

It is noteworthy that cores which emulate obsolete or obsolescent designs, primarily of interest to hobbyists, are more likely to be licensed under copyleft licences. For example, the Neo430, OpenMSP430 and T400 and T48 μ Controller cores, examples of cores from selected OpenCores projects which fall into this category, are all licensed under copyleft licences.

After the initial phase of interviews, a second set of questions were sent to the interviewees focusing specifically on community building. We received two comprehensive responses within the time available, and both noted that permissive licences would be more attractive to commercial projects owing to avoidance of the problems around perceived linking. Both also pointed out that there probably was not enough data available to determine whether projects using non-open-hardware licences would have chosen an open hardware-specific licence if one was available. A potential illustration of this is that the OpenPiton list only three projects out of 26 chose a hardware specific licence (in all cases, the Solderpad licence.²⁴ In no case was a hardware-specific copyleft licence chosen.

Both responses also indicated that, most commonly, projects based on a permissive licence retained the same licence when out- bound licensing (i.e. the licence under which the design is to be licensed to third parties), as for the in-bound contributions.

In terms of community development, interviewees stressed the importance of evangelism and outreach, and funding community development. One individual also stressed the importance of becoming involved in projects like the FOSSi foundation.

Toolchains

One issue that came up frequently, although detailed discussion is outside the scope of this report, was that open source toolchains are much scarcer in the world of open hardware than they are in software. The extent to which the toolchain will incorporate code of its own into the output, and what the effect of that code is from a legal point of view, is highly problematic: it is a debatable point as regards software but becomes even more so when applied to hardware. Questions arise such as whether a bitstream is in any sense a computer program, and - if so - who 'runs' it when the hardware starts up.

Patents

The interviewees generally noted that patents were a potential problem but had no clear suggestions on how to address this challenge. It was noted that members of the RISC-V foundation get the benefit of a cross-licence agreement from the other members, but that non-members, although they are able to use the ISA specification freely, gain no form of explicit patent licence or protection.

²⁴ The results in the appendix have been updated to 29 January 2019, and are not the version of the table provided to WD. The text of the version of the report released to WD accordingly read "two projects out of 26" in the sentence to which this is a footnote.

One interviewee noted that there was a move towards licences such as Apache 2.0 away from BSD or MIT because of its explicit patent provisions. One noted that the Solderpad licence (an Apache variant) had been adopted by LowRISC and PULPino because it was a relatively simple licence which had been modified specifically for hardware and had Apache-like patent provisions.

Establishing a default licence to use - recommendations

Broadly, licence choice should be limited to one of the more popular licences. Which specific licence is chosen depends depending on business needs for that the relevant project. The most popular software licence choices include the licences of the GPL family, Apache 2.0 and potentially MPL. For hardware, these may roughly correlate with CERN OHL/TAPR, Solderpad or BSD/MIT and Open Hardware Description License. Less well-used licences should be avoided, because they may cause licence incompatibility problems, and it makes project adoption more problematic. It is worth bearing in mind that the lawyers acting for counterparties prefer to work with the text of better-known licences to avoid having to spend expensive time to become familiarised with them. The informal drive towards licence standardisation is a topic which arises at legal licensing conferences quite frequently: the observation goes that the GPL, for all its flaws, is well understood, so tends to lead to a better legal outcome for both parties in contrast to a licence like (for example) the Open Software Licence, which is arguably better drafted, but less well used and understood.

It may be the case that there is, in practice, no choice that can be made, if the project uses, for example, a GPL component at its core which cannot be sufficiently decoupled from the rest of the work. In that case, the whole project would likely have to be released under the relevant version of the GPL.

For projects where there is no such constraint the specific choice of licence will depend upon the criteria of the specific project. The key question is whether the licensor is seeking to maximise either *utilisation* or *freedom*.²⁵

If the licensor is seeking to maximise utilisation then a permissive licence such as Solderpad²⁶ will be most appropriate. In this case, the licensor must be comfortable that the software or hardware design may be incorporated into proprietary systems, and that the source code/design of any modifications may not be made available.

On the other hand, to maximise freedom, a good choice is the CERN OHL (adopting, where appropriate, one of the reciprocal versions, when v2 is released).

Another option as referred to previously in this paper is to sell proprietary unrestricted licences alongside a given open source licence (assuming the licences of the other components allow this). It is common practice to use a restricted licence (such as GPL or CERN OHL with no exceptions) to enhance the attractiveness of the proprietary option, though while this is common it is frowned upon by the GPL community. On the other hand, a legitimate reason for dual licensing may be that the licensee wishes to use a GPL-licensed core alongside third-party proprietary components, and therefore has to seek a licence from the licensor of the core which is compatible with those components.

²⁵ In the sense of 'liberty'. In other words, the designer's intention is that the design, in all its incarnations, remains free of constraints on reuse, modification and distribution, and also has the effect of causing other designs combined with it to be equally free, with the overall intention of increasing the commons of free designs.

²⁶ Or the newly (January 2019) announced permissive version of the CERN OHL.

Conclusion

All interviewees believed that the most commercially effective open hardware core designs were those which adopted permissive licences. The prevalence of these licences is borne out by desktop research. The stated various reasons for this are:

- that the currently available copyleft open hardware licences are insufficiently clear in their effect to be safely used;
- that the potential benefits of copyleft licensing in core designs are not yet sufficiently clear to show an overwhelming need to shift to a copyleft model;
- that copyleft licensing is certainly interesting and may have a place as the market matures. No interviewee was against copyleft core licensing in principle (although there was consensus that a weak copyleft with clearly defined boundaries was more likely to be commercially successful).²⁷

Note that even though the interviewees selected were intended to represent a cross section of the core-developing communities, RISC-V was referred to by every interviewee. The emphasis on permissive licensing may therefore be an artefact of the relatively small sample size and a shared familiarity by the interviewees with RISC-V. It may, on the other hand, reflect a reality that RISC-V is the most prominent and widely adopted open ISA currently in use.

About the author

Andrew Katz is a partner at boutique law firm Moorcrofts LLP, based in England's Thames Valley. Andrew specialises in technology law and has a particular interest in open design and development. He can be contacted at andrew.katz@moorcrofts.com

²⁷ At the time of the original research, some of the interviewees were aware that CERN was in the process of redrafting the CERN OHL to create a new version intended to address the concerns of FPGA and ASIC developers. The approach which was being taken at that stage was by way of application-specific exceptions to the licence. The current approach (as at January 2019) is somewhat different and now allows code libraries, macros, etc. which are provided as part of the toolchain to be included as an 'Available Component' - see footnote 22.

Appendix 1

Interviewees

- Krste Asanovic, Dept EECS, UC Berkeley.
- Andrew Back, Managing Director, AB Open.
- Julius Baxter, Director, FOSSi Foundation.
- Dr Jeremy Bennett, Chief Executive, Embecosm.
- Alex Bradbury, lowRISC CIC.
- David May, Professor of Computer Science, Bristol, Founder XMOS, FEng, FRS.
- Simon Phipps, Founder, Meshed Insights Ltd.
- Dr Davide Rossi, University of Bologna.

Appendix 2

Taxonomy of Open Source Processors from OpenPiton

Processor	Architecture	Licence	Last Update to Project	Last Update to Code
aeMB	32b MicroBlaze	LGPL v3	Feb 2012	-
AltOr32	32b ORBIS	LGPL v3	Feb 2015	Jun 2014
Amber	32b ARM v2a	LGPL	Sept 2017	Nov 2015
Ariane	64b RISC-V	Solderpad	Jan 2019	-
BERI	64b MIPS/CHERI	BERI HW-SW	Mar 2017	-
CPU86	16b x86	GPL	Jun 2014	-
LatticeMicro32	32b LatticeMicro32	GPL	Oct 2017	-
LEON 3	32b SPARC v8	GPL	Dec 2017	-
MIAOW GPGPU	AMD Southern Islands	BSD 3-Clause & GPL v2	Sept 2017	-
MIPS32 r1	32b MIPS 32 r1	LGPL v3	Jul 2015	-
morlkx	32b ORBIS	OHDL	Jan 2019	Jan 2019

Processor	Architecture	Licence	Last Update to Project	Last Update to Code
openMSP430	16bMSP430	BSD	May 2018	Apr 2018
OpenPiton	64b SPARC v9	BSD 3 Clause & MIT	Jan 2019	-
OpenRISC	32b/64b ORBIS	LGPL	Nov 2018	-
OpenScale	32b MicroBlaze	GPL v3	Jan 2012	-
OpenSPARC T1/T2	64b SPARC v9	GPL v2	Nov 2008	-
or1200	32b ORBIS	LGPL	Oct 2015	Jun 2015
pAVR	8b AVR	GPL v2	Jul 2009	Mar 2009
Pico RV	32b RISC-V	ISC	Nov 2018	Nov 2018
PULP-RISCY	32b RISC-V	Solderpad	Jan 2019	-
RISC-V Boom	64b scalar RISC-V	BSD 3-clause	Jan 2019	-
RISC-V Rocket	64b scalar RISC-V	BSD 3-clause	Jan 2019	-
SecretBlaze	32b MicroBlaze	GPL v3	Dec 2012	Dec 2012
Simply RISC S1	64b SPARC V9	GPL v2	Dec 2008	-
XUM	32b MIPS32 r2	LGPL v3	Jul 2015	-
Zeroriscy	32b RISC-V	Solderpad	Nov 2018	-
Zet	16b x86	GPL v3	Nov 2013	-
ZPU	32b MIPS	FreeBSD + GPL	Apr 2015	-

Table 5: Taxonomy of differences of open source processors (table data last checked 29 January 2019).²⁸

28 Originally published in Balkind, Joseph, et al. (2016) 'OpenPiton: An Open Source Manycore Research Framework', ASPLOS '16, pp 217 – 232. <<http://parallel.princeton.edu/papers/openpiton-asplos16.pdf>> DOI: 10.1145/2872362.2872414, Table 4, updated at http://parallel.princeton.edu/openpiton/open_source_processors.php

Appendix 3

OpenCores

Name of Project	Where Project is Recorded	Brief Description	Type of Licence	Category	Licence Type
Elliptic Curve Group (ecg)	OpenCores	The Elliptic Curve Group core is for computing the addition of two elements in the elliptic curve group, and the addition of identical elements in the elliptic curve group and it is carefully optimized for FPGA	LGPL	Component	Copyleft
Reed Solomon Decoder (204,188)	OpenCores	Reed Solomon Decoder (204,188), with T=8	GPL	Component	Copyleft
Viterbi Decoder (AXI4-Stream compliant)	OpenCores	A fully configurable VHDL Viterbi decoder compliant with the AXI4-Stream interface	GPL	Component	Copyleft
Ethernet 10GE MAC (xge_mac)	OpenCores - GitHub	The 10GE MAC Core implements the Media Access Control functions for 10Gbps operation as defined in IEEE Std 802.3ae.	LGPL	Interface	Copyleft
Ethernet MAC 10/100 Mbps (ethmac)	OpenCores	The Ethernet MAC (Media Access Control), sublevel within the Data Link Layer of the OSI reference model. This core is designed for implementation of CSMA/CD LAN in accordance with the IEEE 802.3 standards.	LGPL	Interface	Copyleft
sd card controller (sdcard_mass_storage_controller)	OpenCores	The "sd card controller" is a Secure Digital Card Host Controller, which main focus is to provide fast and simple interface to SD/SDHC cards.	LGPL	Interface	Copyleft

Name of Project	Where Project is Recorded	Brief Description	Type of Licence	Category	Licence Type
Small 1-wire (onewire) master, with Altera tools integration (socket_owm)	OpenCores	This IP implements the 1-wire communication protocol (http://en.wikipedia.org/wiki/1-Wire).	LGPL	Interface	Copyleft
PCIe SG DMA controller	OpenCores	This package involves a PCIe Scatter-Gather DMA engine for Virtex5 and Virtex6 and implements MAC, Physical (Xilinx Hard and Soft IP Cores) and Transaction Layer (Custom Core) of PCIe.	LGPL	Interface	Copyleft
Wupper: PCIe DMA Engine for Xilinx FPGAs (virtex7_pcie_dma)	OpenCores	A system controller primarily designed to provide an interface to standard FIFOs (a simple Direct Memory Access (DMA) interface to the Xilinx Virtex-7 PCIe Gen3 hard block.)	LGPL	Interface	Copyleft
8/16/32 bit SDRAM Controller (sdr_ctrl)	OpenCores - GitHub	8/16/32 Configurable SDRAM data width which is Wish Bone compatible.	GPL	Interface	Copyleft
High Performance Dynamic Memory Controller (hpdmc)	OpenCores	HPDMC is part of the Milkymist System-on-Chip, the most advanced open source SoC for interactive multimedia applications.	GPL	Interface	Copyleft
VGA/LCD Controller (vga_lcd)	OpenCores	The OpenCores VGA/LCD Controller core is a WISHBONE revB.3 compliant embedded VGA core capable of driving CRT and LCD displays. It supports user programmable resolutions and video timings, which are limited only by the available WISHBONE bandwidth.	GPL	Interface	Copyleft

Name of Project	Where Project is Recorded	Brief Description	Type of Licence	Category	Licence Type
I2C controller core (i2c)	OpenCores - GitHub	I2C is a two-wire, bidirectional serial bus that provides a simple, efficient method of data exchange between devices. It is primarily used in the consumer and telecom market sector.	BSD	Interface	Permissive
UART to Bus (uart2bus)	OpenCores	The UART to Bus IP Core is a simple command parser that can be used to access an internal bus via a UART interface and provides a quick and easy way to test a new FPGA board.	BSD	Interface	Permissive
Plasma - most MIPS I(TM) opcodes (plasma)	OpenCores	The Plasma CPU is a small synthesizable 32-bit RISC microprocessor currently running a live web server with an interrupt controller, UART, SRAM or DDR SDRAM controller, and Ethernet controller.	Others	Pcore	
Tate Bilinear Pairing	OpenCores	The Tate Bilinear Pairing core is specially designed for running Tate bilinear pairing algorithm for hyperelliptic curve $y^2 = x^3 - x + 1$ defined over $GF(3^m)$, where $m=97$ and $GF(3^m)$ is defined by $x^97 + x^{12} + 2$ and it is carefully optimized for FPGA.	LGPL	Pcore	Copyleft
Amber ARM-compatible core (amber)	OpenCores	The Amber processor core is an ARM-compatible 32-bit RISC processor. The Amber core is fully compatible with the ARM® v2a instruction set architecture (ISA) and is therefore supported by the GNU toolset.	LGPL	Pcore	Copyleft
NEO430 Processor (MSP430-compatible)	OpenCores and librecores	This processor is based on the Texas Instruments MSP430 ISA and provides 100% compatibility with the original instruction set but is not an MSP430 clone.	LGPL	Pcore	Copyleft

Name of Project	Where Project is Recorded	Brief Description	Type of Licence	Category	Licence Type
minsoc	OpenCores	The Minimal OpenRISC System on Chip is a system on chip (SoC) implementation with standard IP cores available at OpenCores.	LGPL	Pcore	Copyleft
CORDIC core	OpenCores	The CORDIC algorithm is an iterative algorithm to evaluate many mathematical functions, such as trigonometrically functions, hyperbolic functions and planar rotations.	GPL	Pcore	Copyleft
T400 μ Controller (t400)	OpenCores	The T400 μ Controller is an implementation of National's 4-bit COP400 microcontroller family architecture intended to be used as a replacement for the original chip in SOCs recreating legacy systems.	GPL	Pcore	Copyleft
T48 μ Controller	OpenCores	The T48 μ Controller core is an implementation of the MCS-48 microcontroller family architecture. While being a controller core for SoC, it also aims for code-compatibility and cycle-accuracy so that it can be used as a drop-in replacement for any MCS-48 controller.	GPL	Pcore	Copyleft
openMSP430	OpenCores - librecores	The openMSP430 is a synthesizable 16bit microcontroller core written in Verilog. It is compatible with Texas Instruments' MSP430 microcontroller family and can execute the code generated by any MSP430 toolchain in a near cycle accurate way.	BSD	pcore	Permissive

Table 6: OpenCores

Appendix 4

Librecores

Name of Project	Where Project is Recorded	Brief Description	Type of Licence	Category	Licence Type
ZAP Processor	ARM librecores	ZAP is a pipelined ARM processor core that can execute the ARMv4T instruction set. It is equipped with ARMv4 compatible split writeback caches and memory management capabilities.	GPL	pcore	Copyleft
morlkc	librecores	This repository contains an OpenRISC 1000 compliant processor IP core.	MPL RC2	2.0 pcore	Copyleft
neo430	librecores	This processor is based on the Texas Instruments MSP430 ISA and provides 100% compatibility with the original instruction set but is not an MSP430 clone	LGPL	pcore	Copyleft
kpu-soc	librecores	KPU is a minimal system on chip (SoC) created for use as a testbench for the KPU core	GPL	pcore	Copyleft
PULPino	librecores	Single-core microcontroller system based on 32-Bit RISC-V cores (ETH Zurich)	SOLDERPAD HW LICENCE V0.51	pcore	Permissive
parallella-riscv	librecores	Integration of the RISC-V rocket core, inside the Zynq FPGA device of Parallella	MIT and The Regents of the University of California	pcore	Permissive
RgGen	librecores	Code generation tool for control/status in a SoC design	MIT	pcore	Permissive
picorv32	librecores	PicoRV32 is a CPU core that implements the RISC-V RV32IMC Instruction Set. It can be configured as RV32E, RV32I, RV32IC, RV32IM, or RV32IMC core, and optionally contains a built-in interrupt controller.	ISC	pcore	Permissive

Name of Project	Where Project is Recorded	Brief Description	Type of Licence	Category	Licence Type
SimpleVOut	librecores	A simple set of FPGA cores for creating video signals in various formats.	ISC	pcore	Permissive
NyuziProcessor	librecores	Nyuzi is an experimental GPGPU processor hardware design focused on compute intensive tasks. It is optimized for use cases like deep learning and image processing.	Apache v2.0	pcore	Permissive
riscv-sodor	librecores	educational microarchitectures for risc-v isa	Sodor based on the BSD 3-clause licence	pcore	Permissive
TV80 Z80-compatible microprocessor	librecores	TV80 is a Z80-compatible synthesizable Verilog core and aims to be an area-efficient core which closely mimics the original operation and cycle timing of the Zilog Z80.	MIT	pcore	Permissive
Ariane	librecores	Ariane is a 6-stage, single issue, in-order CPU which implements the 64-bit RISC-V instruction set. It has configurable size, separate TLBs, a hardware PTW and branch-prediction (branch target buffer and branch history table). The primary design goal was on reducing critical path length.	Solderpad v0.51	pcore	Permissive
RV12 RISC-V Processor	librecores	The RV12 is a highly configurable single-issue, single-core RV32I, RV64I compliant RISC CPU intended for the embedded market.	other	pcore	
openGFX430	librecores	The openGFX430 is a synthesizable Graphic controller written in Verilog and tailored for the openMSP430 core.	3-Clause BSD	interface	Permissive

Name of Project	Where Project is Recorded	Brief Description	Type of Licence	Category	Licence Type
liteeth	librecores	LiteEth provides a small footprint and configurable Ethernet core whose aim is to lower entry level of complex FPGA cores used in today's SoC such as Ethernet, SATA, PCIe, SDRAM Controller.	2-clause BSD	interface	Permissive
litesata	librecores	LiteSATA provides a small footprint and configurable SATA gen1/2/3 core whose aim is to lower entry level of complex FPGA cores used in today's SoC such as Ethernet, SATA, PCIe, SDRAM Controller	2-clause BSD	interface	Permissive
litedram	librecores	LiteDRAM provides a small footprint and configurable DRAM core whose aim is to lower entry level of complex FPGA cores used in today's SoC such as Ethernet, SATA, PCIe, SDRAM Controller	2-clause BSD	interface	Permissive
litepcie	librecores	LitePCIe provides a small footprint and configurable PCIe gen1/2 core whose aim is to lower entry level of complex FPGA cores by providing used in today's SoC such as Ethernet, SATA, PCIe, SDRAM Controller	2-clause BSD	interface	Permissive
litejesd204b	librecores	LiteJESD204B provides a small footprint and configurable JESD204B core whose aim is to lower entry level of complex FPGA cores by providing used in today's SoC such as Ethernet, SATA, PCIe, SDRAM Controller	2-clause BSD	interface	Permissive
EurySpace	librecores	Space Communication System based on CCSDS recommendations	MIT	interface	Permissive

Name of Project	Where Project is Recorded	Brief Description	Type of Licence	Category	Licence Type
HDMI2USB	librecores	The HDMI2USB project develops affordable hardware options to record and stream HD videos (from HDMI & DisplayPort sources) for conferences, meetings and user groups.	2-clause BSD	interface	Permissive
USB 1.1 Device IP Core	librecores	USB 1.1 slave/device IP core derived from USB 2.0 Function IP core save that all the high speed support logic has been ripped out and the interface changed from shared memory to FIFO based	3-clause BSD	interface	Permissive
USB 2.0 Device IP Core	librecores	This is a USB 2.0 compliant core. Due to the high interface speed, an external PHY will be required and an industry standard PHY interface for USB has been developed. This interface is called USB Transceiver Macrocell Interface (UTMI) and is WISHBONE SoC compliant.	3-clause BSD	interface	Permissive
AES (Rijndael) IP Core	librecores	AES (Rijndael) IP Core (128 bit version)	3-clause BSD	interface	Permissive
NoC Implementation Written in SystemVerilog	librecores	This is a Network on Chip (NoC) Router/Fabric implementation written in SystemVerilog.	Apache v2.0	interface	Permissive
MIPI CSI-2 Receiver	librecores	This project is an open source (MIT license) MIPI CSI-2 receive core for Xilinx FPGAs, supporting 4k resolution at greater than 30fps.	MIT	interface	Permissive
Wishbone	librecores	Wishbone is an interconnect for Systems-on-Chip.	other	interface	
scct	librecores	SCCT is a Simple Capture/Compare Timer written in Verilog. It provides multiple capture/compare channels that use a common counter.	GPL	component	Copyleft

Name of Project	Where Project is Recorded	Brief Description	Type of Licence	Category	Licence Type
libstorage	librecores	Library of RTL components for data storage	ISC	component	Permissive
The PicoBlaze-Library	librecores	The PicoBlaze-Library offers several PicoBlaze devices and code routines to extend a common PicoBlaze environment to a little System on a Chip (SoC or SoFPGA).	Apache v2.0	component	Permissive
PicoBlaze-Examples	librecores	PoC - "Pile of Cores" provides implementations for often required hardware functions such as FIFOs, RAM wrapper, and ALUs.	Apache v2.0	component	Permissive
The PoC-Library	librecores	PoC - "Pile of Cores" provides implementations for often required hardware functions such as Arithmetic Units, Caches, Clock-Domain-Crossing Circuits, FIFOs, RAM wrappers, and I/O Controllers.	Apache v2.0	component	Permissive
litescope	librecores	LiteScope is a small footprint and configurable embedded logic analyzer for use in an FPGA and aims to provide a free, portable and flexible alternative to large vendor solutions	2-clause BSD	component	Permissive
WISHBONE Interconnect IP Core	librecores	This is a WISHBONE Interconnect Matrix IP core. It can interconnect up to 8 Masters and 16 Slaves.	3-clause BSD	component	Permissive
sha256	librecores	Hardware implementation of the SHA-256 cryptographic hash function with support for both SHA-256 and SHA-224	2-clause BSD	component	Permissive

Name of Project	Where Project is Recorded	Brief Description	Type of Licence	Category	Licence Type
siphash	librecores	This is a hardware implementation of the SipHash [1] keyed hash function written in Verilog 2001 and is designed as a self contained core that performs the message block processing including initialization, compression and finalization operations.	2-clause BSD	component	Permissive

Table 7: LibreCores

Appendix 5

Analysis

		OpenCores	Librecores	Total
Type of project	Processor Core	9	14	23
	Component	3	9	12
	Interface	11	14	25
TOTAL:		23	37	60

Table 8: Summary Analysis

		OpenCores	Librecores	Total
Licences	Copyleft	19	5	24
	Permissive	3	30	33
	Other	1	2	3
Total:		23	37	60

Table 9: Licence Analysis

		Processor Core	Component	Interface	Total
OpenCores	Copyleft	7	3	9	19
	Permissive	1	0	2	3
	Other	1	0	0	1
Total:		9	3	11	23

Table 10: OpenCore Analysis

		Processor Core	Component	Interface	Total
Librecores	Copyleft	4	1	0	5
	Permissive	9	8	13	30
	Other	1	0	1	2
Total:		14	9	14	37

Table 11: Librecore Analysis

		Processor Core	Component	Interface	Total
Copyleft		11	4	9	24

		Processor Core	Component	Interface	Total
Both	Permissive	10	8	15	33
	Other	2	0	1	3
Total:		23	12	25	60

Table 12: Analysis of Opencores and Librecores

Licence and Attribution

This paper was published in the International Free and Open Source Software Law Review, Volume 10, Issue 1 (December 2018). It originally appeared online at <http://www.ifosslr.org>.

This article should be cited as follows:

Katz, Andrew (2018) 'A Survey of Open Processor Core Licensing', *International Free and Open Source Software Law Review*, 10(1), pp 21 – 46

DOI: [10.5033/ifosslr.v10i1.130](https://doi.org/10.5033/ifosslr.v10i1.130)

Copyright © 2018 Moorcrofts LLP.

This article is licensed under a Creative Commons Attribution 4.0 CC-BY available at

<https://creativecommons.org/licenses/by/4.0/>

