Fordham Intellectual Property, Media and Entertainment Law Journal

Volume 30 *XXX*
Number 4

Article 1

2020

An Instance of Open Hardware: A Different Approach to Free and Open Source Hardware Licensing

Timothy Murphy
University of Idaho College of Law, temurphy@uidaho.edu

An Instance of Open Hardware: A Different Approach to Free and Open Source Hardware Licensing

Cover Page Footnote

Visiting Associate Professor and Director, Entrepreneurship Law Clinic, University of Idaho College of Law.

The author would like to thank Barbara Cosens, Benjamin Cover, Courtney Cross, Marcy Karin, June T. Tai, and Anastasia Telesetsky for their review and comments on earlier drafts of this Article.

An Instance of Open Hardware: A Different Approach to Free and Open Source Hardware Licensing

Timothy Murphy*

As open source software ("OSS") has become more prevalent, and more widely accepted, many different OSS licenses have proliferated to provide different licensing constructs for licensors and licensees. The most popular OSS license is the GNU General Public License ("GPL"), which is protective of author rights and intended to foster an open software community. Because software source code and object code files are primarily protected by copyright, the options for license terms are relatively straightforward and well-known. To the extent patent rights become an issue, various additional provisions have been proposed to address that issue in the context of the overall, copyright-focused license.

By contrast, open hardware ("OHW"), a relatively new entrant to the open source arena, does not have a robust ecosystem of potential licenses. Because of the many different types of OHW and the different types of intellectual property that are applicable at each stage of the OHW development cycle, crafting a single license to govern all aspects of OHW has proven difficult.

This Article will explore the technical environment for OHW and the underlying principles and drivers of the open source community. The applicability of different forms of intellectual property at each stage of the OHW design/productization cycle will be discussed, along with the accompanying challenges presented by OHW. Finally, the Article will review existing licenses before proposing a

* Visiting Associate Professor and Director, Entrepreneurship Law Clinic, University of Idaho College of Law. The author would like to thank Barbara Cosens, Benjamin Cover, Courtney Cross, Marcy Karin, June T. Tai, and Anastasia Telesetsky for their review and comments on earlier drafts of this Article.

new licensing approach that focuses on permissive instantiations of OHW and distribution in non-editable form to provide a different approach to building a robust OHW community.

INTRODUCTION 1048
I. TECHNICAL ENVIRONMENT 1051
A. The Different Types of Hardware Models 1051
B. Hard IP vs. Soft IP 1053
C. The Rise of Point-of-Use Manufacturing 1055
II. OPEN SOURCE AND LICENSING COMMUNITY PRINCIPLES 1056
A. Open Source Software Principles 1057
B. Open Source Hardware Principles 1057
C. Commercial Agreements 1060
D. Licensee and Licensor Motivations 1062
III. THE CHALLENGES WITH EXISTING INTELLECTUAL PROPERTY AND CONTRACT REGIMES 1064
A. Copyright 1065
B. Trade Secret 1071
C. Patents 1074
D. Trademarks 1076
E. Contracts 1079
IV. OTHER HARDWARE LICENSES 1081
A. Commercial Licenses 1081
B. TAPR 1083
C. Apache Derivative (Solderpad) 1084
D. CERN 1085
E. Three-Dimensional Printing Open License 1085
F. GPL 1086
V. THE INTELLECTUAL PROPERTY INSTANTIATION LICENSE 1087
A. Proximity to a Negotiated License 1088
B. Instantiation Provisions 1090
C. Interface Provision 1092
D. Have-made and Sublicense Rights 1093
E. Circumvention 1094
F. Downstream Requirements 1094
G. Termination and Safe Harbor 1094
H. Compliance with Open Source Principles 1096
I. Enforceability 1099
CONCLUSION 1100
APPENDIX A 1102

INTRODUCTION

Prior to the 1990's, the notion that thousands of developers would pour their creative effort and countless hours into projects with little-to-no hope of financial reward may have sounded ludicrous, or at least one could be excused for being skeptical. However, the current popularity, and ubiquity, of OSS[1] bears testament to exactly that result.[2] OSS is widely available on the internet and boasts millions of projects covering all manner of applications.[3] Moreover, Android is the open source operating system software running on approximately 75% of the world's mobile phones,[4] and the Linux open source operating system is running approximately 47% of the websites available on the internet.[5]

Although from the perspective of a person's wallet, this software may be considered "free," working with OSS often comes with some strings attached.[6] In particular, OSS is typically licensed under one of several dozen available open source licenses.[7] While these

---

1 At the most basic level, software exists in one of two forms: source code or object code. Source code is the human-readable and human-editable version of software. Source code is processed through a specialized program called a compiler, which produces object code from the source code. Object code is the computer-readable version of the software and is the form in which most software is sold/licensed commercially. For purposes of this discussion, object code is considered not to be generally human-readable or human-editable.

2 See Brian W. Carver, *Share and Share Alike: Understanding and Enforcing Open Source and Free Software Licenses*, 20 BERKELEY TECH. L.J. 443, 443 (2005), for a thorough discussion of the origins and underlying philosophy of open source software.

3 *See, e.g.*, *The Largest Open Source Community in the World*, GITHUB, https://github.com/open-source [https://perma.cc/8RLY-RB25]; SOURCEFORGE, https://sourceforge.net [https://perma.cc/CB4Q-47W9]; BITBUCKET, https://bitbucket.org/ [https://perma.cc/46DV-S34S].

4 *Mobile Operating System Market Share Worldwide*, STATCOUNTER (July 2018), http://gs.statcounter.com/os-market-share/mobile/worldwide#monthly-201807-201807-map [https://perma.cc/EBV9-MK5P].

5 *Usage Statistics of Unix for Websites*, W3TECH, https://w3techs.com/technologies/details/os-unix/all/all [https://perma.cc/QES5-4RWQ].

6 See *What Is Free Software?*, GNU, https://www.gnu.org/philosophy/free-sw.html [https://perma.cc/5G73-JKSJ], for a discussion of the differences between "free" and "open source" software from the Free Software Foundation's perspective. But, note that, from this perspective, the cash cost to acquire the software is only one component of whether it is considered "free."

7 *See* Carver, *supra* note 2, at 452–53 n.57 (discussing various available open source software licenses).

licenses do not require a cash payment, they do impose obligations and restrictions, to varying degrees, on licensees.[8] The GPL license, for example, requires that licensees distributing modified versions of the licensed software must provide the modified version under the same license under which the original was obtained.[9] This obligation prevents licensees from using open source components to build proprietary[10] software that is withheld from the open source community.[11]

By contrast, OHW[12] has not yet enjoyed the success of OSS. While there are repositories available for OHW, the number of projects available is significantly less than those available for OSS.[13] OHW may yet have its moment in the sun, however, as three-dimensional (3D) printing, additive manufacturing, and other point-of-use manufacturing technologies become more advanced and more generally available.

Licensing of OHW is more complicated than licensing OSS because OHW projects exist for a time in software form and then ultimately exist in hardware form. Accordingly, the software forms of the project can be licensed under OSS licenses with essentially

---

[8] Some licenses, often termed "permissive" licenses, do not include the more controversial license terms, such as a copyleft provision, but do include other licensee obligations. An example would be the Apache License. *See Apache License, Version 2.0*, APACHE (Jan. 2004), https://www.apache.org/licenses/LICENSE-2.0 [https://perma.cc/F35S-UM2X].

[9] *See GPLv2, Sec. 2*, GNU, https://www.gnu.org/licenses/gpl-2.0.html [https://perma.cc/S4XX-3CT5].

[10] For purposes of this Article, proprietary software will be considered software for which the source code is not provided on a no-cost or open source basis. See Carver, *supra* note 2, at 445 n.13, for further discussion on this point.

[11] See Richard Stallman, *Copyleft: Pragmatic Idealism*, GNU, https://www.gnu.org/philosophy/pragmatic.html [https://perma.cc/6H6B-6Y5R], for a discussion of the philosophy behind copyleft restrictions.

[12] This Article makes reference to "open hardware" to avoid suggesting compliance with any particular community or licensing regime. Open Hardware is intended to cover hardware projects that are made available over the internet, or through some other electronic medium, by their creator for general use either by end users or other creators/contributors. This primarily includes projects that are offered for free but could also include projects for which a fee is charged. Many, if not most, projects that fall within the concept of Open Hardware are currently licensed under some version of open source license, using that term in the most general sense.

[13] *See, e.g.*, OPENCORES, https://opencores.org [https://perma.cc/2SBE-CX5M]; OPEN HARDWARE REPOSITORY, https://ohwr.org/welcome [https://perma.cc/E2ZB-UE4C].

the same benefits and costs as OSS projects.[14] However, these licenses do not readily translate over to the physical realm. Consequently, licensing an OHW project under an OSS license likely does not provide license terms that are applicable to the entirety of the project, and particularly, the ultimate physical hardware that is the object of the project.

As discussed in this Article, there have been several attempts to create an OHW license that mimics the perceived desirable provisions of the most prominent OSS licenses.[15] In particular, attempts have been made to craft a license that provides an enforcement mechanism for a copyleft[16] provision that would be applicable to the resulting physical objects.[17] The question arises, however, whether having a copyleft provision in an OHW license actually comports with licensee and licensor expectations and whether a more permissive approach would be better suited to the OHW context. Moreover, these licenses do not address the instantiation of the licensed materials into physical form or the provision of OHW materials in a non-source-code form.[18]

This Article discusses a different approach to OHW licensing that addresses both the software form and the hardware form of OHW projects. The goals of this licensing approach are to balance the community-building aspects of OSS licenses with the desire to provide widespread use and dissemination of OHW projects, including in non-editable, or hard, form. Part I sets forth the technical environment in which OHW operates by discussing the different types of OHW models, the different forms in which OHW

---

[14] Using an OSS license provides the same benefits for OHW projects while those projects remain in software form. As discussed herein, those benefits likely fall away, particularly on the enforcement side, when an OHW project is ultimately converted to hardware.

[15] *See infra* Part V.

[16] For purposes of this Article, "copyleft" refers to license obligations that require, as a condition of the license, release of any modifications made by a licensee under an open source license, and in some cases, the same open source license under which the materials were received. *See* Andrew Katz, *Towards a Functional Licence for Open Hardware*, 4 INT'L FREE & OPEN SOURCE L. REV. 44 (2012). The Free Software Foundation's view on copyleft is available here: https://www.gnu.org/copyleft [https://perma.cc/2BHV-LG4T].

[17] *See infra* Part V.

[18] *See, e.g.*, *GPLv2*, *supra* note 9.

projects exist, and the technological advancements that could make OHW more prevalent in the future. Part II surveys the open source and commercial licensing community principles and expectations and discusses the possible motivations of licensors and licensees participating in the OHW community. Part III looks into the challenges that OHW creates for traditional intellectual property and contract enforcement. Part IV discusses other licenses that have been proposed and/or used for OHW licensing and addresses the limitations of these licenses. Finally, Part V proposes the IP Instantiation License ("IPIL") and discusses the various provisions in the license and how they are intended to operate in the OHW community. The proposed IPIL is provided at the end of the Article.

I. TECHNICAL ENVIRONMENT

A. The Different Types of Hardware Models

The concept of Open Hardware encompasses a multitude of models, each with unique technical aspects.[19] At the direct, software-to-hardware end are field-programmable gate array ("FPGA") devices. FPGAs are programmed with a bitstream and are then able to provide the desired hardware functionality.[20] However, the FPGA-bitstream model is not significantly different from a traditional software object code-processor model,[21] and so this model will not be considered OHW for purposes of this Article.

The next level model would be where a source code-type file[22] is used to generate an object code-type file (for example, a GCODE file) through some type of translation/slicing or compilation process,

---

19 These "models" are constructs built solely to support the discussion in this Article.

20 See *FPGA Bitstream*, XILINX, https://www.xilinx.com/html_docs/xilinx2018_1/SDK_Doc/SDK_concepts/concept_fpgabitstream.html [https://perma.cc/75LQ-JHE2], for a brief discussion of how FPGAs are programmed with bitstreams.

21 In other words, the result of FPGA programming is not the creation of a new physical object. Instead, a programmed FPGA device is produced from an unprogrammed FPGA device. The author recognizes that technologists in this area could quibble with that statement, but those technical distinctions are unlikely to impact the discussion in this Article.

22 For example, a file associated with a particular computer-aided design ("CAD") program being used or a .stl file.

and then that file is used by a producer tool to create a hardware object. This model is best exemplified by a 3D printer.[23] The object code-type file in this model is not exclusively machine readable/editable, as it is possible with specialized viewers and/or expertise[24] that a person could read and manipulate the contents of these files directly. Consequently, these files are not beyond the reach of skilled technologists.[25] However, the end result is a hardware object created by only two software steps, one of which would generally be considered akin to source code and one of which is more like object code. A developer of this model of OHW would typically work at the source code-type file end of the process, but may desire to distribute their project or contribution either as source code (i.e., in a traditional open source model), as object code (i.e., not open source), or a combination of the two (i.e., source code to some licensees and object code to others).

The final level model is one in which there are multiple steps between the source code-type file and the finished hardware product. This model is exemplified by application-specific integrated circuit ("ASIC") devices. On the path from concept to hardware, an ASIC design can exist as a circuit description, a hardware design file, a netlist, a layout file, multiple photolithography masks, and ultimately, an ASIC device/chip produced by a semiconductor manufacturer.[26] Each of the files or devices in this process contain a different amount of technical accessibility, originality, expression, and human editability, leading to varying levels of protection or applicability of the different forms of intellectual property protection.

Moreover, each of these steps in the process contains more or less abstraction from the actual end physical product. For example, the first level schematic and the netlist of an ASIC generally bear no

---

[23] *See, e.g.*, Eli Greenbaum, *Three-Dimensional Printing and Open Source Hardware*, 2 N.Y.U. J. INTELL. PROP. & ENT. L. 257, 270–71 (2013) (discussing 3D printing technology).

[24] As an example, GCODE files are stored in plain text and can be directly edited by a person with appropriate knowledge of the format.

[25] This is also the case for some software object code, despite the fact that software object code is generally treated as exclusively machine-readable.

[26] There is actually more granularity possible in this description but adding the additional detail would not provide any benefit to the analysis in this Article, so it has been disregarded.

structural relationship to the ultimate physical layout of the ASIC device.[27] Instead, they are more along the lines of a functional description of how the ASIC device will ultimately operate. Conversely, the layout files bear a close resemblance to the ultimate physical ASIC device. In the ASIC model, the ultimate physical object derives from multiple software steps before the software is converted into the hardware object.[28] A developer of this model of OHW may be contributing at any source code step along the development cycle, and again, may desire to distribute their contributions in either source code form, object code form, or both.

The common thread in these different OHW models is that, ultimately, a physical object or device is produced, but such ultimate physical object is represented by software throughout the design cycle or through the bulk of the design cycle. In this Section, the distinction was drawn between source code- and object code-type forms of the project. The next Section discusses a further distinction between the forms of the software steps of OHW projects.

B. Hard IP vs. Soft IP[29]

Although it may make lawyers cringe, technologists and companies working with hardware of the types described above for ASIC designs commonly refer to the design files for such hardware as "IP." Technologists generally do not intend to use "IP" as an abbreviation for "Intellectual Property" in the legal sense.[30] Instead, "IP"

---

[27] Schematics and netlists typically contain information about what components exist in a circuit and how they are connected together, but they do not provide any information about where the individual components will ultimately be placed on the silicon substrate during fabrication. *See, e.g.*, Eli Greenbaum, *Open Source Semiconductor Core Licensing*, 25 HARV. J.L. & TECH. 131, 135 (2011). These files may also provide information about the physical/electrical characteristics of the components, but these are not set in stone and are likely to change as simulation, timing analysis, and testing reveal the need to tweak these characteristics. *Id*.

[28] It is not inconceivable that, at some point in the future, ASICs or similarly functional devices could be produced by an additive manufacturing type process like 3D printing. However, that model would likely fit somewhere between the two OHW models described herein, as there will likely be multiple software design steps prior to the final hardware manufacture step.

[29] The author is relying on his industry experience for the assertions made in this Section.

[30] If you asked one of these technologists what "IP" stands for, they would almost certainly say "intellectual property." However, it would be extremely odd to hear someone

is simply used as shorthand for the bundle of files, circuit descriptions, etc., that represent the finished hardware product. Using this parlance, technologists have come to distinguish "hard IPs" from "soft IPs." In the context of ASIC designs, "soft IPs" generally refer to the pre-layout design files, which can be modified to alter the functionality of the ASIC, while "hard IPs" generally refer to the layout files, which are not readily modifiable to alter the functionality of the ASIC.[31] It is also of significance that hard IPs are generally created using libraries for a particular foundry, using that foundry's design rules and parameters;[32] thus, for this additional reason, modification of hard IPs may be impermissible (by contract) and/or impractical.

In addition to these distinctions, IPs are often licensed as either encrypted or unencrypted. One might reasonably posit that the encryption referred to in this case has to do with controlling who has access to the files. To the contrary, however, the encryption in this case generally refers to whether or not the customer/user of the file is permitted to modify the file before instantiation into a project.[33] Accordingly, to avoid confusion, this Article will adopt the technologists' approach and refer to the bundle of design files (at any stage) that can be licensed as "IP." These can be licensed as "hard IP" or "hard form," or they can be licensed as "soft IP" or "soft form," as appropriate. The use of "IP" to mean intellectual property will be avoided and instead, the full words will be used when referring to legal intellectual property rights.

---

say something along the lines of "I will license you these five intellectual properties for twenty dollars per instantiation." Instead, the shorthand "IP" would always be used because the technologist would be referring to the licensed materials in the sense of a tangible thing (likely one or more software files) and not to the bundle of intangible rights that protect that thing.

[31] *See, e.g.*, Greenbaum, *supra* note 27, at 138 (discussing the technical aspects of "soft cores" and "hard cores").

[32] *See, e.g.*, the documentation and libraries available from GlobalFoundries at: https://www.globalfoundries.com/tech-resources/document-center [https://perma.cc/V5HK-2Y8S].

[33] See Scott Barrick, *Designing Around an Encrypted Netlist: Is the Pain Worth the Gain?*, DESIGN & REUSE, https://www.design-reuse.com/articles/18205/encrypted-netlist.html [https://perma.cc/RNS8-SW47], for a discussion of encrypted versus source code RTL and advantages/disadvantages of each.

Because some OHW projects, and particularly ASIC projects, have additional distinctions between hard IPs, soft IPs, and encrypted versus unencrypted forms, which each address different levels of accessibility and modifiability, these different forms should be accounted for in licensing OHW projects. OSS licenses are not generally directed to addressing this problem because the hard IP and encrypted forms, if there were an analog in software, would most likely not be considered open source formats suitable for open source licensing. Consequently, simply using OSS licenses to license OHW projects limits the forms in which the OHW projects can be effectively licensed.

C. The Rise of Point-of-Use Manufacturing

As 3D printing, additive manufacturing, and other similar technologies become more prevalent and more advanced, the opportunities for point-of-use manufacturing ("POUM") increase. For purposes of this discussion, POUM includes everything from household use of a POUM device (e.g., a 3D printer or successor technology), hospital or primary care facility use of a POUM device,[34] and retail use of a POUM to create products for direct sale. The latter two examples will most likely be commercial operations, while the first example could be a combination of commercial and non-commercial uses.[35] Taking as an example the history of the .mp3 file format, commercial purveyors of POUM files may struggle to keep their proprietary files from falling victim to widespread copying and distribution.[36] Similarly, online repositories of POUM files that are attempting to operate in good faith may struggle to keep proprietary/commercial files out of their repositories due to user

---

[34] This can be referred to as "point-of-care manufacturing." *See* Cosimo Orban, *The Rise of Point of Use Manufacturing*, AUTHENTISE (Jan. 4, 2018), https://authentise.com/news/2018/04/the-rise-of-point-of-use-manufacturing-authentise-weekly-news-in-review-week-66 [https://perma.cc/DMB2-ADUY].

[35] Here, commercial and non-commercial uses are used to distinguish fee-based versus non-fee-based manufacturing. In other words, the purchaser of a retail POUM product would expect to pay for the product, while a home user of POUM may purchase some files to make products, but may also swap files with others, obtain free files from online repositories, or create their own files, depending on the sophistication of the user.

[36] *See, e.g.*, Metro-Goldwyn-Mayer Studios Inc. v. Grokster, Ltd., 545 U.S. 913 (2005).

uploads.[37] Thus, both commercial vendors and site operators could benefit from a transparent and relatively standardized license construct. The continuing improvement in POUM technologies and the increase in applications of POUM make the necessity of a standardized OHW licensing construct more urgent.

Another issue, one with parallels in the OSS licensing world, is the need for commercial purveyors of POUM products/files to manage any OHW that finds its way into their products/files. If a particular piece of OHW is licensed on the right terms, a commercial vendor/operator could incorporate that OHW into their commercial product without fear of violating the license. On the other hand, if the license is ambiguous or restrictive, commercial vendors will likely take steps to avoid the incorporation of OHW into their products in the same way that commercial operations today take measures to prevent OSS from entering their products.[38] Consequently, as POUM becomes more prevalent, the availability of a license that is unambiguous and permissive could be an important factor in determining how the technology develops, and in particular, whether commercial and non-commercial players will develop collaboratively or in a disconnected manner.

II. OPEN SOURCE AND LICENSING COMMUNITY PRINCIPLES

Because OHW projects exist in both the software and hardware world, it is useful to consider the community principles applicable to both OSS and OHW in crafting a license for OHW projects. Moreover, the approaches taken by commercial licensors of IPs can be instructive in considering different approaches to OHW licensing.

---

[37] *Id.* at 952. Note that approximately ten percent of the total Grokster catalog was material that was not infringing any copyrights. *Metro-Goldwyn-Mayer Studios Inc.*, 545 U.S. at 952. This is not to suggest that Grokster was operating in good faith, which, based on the record, it surely was not. *See id.* at 941.

[38] Many companies use products/services like Blackduck, which is a commercial product designed to analyze commercial software products and report on the existence of open source components in the project. *See* BLACKDUCK, https://www.blackducksoftware.com [https://perma.cc/PC7W-HQF6],

A. Open Source Software Principles

The Open Source Initiative's ("OSI") Open Source Definition lists the following criteria for software to be considered open source: (1) free redistribution; (2) access to source code; (3) allowance for modifications distributed under the same license terms; (4) maintenance of the integrity of the author's source code; (5) non-discrimination against persons or groups of persons; (6) non-discrimination against fields of endeavor; (7) the application of license terms to all downstream recipients; (8) the license rights cannot be tied to a particular software distribution; (9) the license cannot place restrictions on other software distributed in tandem; and (10) the license must be technology neutral.[39] As shown in the next section, these same concepts, and more, are built into the Open Source Hardware Association ("OSHWA") definition. Consequently, one could expect that an OHW license that complies with the OSHWA definition is also likely to comply with the OSI definition and thus the OHW project would be in compliance with the open source community definitions independent of what form the project is distributed in. The individual requirements of the OSI definition will not be addressed independently because most of that discussion would be redundant of the discussion in the next Section with respect to the OSHWA principles.

B. Open Source Hardware Principles

According to OSHWA, to fall within the definition of Open Source Hardware, the distribution must comply with the following criteria: (1) the hardware must be released with documentation including design files, and must allow modification and distribution of the design files; (2) the documentation must specify the scope of the license; (3) the license may require that any necessary software be available as open source or easily reproducible; (4) derivative works are allowed and must be licensed under the same license as the original; (5) the license shall not restrict any party from selling or giving away the project documentation; (6) the author must be attributed; discrimination cannot occur against (7) persons or (8)

---

[39] *The Open Source Definition (Annotated)*, OPEN SOURCE INITIATIVE, https://opensource.org/osd-annotated [https://perma.cc/QZ9H-Q5EP].

fields of endeavor; (9) the license travels with the documentation without need of additional licensing; (10) the license cannot be tied to a particular product; (11) the license must not restrict other software or hardware combinations; and (12) the license must be technology neutral.[40]

Depending upon how one understands the term "released with documentation," the first requirement could be a large or small impediment to a potential licensee for incorporating OHW into their project. Fortunately, the definition goes on to clarify that the documentation could be provided with the physical product or some other "well-publicized means."[41] On the one hand, packaging/shipping the documentation with the end product could be a substantial burden on any licensee incorporating OHW into distributable products at any significant scale. On the other hand, if the term simply means that the documentation has to be contemporaneously available once hardware is distributed, a good faith licensee should be able to comply with this provision by, for example, posting the documentation online either at their own website or at an online repository. The extent to which such an approach complies with the "well-publicized means" requirement could be open to debate.[42]

One interesting question, though, is whether the documentation has to be available to the whole world upon the first distribution or only to the objects of that distribution. In other words, is it permissible for a licensee to do a limited hardware distribution and only release the associated documentation to the objects of that distribution? In practice, this approach may be more trouble than it is worth unless the documentation is actually distributed with the physical hardware. Yet it is possible that a licensee may restrict access to online documentation to only those who possess a key derived from their hardware distribution.

At first blush, the fifth requirement of the OSHWA definition would seem to resolve this question in favor of making the

---

[40] *Definition*, OPEN SOURCE HARDWARE ASSOC., https://www.oshwa.org/definition [https://perma.cc/C8AC-N8S7].

[41] *Id*.

[42] For example, does the requirement that the location of the documentation be well-publicized mean well-publicized to the recipients of a distribution or well-publicized to the OHW community in general?

documentation available to the whole world, but it is not clear how this requirement could be enforced in a real-world transaction. In particular, the licensee/distributor has the documentation obligation, but the obligation is only to ensure that the license does not restrict distribution. Recipients of a hardware distribution do not need to take a license to the documentation and thus, the provision is irrelevant to them unless they intend themselves to exercise the license that comes with the documentation that they either received physically or have the option to obtain electronically. In short, the licensee/distributor does not have a meaningful requirement to make the documentation for their particular hardware version available to anyone other than the recipients of their hardware. The OSHWA does not seem to indicate that more is required, but such an approach is not necessarily consistent with an original creator's expectations when releasing their project as open source, with the expectation that all later contributors would publicly release their contributions.[43]

Requirements 2, 4, and 9 should not be significant hurdles for OHW licensing and should not pose any surprise to licensors/licensees who are participating in good faith in the OHW ecosystem. These are the types of requirements that one should expect for any open source licensing scheme in which the primary purposes are to protect the contributions of developers and maintain an open system. Requirement 3 could be a very significant impediment in the design chain, depending on the complexity of the project, and particularly for ASIC design projects. However, this requirement is optional and so a license does not necessarily have to incorporate this provision in order to be in compliance.

With respect to the non-discrimination provisions (requirements 7, 8, and 10–12), these principles are easy to implement in a license

[43] For example, when an original developer discovers that some version of their project has been modified and distributed, they might desire to see the modifications. However, a license could comply with the first requirement of the OSHWA definition and still permit the downstream licensee to refuse to provide the modifications to the original creator (by, for example, refusing to distribute the hardware to the original creator). Of course, the original creator might be able to obtain the modified documentation from one of the persons who received a distribution; and the OHW license could not prohibit such person from giving the documentation to the original creator and still be in compliance with the definition.

agreement because they simply require the absence of certain prohibitions (i.e., no negative covenants on these issues). The challenge with these provisions is that a particular licensor, particularly someone who is not active in the OHW community on the policy side, may want to place restrictions on the use of their project, or may simply assume that such restrictions would be in any OHW license. Consequently, these provisions are straightforward to implement, but they may not be completely commensurate with licensor expectations.

On the whole, compliance with the OSHWA definition is not a significant impediment to creating a license for the software aspects of an OHW project. However, the enforceability of these provisions against the hardware aspects of an OHW project is questionable, as discussed below. Furthermore, a license directed solely at compliance with the OSHWA definition would not include the ability for licensors to license their projects in either hard form or soft form, or both, as they choose. This latter issue represents a fundamental constraint on the ability to create a license that both complies with the OSHWA definition and allows licensing in multiple forms.[44]

C. Commercial Agreements

Commercial hardware design licensors are no strangers to open source. This familiarity is not driven completely by the competitive aspect of open source solutions to the vendors' commercial products, although some vendors do explain their version of the risks inherent in choosing open source alternatives to their products.[45] Rather, commercial vendors acknowledge that some aspects of their commercial products may, and presumably do, include open source components.[46] The approach taken by these companies is to simply

---

[44] *See infra* Section V.H.

[45] *See, e.g.*, *Open Source Software*, SYNOPSYS, https://www.synopsys.com/software-integrity/resources/knowledge-database/open-source-software.html [https://perma.cc/6T59-99S8].

[46] See *Software License and Maintenance Terms and Conditions for Floating Pool Subscription License Model*, CADENCE, https://www.cadence.com/content/dam/cadence-www/global/en_US/documents/terms-and-conditions/Cadence-sub-v7.pdf [https://perma.cc/2XVF-Y433] [hereinafter CADENCE], for an example of a Cadence design software license addressing open source, and see *End-User Software License and Maintenance Agreement*, SYNOPSYS, https://www.synopsys.com/verification/prototyping/

acknowledge that open source materials may be included and then to direct their customers to the license terms applicable to the open source content, rather than the terms under which the remainder of the materials are licensed.[47]

With respect to the proprietary aspects of the commercial vendors' IPs, the license agreements typically reflect an approach of allowing the licensee to do certain things, rather than licensing particular forms of intellectual property. These licenses are designed to protect the licensors' rights in their proprietary IPs, rather than fostering any type of open community.[48] If forced to litigate their licenses, commercial vendors would likely assert contract and copyright claims, and depending on the circumstances, patent and trade secrets claims, as well. Thus, commercial vendors are not completely reliant on copyright law to protect their IPs in the way that OSS licensors generally are. Moreover, notwithstanding their desire to keep their IPs proprietary, some commercial vendors do participate in the open source community in conjunction with their proprietary products.[49] Thus, commercial vendors, even when they do use open source material in their products or participate in the open source community, are motivated to ensure that their license agreements for the proprietary aspects of their projects are as restrictive as possible and enforceable under both contract theories and intellectual property rights theories. Accordingly, these licenses (for the proprietary portions) essentially do not comply with the bulk of the OSHWA definition requirements, but they are reflective of what a non-commercial developer might try to achieve in an OHW license, particularly if they were interested in licensing their project on a free, but not open source, basis.

---

haps/synopsys-license-agreement.html [https://perma.cc/F8T3-4PM5], for an example of a Synopsys software/IP license addressing open source.

47 *See* CADENCE, *supra* note 46.

48 *See* discussion *infra* Section IV.A.

49 As an example, Wittenstein provides an open source and a commercial version of its RTOS software. *See A New Approach To Embedded Software*, HIGH INTEGRITY SYSTEMS, https://www.highintegritysystems.com/openrtos [https://perma.cc/ENT4-CG24]. The open source version is licensed under the MIT Open Source License, which has been approved by OSI. *See Licenses & Standards*, OPEN SOURCE INITIATIVE, https://opensource.org/licenses [https://perma.cc/BG2A-C6YE].

D. Licensee and Licensor Motivations

In attempting to craft a license for OHW, it is helpful to consider what the motivations of the licensees and licensors are in engaging in the licensing exercise in the first place. On the licensor side, developers of OHW may be motivated by the desire to get name recognition, work on projects outside of their day job that interest them, tinkering for their own benefit, or they may simply have an altruistic desire to help other tinkerers. Additionally, it is likely that some OHW licensors desire that their projects stay in the open source ecosystem and that no licensee can "free-ride" off of their work.[50] For purposes of this discussion, a licensee would be considered to be "free-riding" if they took an open source project, modified it, distributed the modified hardware or software, and then refused to make their modifications available to the community on open source terms.[51] Although it may be counterintuitive, simply taking OHW project materials and using them directly, without modification, to make hardware products, even if done for commercialization, is not considered "free-riding" for purposes of this Article.[52] Some licensors may not have this concern at all and, instead, may simply desire a license that does not impede the broadest reasonable distribution of their designs, even for commercialization.[53] Finally, some licensors may use open source licensing in conjunction with commercial licensing to provide a "try-it-before-you-buy-it" model for the licensees.[54]

On the licensee side, licensees may be tinkerers or commercial ventures. Tinkerers probably do not worry significantly about the license terms under which they receive OHW because they are

---

50 *See* Katz, *supra* note 16, at 41.

51 *See* John R. Ackermann, *Toward Open Source Hardware*, 34 U. DAYTON L. REV. 183, 192–93 (2009).

52 This assumption basically places OHW on the same plane as OSS that is not licensed under a no-commercialization license. For example, licensees are free to use OSS that is licensed under the GPL without modification for commercial or non-commercial purposes without fear of such commercialization, standing alone, being a breach of the license agreement.

53 *See, e.g.*, Katz, *supra* note 16, at 53 (discussing broad use of open source materials as a licensor motivation).

54 *See generally* Greenbaum, *supra* note 23 (discussing a licensing approach of making available both open source and commercial versions of a particular project).

unlikely to end up being a defendant as long as they confine their activities to just tinkering. On the other hand, tinkerers that are also modifiers (and thus future licensors) may be concerned with the license terms because the terms under which they can license their modifications are likely going to be dictated, or at least confined, based on the incoming license.

Commercial licensees present a fundamentally different situation. Commercial ventures will want to have license terms that are clear and allow the use of the OHW for which the commercial venture licensed it in the first place. From the commercial licensee's perspective, a perfect license would allow them broad use rights with essentially no restrictions, and particularly no restrictions on what the licensee can do with modifications.[55] As an example, any kind of copyleft requirement is going to be concerning for a commercial licensee because it raises concerns about compliance and costs. Moreover, the licensee may wish to keep the modifications a trade secret, which would be forbidden under the copyleft provision.[56] At a minimum, a commercial licensee would want the scope of any restrictions, and particularly copyleft restrictions, to be perfectly clear from the language of the license.

Many of the possible motivations of licensees and licensors are consistent with the OSHWA definition, but some are not. Nor are the licensees' motivations necessarily consistent with the licensors' motivations. Accordingly, some compromise would likely be necessary in crafting a license that is applicable to the broadest number of projects possible, especially where commercial licensees are going to be accommodated to at least some extent.

---

[55] This is not to suggest that commercial vendors are bad actors, are unwilling to participate in the open source community, or are generally opposed to open source principles. Instead, this simply reflects that compliance with open source terms, especially when multiple different licenses are involved, represents an additional burden, cost, and liability risk for a commercial vendor that they would likely prefer to avoid.

[56] *See generally supra* note 16.

III. THE CHALLENGES WITH EXISTING INTELLECTUAL PROPERTY AND CONTRACT REGIMES

In order for there to be a valid license, there has to be something to license. For purposes of this discussion, that something is intellectual property. However, the use of the intellectual property without a license must be unlawful in some way or the license would not be necessary, and no rational licensee would enter into it. The licensing challenge for OHW really comes at the final step, when software files are converted, or instantiated, into hardware products. Prior to that step, as discussed below, copyright laws are likely to provide adequate protection for the software files, thus making the copying, distribution, and creation of derivative works unlawful without a license. Moreover, when considered solely in the software stages, already-existing OSS licenses may provide sufficient protection for licensors simply wanting to distribute open source projects.

One concern with respect to the hardware piece of OHW is the conflict between primarily creative versus primarily functional hardware end products. This is an issue that impacts multiple areas of intellectual property, as copyright and trademark both exclude functional materials from protection, and with respect to patents, there are two different types of patents available to protect ornamental versus functional inventions. For purposes of OHW licensing, the primarily functional hardware presents the hard case. Primarily creative/ornamental hardware presents an easy case because for such hardware, which could be classified as "pictorial, graphic, and sculptural works,"[57] copyright protection may be available and thus, copyright law may provide a complete solution, similar to OSS. However, for primarily functional hardware, no such copyright protection exists (or it exists in a very weak form) and thus, any licensable rights to the hardware must come from another intellectual property regime. This Article focuses on primarily functional OHW projects, as the existence of other licenses that may cover

---

[57] 17 U.S.C. §§ 101–02 (2010); *see also* Kyle Dolinsky, *Cad's Cradle: Untangling Copyrightability, Derivative Works, and Fair Use in 3d Printing*, 71 WASH. & L. REV. 591, 609 (2014) (discussing copyright implications for 3D printing).

primarily creative/ornamental projects reduces the need to address those projects in the IPIL.[58]

A. Copyright

Copyrights protect "original works of authorship fixed in any tangible medium of expression."[59] As relevant here, the owner of a copyright has the exclusive right to reproduce copies of, distribute copies of, and prepare derivative works of the copyrighted work.[60] With respect to "pictorial, graphic, or sculptural works," the copyright owner has the exclusive right to display the work publicly.[61] In an action for copyright infringement, the copyright owner can obtain actual damages and profits or statutory damages of up to $30,000 per work.[62] One important note however, is that not all violations of a license agreement are actionable as copyright infringement.[63] For example, a person who fails to provide a notice required by a software license may only be liable for breach of contract, while a person who uses more copies of a particular software than their license allows could be liable for copyright infringement, despite the fact that both of these actions are expressly addressed in the contract language.[64]

To the extent one considers OSS to be a success story, a large amount of the credit for that success has to go to copyright law.[65] For it is copyright law, and its statutory damages scheme (in the

---

58 This is not to suggest that the IPIL is not applicable to such projects. However, creators of these projects might choose to rely on other licenses that are more protective of their copyrights in their creative works than the IPIL.

59 17 U.S.C. § 102.

60 *Id.* § 106.

61 *Id*.

62 *Id.* § 504.

63 *See* MDY Indus., LLC v. Blizzard Entm't, Inc., 629 F.3d 928, 939–40 (9th Cir. 2010) (discussing the differences between conditions and covenants in license agreements); *see also* Sun Microsystems, Inc. v. Microsoft Corp., 188 F.3d 1115, 1121 (9th Cir. 1999) (discussing under what circumstances an infringement claim is viable for a breach of license terms).

64 This may depend on how the license language is worded, however. *See* sources cited supra note 63.

65 One could also argue that the success of OSS is largely due to the self-perpetuating feature in the GPL license, the most widely used OSS license. *See* Carver, *supra* note 2, at 47–48. However, the self-perpetuating feature of the GPL license is itself dependent on copyright law for its enforceability.

United States) and automatic coverage, that put the teeth into the unilateral contracts under which almost all OSS is distributed. It would be very difficult for an open source licensor to prove up significant actual damages for the unauthorized reproduction/modification/distribution of software that the licensor willingly provides for free on the internet.[66] Accordingly, the real teeth of U.S. copyright law is in the statutory damages, providing up to $30,000 per copyrighted work for infringement,[67] even when no actual damages are proven. Furthermore, because copyright arises upon fixation, OSS developers do not need to register their copyrights unless they intend to enforce them through litigation.[68] Another facet of U.S. copyright law that is critical for the OSS movement is the coverage of both source and object code.[69] It is well-established that both source code and object code constitute copyrightable subject matter in the United States.[70]

While the availability of statutory damages also makes copyright protection an attractive option for OHW, the source code/object code distinction gets quite a bit muddier for OHW. Many authors have written on the challenges of trying to apply traditional OSS licenses to open source hardware.[71] One challenge is that, because the resulting physical objects are functional, they are not the proper subjects of copyright law.[72] Courts, and even the Supreme Court, have wrestled with the difficulty of separating the functional aspects

---

66 Note that there is some debate in the open source community about the term "free" and to what extent it is equivalent to "no cost." For purposes of this discussion, "free" simply means "without payment."

67 17 U.S.C. § 504(c)(1). Note that statutory damages can be increased up to a maximum of $150,000 per work in cases of willful infringement. *Id.* § 504(c)(2).

68 Failure to register prior to infringement may impact the availability of certain damages. *Id.* § 412.

69 Oracle Am., Inc. v. Google Inc., 750 F.3d 1339, 1355 (Fed. Cir. 2014); Apple Comput., Inc. v. Franklin Comput. Corp., 714 F.2d 1240, 1253 (3d Cir. 1983).

70 *See Apple*, 714 F.2d at 1253; *see also Oracle*, 740 F.3d at 1355.

71 *See, e.g.*, Ackermann, *supra* note 51, at 183; Greenbaum, *supra* note 31, at 134; Katz, *supra* note 16, at 46; .

72 17 U.S.C. § 102. Obviously, this restriction would not apply to artistic hardware endeavors, but the availability of copyright protection from end-to-end for artistic endeavors would potentially make the license model described here less desirable from a licensee's perspective because a license with more robust copyleft protections could be validly applied to the project.

from the expressive portions of physical works.[73] However, for the types of OHW that are available in the community, the overriding object appears to be functionality, rather than expression. In other words, the primary purpose of OHW projects is to create some functional end product. Consequently, copyright law will not likely provide end-to-end protection for these projects. However, the design files and soft IPs may have copyright protection up to the point of instantiation,[74] as further discussed below.

End-to-end copyright protection for OHW is further complicated by the different formats of the various software phases of OHW. With respect to 3D printing, there is typically AE code form, which is converted by software into an object code form, and then the object code form is instantiated by the printer.[75] Assuming the source code form contains expressive content, and is therefore copyrightable, the object code form would also be copyrightable. The difficulty is with the resulting instantiated hardware, which, as mentioned above, is primarily functional.[76] The functional hardware would not be copyrightable subject matter and thus distributing such hardware would not be a violation of the creator's copyright.[77] Consequently, for purposes of this discussion, we can assume that expressive content in 3D printing OHW is copyrightable only up to the point of instantiation.[78]

---

[73] *See, e.g.*, Star Athletica, LLC v. Varsity Brands, Inc., 137 S. Ct. 1002, 1004 (2017).

[74] For purposes of this Article, "instantiation" refers to creating a physical embodiment of a hardware design from the associated software design files.

[75] *See supra* note 22.

[76] For primarily creative/ornamental projects, we must assume that if there is expressive content in the source form materials and in the resulting instantiation, there must also be copyrightable content in the object form materials, unless we are prepared to accept that the creator's copyright protection depends on the form in which the materials are licensed by the creator.

[77] *See supra* note 72.

[78] It is worth noting that there could be expressive content in the soft form materials that do not end up in the instantiation, such as non-printed comments, which would mean that neither the instantiation nor the hard form materials would be subject to copyright were it not for the fact that object code is copyrightable. *But see* Mitel, Inc. v. Iqtel, Inc., 124 F.3d 1366 (10th Cir. 1997) (on the issue of *scènes à faire* materials). Also, Greenbaum asserts that, in the 3D printing context, the act of printing a design will infringe any applicable copyrights without a license due to the way that the design files are manipulated in the printing process. Greenbaum, *supra* note 23, at 277. The validity of this assertion does not

As an example, consider an OHW developer who creates a 3D printable plastic repair part for a household appliance. Assuming the developer includes some expressive content in the original design file, such design file, and the printer file derived from it, are likely copyrightable. However, assuming the end product of the 3D printing process is completely functional, that end product would not be copyrightable. Thus, the developer's copyright protection extends only to the software portions of the project, not to the end hardware product.

ASIC design materials are more challenging for copyright analysis.[79] At the first step in the design cycle, you have the circuit design or hardware specification. To the extent these contain expressive content, which is quite likely for all but the simplest of designs,[80] these soft form[81] materials are likely copyrightable. At the next step, software transforms the design into register-transfer level ("RTL") and/or netlists.[82] The only way these materials, which could be considered hard or soft form, would not have copyrightability is if all of the expressive content is stripped out.[83]

The next step can be multiple iterations of simulation and adjustment.[84] Although it is possible that creative content is added

---

change the analysis in this Article or the terms of the IPIL, but it may provide additional remedies in the context of 3D printing and similar technologies, in some cases.

79 *See generally* Greenbaum, *supra* note 31, at 134.

80 For any given hardware functionality, a circuit designer has many different options and circuit designs to choose from, including the use of different electrical or electronic devices and the arrangement of those devices relative to each other. *See, e.g.*, *id.* at 135.

81 Because of the complexity discussed above with respect to soft IPs, hard IPs, and encrypted versions, instead of referring to source code or object code, reference will be made to soft form and hard form versions to distinguish editable versus non-editable forms of a project.

82 *Id.* at 136. *See generally* FRANK VAHID, DIGITAL DESIGN WITH RTL DESIGN, VHDL & VERILOG 247–316 (2011) (ebook) (describing the use of RTL in device design).

83 While some have suggested that netlists are not protectable due to merger, such a result would mean one of two things, either all copyright protection is cutoff when a design is converted to a netlist and remains cut off until new creative content is added, or that the different aspects of the design move in and out of protection as the project proceeds through the process. *See* Ackermann, *supra* note 51, at 202. Each of these has conceptual challenges. Instead, it will be assumed that copyright subsists at the netlist stage because at least some aspect of the original expression contained in the original schematic carries through in the arrangement of components that is represented by the netlist.

84 Greenbaum, *supra* note 31, at 136–37 (2011).

during this process, for simplicity, it is assumed that it is not.[85] Thereafter would be a transformation into layout, done by additional software. Again, we have to assume that either the expressive content continues into the layout files or is stripped out. If the expressive content is stripped out, there should not be any further copyright protection downstream from this step; if not, there could be. Next, the layout is converted into another hard form by manufacturing the actual photomasks. Because the photomasks are physical manifestations of the layout, these are likely not copyrightable other than within the mask work protection scheme.[86]

Finally, the photomasks are used in a complex manufacturing process consisting of dozens of discrete steps, the ultimate goal of which is to instantiate the layout into physical form on silicon (or some other substrate). Again, it is this last step, the instantiation, at which the creator loses the benefit of copyright protection. Thus, for purposes of this Article, we will assume that copyright protection is available for ASIC designs up to the instantiation, so long as there is expressive content in the original circuit design/hardware specification that will end up in the final instantiation.

Another challenge on the copyright front comes from the many software transformations present in some OHW projects, and particularly for ASIC-type projects, because these transformations could give rise to questions over who holds the copyright in the compiled/transformed result. In software, many of the creative aspects of source code do not make it into the object code.[87] Moreover, some aspects of the object code arise from the compiler, not from the source code.[88] Nevertheless, copyright ownership for both

---

[85] Drawing the opposite conclusion (that creative content is added during simulation and testing) raises fundamental problems with respect to who the author/creator of this new creative content is and thus who owns the copyright in the modified work.

[86] *See* 17 U.S.C. § 901 (2012). Note that the Copyright Office takes the view that mask work protection is not a form of copyright protection, despite the fact that the two forms of protection are in the same U.S. Code title. Moreover, protection for mask works does extend to the physical devices that embody the masks. Greenbaum, *supra* note 31, at 154.

[87] As an example, all of the comments that are included in the source code as an aid in understanding what the different sections of the code are doing will not end up in the object code.

[88] If this were not the case, one would expect that different compilers would result in the same output object code if given the same input source code. However, different compilers

source code and object code with the original creator is well-established.[89] Were that not the case, significant ownership problems would arise.

As an example, if one were to consider a video game program, the source code contains all of the graphical expression that will end up in the game and the physical display, of course, displays the creative graphics. But the way in which the graphics are displayed may also be impacted and/or dictated by the compilation process with materials outside of the creator's source code. To suggest that the creative content somehow does not exist in the object code would be troubling because it would imply that the creative content was somehow added to the object code at runtime or that it disappeared while the project was in object code, but then reappeared at runtime. Such a phenomenon is not unthinkable for the source code to object code compilation because significant content may be added to the object code from the compiler and associated libraries. However, it would not make sense for the runtime transformation, particularly when the graphic content is specified in the source code.

This issue also arises for hardware because the transformation programs at each level add significant additional information to the built files, based upon foundry libraries, design rules, process-specific physical inputs, etc. An argument could be made that each of these additional materials simply contain facts and thus are not copyrightable,[90] but the creator/owner of these compilation materials may have concerns with that position. Nevertheless, taking this approach avoids the problem of having to wrestle with who owns the copyright in this combined work. The persistence of the copyright throughout the project flow in a multi-software-step process raises interesting issues that are avoided by simply taking the view that both soft form and hard form materials are copyrightable.

---

can indeed provide different object code outputs. For example, a compiler designed to produce executable code to run in a Microsoft Windows environment will obviously produce different output than a compiler designed to produce Android OS executable code. This is particularly true for compilers directed to providing all of the background code for a particular program to run effectively within a particular operating system environment, such as a compiler to create a program to run in Microsoft Windows.

89 *See* Apple Comput., Inc. v. Franklin Comput. Corp., 714 F.2d 1240, 1249 (3d Cir. 1983).

90 *See* Feist Publ'ns, Inc. v. Rural Tel. Serv. Co., 111 S. Ct. 1282, 1289 (1991).

The different models discussed above for OHW projects also present challenges for applying copyleft restrictions in OHW licenses. Copyleft provisions typically trigger at the time of a "distribution" or other dissemination of the licensed material.[91] Because distribution of a physical OHW object does not constitute distribution of a copyrighted work, unauthorized distribution does not give rise to a copyright remedy.[92] Thus, the licensor would need some other cause of action, such as breach of contract, to try to remedy the license violation, and thus would not get the benefit of the copyright statutory damages scheme. Consequently, a licensee could receive soft form materials, make modifications, compile the modifications into hard form materials, instantiate the hard form materials, and distribute the resulting hardware without complying with the copyleft obligations in the license, and the licensor would likely not have a copyright remedy available for this violation.

Setting aside the issues about functionality and third-party content infusion at the compilation/transformation stage, it is likely that copyright protection is available for all of the software phases of the OHW project cycle in the same way that it is available for OSS projects. However, that copyright protection ends when the project is instantiated into a primarily functional object. Thus, copyright provides only a partial solution for protecting an OHW project and, in particular, enforcing restrictive provisions in the license, such as a copyleft provision.

B. Trade Secret

A trade secret can be just about any confidential information that the owner has taken "reasonable measures" to keep secret and that "derives independent economic value" from being secret and "not being readily ascertainable through proper means."[93] Although it is

---

[91] *See supra* note 18. This is probably the only reasonable place in a license agreement to trigger the copyleft obligation because it would be unwieldy and/or unfair to trigger at other times, such as upon receipt of the materials, upon every modification, or upon every compilation into object code.

[92] In other words, manufacturing hardware products from OHW soft form materials would likely not be a distribution.

[93] 18 U.S.C. § 1839 (2012). For purposes of this Article, reference will be made to the Defend Trade Secrets Act version of trade secrets law. However, each state also has its own trade secret regime, most of which are very similar to the DTSA in the portions that

not uncommon to hear references to such, from a legal perspective, trade secret infringement is not actionable.[94] Rather, the cause of action relevant to trade secrets is misappropriation.[95] As opposed to infringement of other intellectual property rights, which may be unintentional, trade secret misappropriation requires a volitional and/or knowing act, such as using "improper means" to acquire the trade secret.[96] If successful in a misappropriation action, the trade secret owner may obtain an injunction, damages, and in some cases, "seizure of property necessary to prevent the propagation or dissemination of the trade secret that is the subject of the action."[97]

Trade secrets attach automatically, assuming the above requirements are met, and trade secrets can protect software source code

---

are relevant to this Article. According to the DTSA, trade secrets include "all forms and types of financial, business, scientific, technical, economic, or engineering information, including patterns, plans, compilations, program devices, formulas, designs, prototypes, methods, techniques, processes, procedures, programs, or codes, whether tangible or intangible, and whether or how stored, compiled, or memorialized physically, electronically, graphically, photographically, or in writing." *Id.*

94 *Cf.* 18 U.S.C. § 1836 (2012).

95 *Id*.

96 *Id.* § 1839. The statute defines "misappropriation" as:

> A) acquisition of a trade secret of another by a person who knows or has reason to know that the trade secret was acquired by improper means; or (B) disclosure or use of a trade secret of another without express or implied consent by a person who— (i) used improper means to acquire knowledge of the trade secret; (ii) at the time of disclosure or use, knew or had reason to know that the knowledge of the trade secret was— (I) derived from or through a person who had used improper means to acquire the trade secret; (II) acquired under circumstances giving rise to a duty to maintain the secrecy of the trade secret or limit the use of the trade secret; or (III) derived from or through a person who owed a duty to the person seeking relief to maintain the secrecy of the trade secret or limit the use of the trade secret; or (iii) before a material change of the position of the person, knew or had reason to know that— (I) the trade secret was a trade secret; and (II) knowledge of the trade secret had been acquired by accident or mistake."
>
> "Improper means" is defined as including "theft, bribery, misrepresentation, breach or inducement of a breach of a duty to maintain secrecy, or espionage through electronic or other means."

Id.

97 *Id.* § 1836.

without being preempted by the copyright laws.[98] However, trade secret protection has not been a significant factor in open source licensing in the past.[99] This is likely due to several issues. First, prior to passage of the Defend Trade Secrets Act, a licensor had to rely on state trade secrets law and did not automatically get access to Federal courts.[100] Thus, trade secret claims were not necessarily any better than state contract claims for enforcing license restrictions. Second, it would be hard to assert that the licensor was using reasonable measures to protect their trade secrets when they were providing the source code to all comers for free on the internet. Third, to establish the existence of a trade secret, the owner must show that the trade secret has economic value by not being publicly known or readily ascertainable by proper means.[101] It would be hard for a licensor to establish economic value in something that is given away for free. Fourth, it would not be improper means to analyze a software file that one is provided, or obtains, without restriction and in human-readable form.

Fifth, to establish improper means for downstream recipients of the licensed materials, the licensor would somehow have to ensure that the restrictions flow down to all subsequent users of the design files, which is a similar problem to the privity of contract issue discussed below. Similar to the contract issue, the chain of potential liability could easily be broken by the first person in the chain that, intentionally or not, neglects to flow down the license provisions. Most of these issues stem from the fact that, with respect to trade secrets, liability flows from misappropriation, generally an intentional or knowing act, while copyright and patent infringement liability can arise unintentionally or unknowingly. For essentially the same reasons as for OSS, trade secret does not seem to be a natural fit for protecting OHW licensors.

Furthering the example of the OHW developer discussed above, if the developer releases the design file through some type of OHW repository, the developer would not likely be able to assert that this

---

98 *See* Dun & Bradstreet Software Servs., Inc. v. Grace Consulting, Inc., 307 F.3d 197, 218 (3d Cir. 2002).

99 For example, the GPL does not make any reference to trade secrets. *See supra* note 9.

100 *Cf.* 18 U.S.C. § 1836(b)(3)(A)(i)(II).

101 *See supra* note 96 and accompanying text.

open source distribution included any trade secrets of the developer. This limitation on the developer arises because any recipient of the open source materials could simply review the materials in human-readable form and reveal any purported secrets contained therein.

However, if the license under which OHW materials were released did contain explicit prohibitions on, for example, circumvention and the OHW materials were distributed in a form in which the trade secrets were not discernable without circumvention, this approach could give rise to the possibility of a trade secret claim for misappropriation. Also, if the license agreement contemplated distribution of materials in hard form and prohibited licensees from converting the materials into soft form, this might also provide for the possibility of trade secret enforcement. Unfortunately, provisions like these would likely not be consistent with the OSHWA definition because a file type or restriction that required circumvention to become viewable/modifiable would not meet the first provision of the definition.[102] Moreover, in the first instance, materials licensed only in hard form could not be considered "open source" by any definition of that term which is currently used. Thus, trade secret law does not provide a natural fit for protecting OHW projects, at least not projects that are compliant with the OSHWA definition and/or are considered open source. However, it might provide a remedy (particularly, the possibility of an injunction or seizure) to the extent it covers materials that are licensed in hard form.

C. Patents

There are two types of patents that are relevant to this discussion: utility patents and design patents. Utility patents protect "any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof."[103] Design patents protect "any new, original and ornamental design for an article of manufacture."[104] To generalize, utility patents protect the functional aspects of innovation, while design patents protect innovative ornamental designs. Regardless of the type of patent, the patent owner

---

102 *See supra* Section II.B.

103 35 U.S.C. § 101 (2010).

104 *Id.* § 171.

has the right to prevent others from making, using, selling, offering to sell, or importing the patented invention.[105] If a patent owner can establish infringement, the patent owner can obtain an injunction and/or damages, which can be no less than a reasonable royalty.[106]

In the United States, patents can be obtained on mechanical inventions (i.e., physical objects), and despite some recent setbacks at the Supreme Court,[107] software is still patentable, as well. The most likely types of patent claims that would be applicable to OHW are device claims on the end product and method claims on the manufacturing process for physical objects. Despite the availability of this intellectual property protection, patents have not traditionally been seen as the primary vehicle for protecting open source innovation. One of the reasons previously discussed for this result is that, due to the cost and length of time it takes to obtain a patent, open source contributors are unlikely to pursue patent protection.[108] However, there are lower-cost and simpler options for obtaining patents in other jurisdictions that may provide the patent holder with significant rights. In particular, it is worth noting the utility model regime in China, through which patent holders face minimal examination, but still have substantial remedies potentially available to them.[109]

The availability of patent protection for licensors of OHW is not as clear of an advantage as the availability of copyright. Patents do not provide for statutory damages and thus, a licensor plaintiff would have to prove up either actual damages or a reasonable royalty for a product that is generally licensed on a royalty-free basis. The likely damages are zero. Moreover, the costs of litigating patent suits are very high. Consequently, there is very little economic upside for a good-faith OHW contributor to spend the money to obtain patents on their OHW projects.

---

105 *Id.* § 271.

106 *Id.* § 284.

107 *See* Alice Corp. Pty. v. CLS Bank Int'l, 573 U.S. 208, 227 (2014).

108 Ackermann, *supra* note 51, at 194–95. Note that this is a primarily U.S.-focused consideration.

109 *See generally Development of China's Utility Model Patent System*, CNIPA (Jan. 5, 2013), http://english.sipo.gov.cn/news/officialinformation/1121942.htm [https://perma.cc/N6KG-HNDV].

While this may be true, licensees, and particularly sophisticated licensees, are unlikely to want to assume the risk of using OHW when the accompanying license is either ambiguous or silent on patents. In particular, a licensee planning to go to market in significant quantities could face substantial per-unit damages, assuming the damages challenges discussed above can be overcome by the patent owner, or an injunction, which could be more problematic than money damages. Similarly, there is no benefit to the licensor for being ambiguous on patent rights unless the licensor intends to set a trap for licensees. Consequently, in the OHW context, there is some value, at least to licensees, of having some form of patent protection in the license. Moreover, the downside to licensors of providing a patent license for their contributions is likely very small, or de minimis, so long as the patent license is sufficiently narrow.

Accordingly, patents could provide a means to protect OHW creators, but it is unlikely in most cases that the creators will have patents covering their projects. Thus, it becomes more of an academic issue than a practical concern, except in those rare cases where a contributor does have a patent. Nevertheless, conservative licensees would likely not want to take the risk of a latent patent problem in their projects. Consequently, patents should probably be addressed in an OHW license, particularly if significant use of the project is desired by the licensor.

D. Trademarks

Trademarks protect "any word, name, symbol, or device, or any combination thereof . . . used by a person . . . to identify and distinguish" their goods or services.[110] For protection under the federal trademark regime system, the trademark must be used "in commerce."[111] A trademark owner can prevent others from using in commerce any mark that is likely to cause consumer confusion with respect to the trademark.[112] If a trademark owner can esta-

---

[110] 15 U.S.C. § 1127 (2012). For purposes of this Article, I will be referring to the federal trademark regime (the Lanham Act), but there are also state trademark regimes that could be relevant in the right circumstances.

[111] *Id.* § 1051. For purposes of the Lanham Act, the "in commerce" requirement refers to "all commerce which may lawfully be regulated by Congress." *Id.* § 1127.

[112] *Id.* § 1114.

blish infringement, the trademark owner can obtain an injunction and/or damages.[113]

Some have suggested that trademarks or certification marks could be an approach to enforcing OHW licenses.[114] While this approach has some challenges, there are practical benefits to protecting OHW through trademarks. The trademark application fees are relatively modest,[115] although maybe not so modest for a no-profit endeavor. A person can file a trademark application themselves, hire a paid attorney to file the application, which should not be that expensive, or possibly take advantage of a local law school clinic.[116] Thus, to the extent trademarks provide any tangible benefit to an OHW licensor, they could be a low-cost option.

Moreover, either at the time of filing or within a couple of years thereafter, a trademark applicant will need to demonstrate that they are using the mark in commerce.[117] Simply adding a particular word mark or design to the software files of an OHW project would probably not be sufficient to establish use in commerce; further, if the OHW contributor was able to sell or distribute the OHW outputs, and thus establish use in commerce, they probably would not be making the materials available in a free or open source form. However, it is conceivable that a licensor could develop trademark rights in association with hardware objects or software files through use of that trademark in commerce and the licensor could include the

---

113 *Id.* §§ 1116–17.

114 Katz, *supra* note 16, at 53. There has been at least one case addressing the intersection of trademarks and open source software. *See* Planetary Motion, Inc. v. Techsplosion, Inc., 261 F.3d 1188, 1198 (11th Cir. 2001).

115 *See* U.S. PAT. & TRADEMARK OFF., OVERVIEW OF TRADEMARK FEES, https://www.uspto.gov/learning-and-resources/ip-policy/public-information-about-practitioners/law-school-clinic-1 [https://perma.cc/N96M-UKGU] (discussing current fees).

116 *See Law School Clinic Certification Program*, U.S. PAT. & TRADEMARK OFF., https://www.uspto.gov/learning-and-resources/ip-policy/public-information-about-practitioners/law-school-clinic-1 [https://perma.cc/N96M-UKGU].

117 The trademark owner would need to demonstrate use in commerce at the time of filing for an in-use application or at a later date, up to approximately three years later, for an intent to use application. *See* 15 U.S.C. § 1051 (2012).

trademark in an OHW project.[118] Thus, it is possible that an OHW licensor could have some trademark rights in the OHW materials.

These trademark rights, should they exist, are not necessarily amenable to unilateral licensing in the way that open source licensing is done. To avoid loss of rights due to naked licensing, any trademark license should have some type of quality control provision.[119] But what OHW developer is going to incur the time and expense of actually policing downstream users of the developer's trademark? One possible solution would be to include a right to terminate the license upon notice if the licensee uses the materials in such a way to bring disrepute upon the licensor. Unfortunately, such provisions are necessarily ambiguous because a person's, or a company's, reputation is an inherently subjective concept. It is certainly not unreasonable to think that an OHW developer, as a licensor, might assert that commercial uses of the developer's materials could bring them into disrepute in the OHW community. This assertion would result in an effective veto right by the licensor against any licensee that the licensor decides to prohibit. In addition to the practical problem this would create for licensees, this approach would likely not be consistent with the nondiscrimination provisions of the OSHWA definition.

For all of these reasons, it would be difficult to include a trademark license in an OHW license like the IPIL for purposes of enhancing the enforcement options for the license.[120] However, foregoing the trademark license puts licensees in a potentially problematic situation. If the licensed materials include trademarked content, the OHW license likely requires the licensee to retain the

---

[118] Note that the trademark could be used in multiple different forms, depending on the particular project. As examples, the trademark could be included in the software files explicitly, it could be incorporated into the resulting hardware object (e.g., a 3D printed object), or it could be incorporated into the layout files of an ASIC project such that the trademark becomes a printed feature in the resulting ASIC (although it may only be viewable with specialized equipment and/or processing).

[119] *See* Dawn Donut Co. v. Hart's Food Stores, Inc., 267 F.2d 358, 367 (2d Cir. 1959); *see also* Tumblebus Inc. v. Cranmer, 399 F.3d 754, 764 (6th Cir. 2005); Kentucky Fried Chicken Corp. v. Diversified Packaging Corp., 549 F.2d 368, 387 (5th Cir. 1977).

[120] One approach that could be used in the open source hardware space is the use of certification marks. However, rights to use a certification mark do not come from the open source licensor themselves and thus they are not addressed in this Article.

content, but the licensee would have no license to use the trademark in commerce. For commercial operations that are trying in good faith to comply with their license obligations, this could make the incorporation of OHW a non-starter. Consequently, it seems reasonable to include trademark rights in the OHW license if the goal is to increase the use of the licensed materials. Doing so would represent some risk for the trademark owner, however, unless some form of quality control was actually exercised.

E. Contracts

Another approach to enforcement for OSS licenses is to pursue contract remedies. However, enforcing license terms through contract law has a host of problems. First, contract claims are state-law claims, so the licensor can only bring a federal claim if they can establish diversity jurisdiction.[121] Establishing diversity of citizenship could be pretty straightforward, but establishing an amount in controversy over $75,000 could be difficult for breach of a contract for materials that are provided for free.[122] Moreover, because OSS is licensed under contracts that are not negotiated and executed, the licensor would face all of the usual formation issues that arise in click-wrap or browsewrap agreements.[123] Finally, there is the issue of establishing contract damages for improperly using materials that are provided for free to the general public, or at least the interested public.[124]

For all of these reasons, licensors would prefer to have the availability of a copyright claim with its corresponding statutory damages; additionally, in effect, this may be the only practical remedy available to an OHW licensor. On the other hand, merely for the sake of minimizing potential exposure for inadvertent breaches, licensees would prefer to have licensors be required to pursue contract remedies such that the available damages are closely tied to the

---

121 *See* 28 U.S.C. § 1441 (regarding removal of civil actions to Federal courts).

122 *See id.* § 1332 (regarding the requirements to establish diversity jurisdiction).

123 *See, e.g.*, Meyer v. Kalanick, 200 F. Supp. 3d 408, 416 (S.D.N.Y. 2016), *vacated sub nom* (holding that contract was not formed through "browsewrap" type of agreement); Meyer v. Uber Techs., Inc., 868 F.3d 66 (2d Cir. 2017) (holding that contract was formed).

124 David McGowan, *The Tory Anarchism of F/OSS Licensing*, 78 U. CHI. L. REV. 207, 216 (2011) (discussing contract remedies).

actual economic injury suffered by the licensor. Under the copyright/statutory-damages regime, most infringements become at least plausibly actionable and the licensor can pick-and-choose those infringements that they choose to act on. Conversely, under contract law, most breaches are not likely to be actionable because the available damages will be far outweighed by the costs of bringing suit. Thus, licensors are confined to vindicating their rights on only a small portion of breaches that resulted in substantial revenue for the breaching licensee, and thus the possibility of substantial damages awards to the licensor.

Another challenge with contract remedies is maintaining privity between the creator and the ultimate offender/breaching licensee. If licensees are allowed to sublicense an OHW license, there is no privity of contract between the creator-licensor and the sublicensees.[125] Thus, the creator-licensor could not bring a breach of contract claim directly against a breaching sublicensee.[126] Moreover, the licensee-sublicensor that is in privity with the breaching sublicensee is unlikely to join a suit because there is no incentive to do so. This issue can be somewhat addressed by prohibiting sublicensing such that every downstream licensee receives a license directly from the original creator-licensor and each subsequent contributor. In a situation like OHW, where copyright remedies are not available throughout the entire production chain (as discussed above), this privity of contract issue becomes extremely relevant and thus, an OHW license probably should not include a sublicensable license and instead should require direct licensing from each creator/contributor to each licensee.

Notwithstanding these challenges, the ability to bring a contract claim has to be a part of an OHW license if the goal is to protect the project from end to end. Without the contract remedy, and without the availability of intellectual property remedies in the majority of cases, licensors could be left with no remedy for licensees who choose to breach the license terms. Accordingly, an OHW license

---

[125] *See* Christy's Auto Rentals, Inc. v. Mass. Homeland Ins. Co., 204 A.3d 1071, 1077 (R.I. 2019) ("A party who is not in privity of contract may not seek enforcement or interpretation of that contract.").

[126] But note that privity is not a requirement for infringement actions. *See* Fitzgerald Publ'g Co. v. Baylor Publ'g Co., 807 F.2d 1110, 1113 (2d Cir. 1986).

should have provisions, likely in the form of negative covenants, that provide the basis for a contract claim.[127]

IV. OTHER HARDWARE LICENSES

A. Commercial Licenses

Generally speaking, commercial licensors do not have to concern themselves with the intellectual property challenges described above because their IPs are not posted on the internet and widely available for download. However, these licensors do spend significant resources in license enforcement because they have a limited pool of marketable products and once the products are in the customers' hands, there are unlikely to be technological hurdles preventing the customer from reusing the materials or otherwise using the materials outside the provisions of their license. Commercial licensors, unlike individual OHW contributors, are not as averse to contract remedies because they have the resources to prosecute contract cases, and in many cases, the license agreement is a negotiated and/or signed agreement, rather than a unilateral, click-through agreement.

However, intellectual property protection is also desirable for commercial licensors, as it may provide greater venue choices, potentially higher damages, and greater risk for breaching licensees.[128] Accordingly, commercial IP licenses often are not structured as pure copyright licenses, or any other pure form of intellectual property license. Instead, commercial licenses include a confidentiality component—covering both the contract itself and the materials being licensed—and often an explicit statement that the materials constitute trade secret information.[129] Also, commercial licenses

---

[127] Such negative covenants would typically be in the form of "Licensee shall not . . ." provisions. Some examples might be: "Licensee shall not execute any sublicense related to the source materials;" "Licensee shall not distribute portions of the source materials independent of the source package;" and the like.

[128] *See* Sun Microsystems, Inc. v. Microsoft Corp., 188 F.3d 1115, 1122 (9th Cir. 1999) (discussing whether provisions of a contract give rise to contract versus copyright remedies).

[129] *See, e.g.*, *End-User Software License and Maintenance Agreement*, SYNOPSIS, https://www.synopsys.com/verification/prototyping/haps/synopsys-license-

are much more focused on what the licensee can do with the materials than what particular intellectual property rights are being licensed.[130] A licensee might be concerned that agreeing to this type of license puts the licensee at risk of falling victim to a claim that particular intellectual property was not licensed in the contract. However, there are negotiating strategies and language that a licensor can employ to largely eliminate that risk.[131]

Because the licensors are on stronger contractual footing, commercial licenses include many restrictions that would not necessarily be practicably enforceable in a license that was relying upon intellectual property as the primary enforcement mechanism. For example, one commercial license includes the following restrictions: restrict collaboration to licensed entities (para 2(4)); limit the number of copies that can be made and only allow use of the documentation with the licensed product (para 2(7 and 8)); restriction to certain foundries/nodes; confidentiality restrictions (para 6); restrictions on decompiling or reverse engineering (para 2.9(2)); prohibiting use of the licensed product to make designs for others (para 2.9(5)); prohibiting use of the licensed product or its output to develop competing designs (para 2.9(6)); prohibiting disclosure of any benchmarking results (para 2.9(8)); restrictions on transfers (para 2.11); and a full license to any feedback provided to the licensor (para 10.1).[132] Moreover, the materials that are purported to be covered by these restrictions include the licensed product, the documentation, and the design techniques included in the materials.[133]

These additional restrictions likely make these commercial licenses particularly ill-suited to the open source environment and are inconsistent with many of the provisions of the OSHWA

---

agreement.html [https://perma.cc/HFA9-UVT5] [hereinafter SYNOPSIS]; *Software License and Maintenance Terms and Conditions for Floating Pool Subscription License Model*, CADENCE, https://www.cadence.com/content/dam/cadence-www/global/en_US/documents/terms-and-conditions/Cadence-sub-v7.pdf [https://perma.cc/RGS8-2C9A].

130 In other words, the license grant provisions refer to how the IP can be used and instantiated in product designs, rather than being focused exclusively on copyright or patent rights.

131 For example, the licensee might insist on a clause in the license grant stating that the grant is "under all of Licensor's intellectual property in and to the Licensed Materials."

132 *See* SYNOPSIS, *supra* note 129.

133 *Id*.

definition. Nevertheless, at least with respect to the license term itself, they can provide some direction for how an OHW license term could be crafted to cover an OHW project from start to finish and can provide some direction on how an OHW licensor that desires to license materials in multiple forms might protect their rights, as long as the draftsperson remains cognizant of how the resource disparity between commercial and non-commercial licensors might impact the accessibility of remedies for violations.

B. TAPR

One of the early licenses proposed for open source hardware was the TAPR license.[134] The TAPR license arose from a request by developers in the ham radio community to create a license similar to GPL for hardware.[135] The hardware in that case primarily consisted of printed circuit boards (PCBs) populated with discrete integrated circuit chips or devices.[136] The TAPR license specifically considers two forms in which a design project exists: documentation (design files) and the products of manufacturing (hardware). In developing the TAPR license, John Ackermann, the author of the TAPR Open Hardware License and an attorney who specializes in software licensing,[137] specifically recognized the difficulties with applying copyright law to hardware designs and products.[138] Thus, the TAPR license incorporates both a copyright and contract remedy approach.

Additionally, Ackermann recognized that patent rights might be implicated by the manufacture and/or distribution of physical hardware, and so the TAPR license also includes a patent license, styled as an "immunity from suit."[139] In other words, to protect the open source community ethos, anyone who makes products or distributes modifications automatically grants a patent immunity to a large class of licensors, licensees, and users.[140] TAPR also arose from a

---

134 Ackermann, *supra* note 51, at 204.

135 *Id*.

136 *Id*.

137 *See The TAPR Open Hardware License*, TAPR, https://tapr.org/?page_id=5968 [https://perma.cc/L9GY-JPNQ] [hereinafter TAPR].

138 *Id*.

139 *Id.*; *see* Ackermann, *supra* note 51, at 207.

140 *Id*.

strong bias against commercial use of the licensed materials.[141] Although only the "non-commercial" version explicitly precludes commercial production, even the permissive version includes features designed to protect the open source community ethos from appropriation for commercial purposes.[142]

The TAPR license is specifically addressed to the problem of adapting open source licenses and principles to the hardware context.[143] Thus, TAPR does not address trade secrets issues and does not contemplate the delivery of materials in either hard or soft form. Accordingly, the TAPR license is most useful for OHW licensors that are focused on strict open source licensing of their projects with robust copyleft provisions, rather than flexibility to license their materials in multiple forms.

C. Apache Derivative (Solderpad)

The many challenges inherent in trying to apply copyleft principles to open source hardware were explored by attorney Andrew Katz.[144] In particular, he recognized that copyright likely does not apply to physical hardware, and even if it did, a timing issue arises as to when a breach is effective at cutting off the licensee's rights.[145] Also, the significant cost differences between creating hardware (including manufacturing costs) and writing software make acceptance of a copyleft restriction with open source hardware less likely. Accordingly, Katz proposed that a permissive license which avoids the copyleft issue, based on the Apache open source software license, would be a more suitable approach for open source hardware.

This approach is a good fit for licensors who are more interested in community building and widespread dissemination of their work but may be less desirable than the other options for those seeking robust protection for their works, especially those seeking protection against free-riders and/or commercial exploiters of OHW. This license is similar to TAPR in that it is primarily directed to open

---

141 Ackermann, *supra* note 51, at 209.
142 *Id.* at 210.
143 *Id.* at 205.
144 Katz, *supra* note 16, at 44.
145 *Id*. at 45.

source community licensors and does not contemplate distribution of materials in multiple forms.

D. CERN

For several years, CERN has been developing and refining an open hardware license ("CERN-OHL") "in the spirit of knowledge sharing and dissemination."[146] The CERN-OHL claims to be for hardware what the GPL is for software. In other words, the license purports to place obligations on licensees with respect to modifications and distributions, similar to the GPL and copyleft restrictions. The challenges with the CERN-OHL are similar to those with the TAPR license discussed above and are discussed further in Katz's article *Towards a Functional Licence for Open Hardware*.[147] Also, the CERN license does not address licensing of OHW in hard form, or both hard and soft form.

E. Three-Dimensional Printing Open License

In a 2013 article, attorney Eli Greenbaum pointed to several concerns with existing OHW licenses and proposed the Three-Dimensional Printing Open License ("TDPL") to address these issues.[148] In particular, Greenbaum was concerned that existing OHW licenses do not flow restrictions down to subsequent acquirers of hardware and thus were unsuitable for modern supply chains.[149] Moreover, Greenbaum takes the position that the unauthorized act of 3D printing is itself a copyright infringement and thus copyright covers the entire design and instantiation cycle for 3D-printed objects.[150] In order to address these issues, the TDPL includes

---

[146] *CERN Open Hardware License*, OPEN HARDWARE REPOSITORY, https://www.ohwr.org/project/cernohl/wikis/home [https://perma.cc/ZJ8P-AGRX].

[147] *See* Katz, *supra* note 16. A full discussion of the CERN-OHL challenges would be largely redundant of the other discussions in this section and is therefore omitted, particularly in view of Katz's previous exploration of the issue.

[148] *See* Greenbaum, *supra* note 23, at 277.

[149] *Id*. at 280.

[150] *Id*. at 276. On its face, this position is appealing, but it may not apply in every circumstance. For example, if a licensee is found to have a valid license, the essential step defense, may be available to avoid infringement, even if the licensee is in violation of a contractual covenant in the license with respect to the actual printing. *See* 17 U.S.C. § 117 (1998).

significant obligations on licensees, such as the insertion of notifications in design files (and on the printed objects themselves) as to where design files can be obtained. Accordingly, the TDPL is more applicable to sophisticated open source community participants and is primarily directed at the distribution of materials in soft form, similar to the other licenses discussed above.

F. GPL

This Article is not intended to be an exploration, at even a perfunctory level, of the GNU GPL and its applicability to hardware. However, a cursory review of online repositories of open source hardware projects shows that many developers choose to release their OHW projects under the GPL.[151] The challenges and concerns with using the GPL for OSS projects are well-documented.[152] Indeed, it may not even be possible to take an ASIC project that includes GPL-licensed IPs from design to fabrication without violating the terms of the GPL.[153] Moreover, the GPL license specifically states that it does not apply to hardware,[154] which makes it ill-suited for OHW. Nevertheless, for those OHW developers who are primarily concerned with protecting the software forms of their project and who wish to attach a copyleft provision to those forms, the GPL probably meets their needs. On the other hand, for those developers who are looking for end-to-end protection for their projects or permissive license terms, the GPL is probably not their best option.

---

[151] *See* OPENCORES, https://opencores.org/projects?license=GPL [https://perma.cc/7HZ3-T4Q4]; *see also Licenses*, OPEN HARDWARE REPOSITORY, https://ohwr.org/licenses [https://perma.cc/35BC-JXEN].

[152] *See* Greenbaum, *supra* note 27, at 139.

[153] *See id.* at 150 (describing how combining a soft core with a proprietary library likely creates a derivative work that would be subject to the copyleft provisions of the GPL). The licensee will most likely not be able to comply with both the GPL license and the proprietary license simultaneously. *Id.*

[154] *See Frequently Asked Questions About the GNU Licenses*, GNU OPERATING SYSTEM, https://www.gnu.org/licenses/gpl-faq.en.html [https://perma.cc/J447-PMRX]. Note that this is not part of the license itself and merely represents FSF's view on the reach of the license. Not everyone agrees with FSF's view. *See, e.g.*, Greenbaum, *supra* note 27, at 151.

Additionally, the Free Software Foundation (the maintainer of the GPL) takes the position that the GPL is not a contract.[155] In particular, the Free Software Foundation states that "licenses are not contracts" and are instead directed at permitting the licensee to do things that would otherwise be unlawful.[156] For those developers relying on contract remedies to protect the hardware aspects of their project, this alone would seem to make the GPL unsuitable. This group probably includes many OHW developers because intellectual property remedies most likely do not extend to the creation of physical objects, as discussed above. Moreover, the GPL does not incorporate the concept of an instantiation, or a suitable analog, of a software design into a hardware form.[157] Despite the fact that the GPL is a common inhabitant of OWH repositories,[158] at least for the reasons discussed above, the GPL is probably not the best fit for most OHW developers and, indeed, was never designed to be.

V. THE INTELLECTUAL PROPERTY INSTANTIATION LICENSE

In order to address the various issues discussed above and to craft a license that is more consistent with licensee and licensor expectations, the IPIL is proposed (and included in Appendix A below). The IPIL is not a pure open source license because it contemplates distribution of both soft and hard materials under the license. For those creators/licensors desiring to build community around their projects, they can release soft materials under the IPIL secure in the knowledge that licensees choosing to make modifications to the soft materials will be required to also license their modifications under the IPIL if the licensee is going to distribute soft materials, hard materials, or hardware. This requirement would be

---

155 *See, e.g.*, Eben Moglen, *Enforcing the GNU GPL*, GNU OPERATING SYS. (Sept. 10, 2001), https://www.gnu.org/philosophy/enforcing-gpl.en.html [https://perma.cc/GJ9J-PLUZ] (position statement). *But see* Artifex Software, Inc. v. Hancom, Inc., 2017 WL 4005508, *1 (N.D. Cal. Sept. 12, 2017) (denying summary judgement to defendant asserting that contract remedies are not available for breach of the GPL).

156 *See* Moglen, *supra* note 155; *see also* Ackermann, *supra* note 51 and accompanying text.

157 *GNU General Public License*, GNU, https://www.gnu.org/licenses/gpl-3.0.en.html [https://perma.cc/PS69-SPAZ].

158 *See* TAPR, *supra* note 137.

enforceable under copyright law in the same way OSS copyleft provisions are enforceable, at least up to the point of instantiation. On the other hand, creators/licensors simply wishing to release hard materials under a license that permits instantiation into hardware without modification can also use the IPIL for this purpose. Finally, licensees get the benefit of a license that is clear as to what their obligations are with respect to the different formats in which they receive soft or hard materials and how they use those materials in their projects.

A. Proximity to a Negotiated License

One of the challenges with open source software licenses (and particularly the GPL) is that some potential licensees consider the terms to be ambiguous.[159] If a licensee is essentially judgement proof, they likely are not concerned with the ambiguity of the terms because the license is unlikely to ever be enforced against them. On the other hand, for a commercial entity that may use OSS in an entire product line, they are unlikely to want to take the risk associated with agreeing to ambiguous terms that might later be enforced against them. In a commercial context, the old maxim *contra proferentem*—or ambiguous terms are construed against the drafter—might give them some comfort. And certainly, in the open source context, the terms would always be construed against the licensor if such a maxim were routinely applied. However, no court to date has used that maxim against the licensor, and instead, the courts have been receptive to enforcement of OSS license terms.[160] For this reason, for over a decade companies have been actively working to prevent open source software from infecting their commercial projects (or in some cases, even entering their infrastructure).[161]

It is possible that some licensors desire this result (i.e., preventing the use of open source software in their commercial projects) and likely have nothing to lose either way, but it does not necessarily advance the goals of the open source community. Instead, the result

---

159 *See supra* note 150 and accompanying text.

160 *See, e.g.*, Jacobsen v. Katzer, 535 F.3d 1373, 1382 (Fed. Cir. 2008); *see also* Carver, *supra* note 2, at 464.

161 *See supra* note 38 and accompanying text.

is significant costs to companies for trying to stay "open source clean," such as producing and enforcing policies against their own software teams and deploying expensive scanning tools on all software projects. On the other hand, if the licensing terms were clear on their face, and presumably suitable to the licensee, there would not be any need for all these administrative losses and headaches.

Accordingly, one goal of the IPIL is to closely approximate what a reasonable licensor and a reasonable licensee would achieve in a neutral negotiation. The ultimate license would not be pro-licensee or pro-licensor but would instead reflect a balance where the licensee and licensor each achieved some "wins," depending on their respective interests. Most importantly, each side could read the contract and understand the terms applicable to the OHW materials being licensed. Thus, the IPIL is drafted to achieve balance between the licensee and licensor and to avoid ambiguities as much as possible.

The use of particular terms that have well-defined meanings in certain jurisdictions, but not others, can also lead to license ambiguity. When including a provision that a contract is constrained by a particular choice of law, it can be easier to avoid ambiguous terms, but this relies on information outside of the contract to fully understand the terms. However, OHW isn't a solely U.S. phenomenon, thus relying solely on U.S. law for the backdrop is not necessarily consistent with licensee and/or licensor expectations. Instead, defined terms are used in the IPIL as much as possible to avoid any sort of regionality or term-based ambiguity.

When a licensing package includes materials that embody multiple forms of intellectual property, it becomes important to know exactly what intellectual property is being licensed and what is not. Indeed, certain types of intellectual property may require certain license provisions to be a valid license.[162] On the other hand, OHW licensees and licensors are unlikely to want to probe deeply into the contours of what is being licensed, particularly for complex projects like an ASIC design.

---

[162] Consider the naked licensing discussion above. *See supra* Section III.D.

One solution would be to include language in the agreement simply making the license applicable "under and to all of Licensor's intellectual property in and to the Materials." For intellectual property other than trademarks, which require special provisions, this language could work. However, it runs the risk of licensing more than the licensor intends and therefore could represent a windfall for the licensee. In particular, for patents, the license cannot just refer blanketly to all intellectual property without risking being overly broad. A single patent may have multiple claims directed to many different embodiments and to different levels of the product chain (e.g., system level, device level, product level). It is unlikely that a patentee would want to grant a broad license to all patent claims, including claims that are not directed to the licensed materials or their physical output. Consequently, a common approach is to only license essential patent claims. The IPIL takes this approach.[163]

As stated above, the goal of the IPIL is not to create the ideal OHW license from a licensee's or licensor's perspective because these parties' goals may be fundamentally inconsistent for certain terms. However, the IPIL endeavors to make a reasonable compromise on issues where there may be disagreement, while still complying, to the extent possible, with open source principles for open source distributions, and avoiding ambiguity wherever possible.

B. Instantiation Provisions

Although fairly ubiquitous in commercial hardware IP licensing, the concept of instantiation is largely missing from the OHW licensing world and the scholarly articles discussing OHW licensing. The likely reason for this is that most open source licenses and related discussions are focused on copyright principles and copyright law does not include a concept of instantiation, at least not explicitly.[164] There are likely as many definitions of "instantiate" as there are different companies or individuals licensing IPs, but the definitions may well differ more in form than in substance.

---

[163] *See infra* Appendix A, IPIL Section 1.

[164] The question of whether a public display or public performance is an instantiation of a copyrighted work will not be addressed in this Article.

Generally, a workable definition of "instantiate" would reflect the fact that one or more software files are being used to generate the physical object described by the software files. This definition works well for situations like 3D printing where there is a single software file type that a single device (the 3D printer) uses to create a single output product. In this case, it can be said that the output product is an instantiation of the software files. The situation is more complicated in, for example, ASIC design/manufacturing.[165] In that case, a GDSII file may be "instantiated" by creating a mask set described by the GDSII file. However, the mask set is not the end product. Rather, the mask set is then used in a semiconductor manufacturing process (typically done by a foundry) to ultimately create the end product. The IPIL's definition of "Instantiation" is designed to be flexible enough to accommodate both of these models.[166]

Depending on the circumstances and the vendor, there are a wide variety of limitations that can be built into the instantiation definition. For example, particularly for hard IPs, one might find a limitation in the definition to only allow manufacturing at a particular foundry. For technical reasons that do not need to be addressed here, it would not be unheard of to see a similar restriction to a particular foundry in soft IP licenses. Additionally, quantity can be addressed in the instantiation definition. For example, the definition could limit manufacturing to a certain number of finished devices, or a certain number of designs. However, such restrictions most likely do not comply with the requirements of the OSHWA definition or general open source principles. Consequently, the IPIL does not include these types of restrictions.[167] Instead, the definition of "Instantiate" used in the IPIL is relatively broad and focused on the physical creation/manufacturing of the object of the OHW

---

165 Note that common parlance would define instantiation as an instance of the circuit included in a design, not the end product.

166 *See infra* Appendix A, IPIL Section 1.

167 Restrictions like this could easily be overlaid on the IPIL by a particular licensor but doing so would remove the benefits of having an open license that is consistent across platforms and technologies. The more likely result is that particular IPs that are designed for a particular foundry process or manufacturing system would be identified as such outside of the license, for example, in the repository listing. A licensee could use the materials outside of those constraints, but this would be done at their own risk.

project, rather than on placing restrictions on the use of the licensed materials.[168]

C. Interface Provision

One of the challenges for OSS licenses, and particularly copyleft licenses, is the extent to which combining OSS with an existing project requires providing the entire project under an OSS license.[169] In an effort to make the IPIL as unambiguous as possible, the IPIL addresses this issue by including in the definition of "Modification" language about interfacing and by explicitly acknowledging that interfacing does not require open source distribution of the portions of the project that are attached to an interface.[170] Unfortunately, this approach is best-suited to complex projects like ASIC design that have explicit interfaces rather than purely mechanical projects, like a 3D printed object. For such purely mechanical projects, the interface approach does not resolve the issue of whether adding some amount of OHW to an existing project causes the whole project to need to be released under the IPIL license unless the OHW project includes an interface, which is unlikely to occur.

This challenge with simpler projects creates an intractable problem because any attempts to wrap words around the interface issue provides significant opportunity for gamesmanship and an opportunity to completely eradicate the benefit of the license to the licensor.[171] However, the Interface provision in the IPIL attempts to straightforwardly address the issue of modification for the cases in which such modification is most likely to occur, by specifically requiring an interface in the licensed materials, such that all modifications would not be automatically considered subject to this carve-out.

---

168 *See infra* Appendix A, IPIL Section 1.

169 Katz, *supra* note 16, at note 3.

170 *See infra* Appendix A, IPIL Section 1.

171 As an example, an OHW license might state that combining the licensed materials into a larger project does not render the license obligations applicable to the larger project. But a nefarious licensee could simply argue that any significant modifications or additions to the licensed materials constitute a separate project, and thus there is no obligation to distribute under an OHW license anything that the licensee has done. This would effectively render the OHW license obligation-free from the licensee's perspective.

D. Have-made and Sublicense Rights

In the world of 3D printing, have-made rights may not be that important.[172] Licensees of OHW are likely to have their own printing facilities and thus would not need have-made rights. This is especially true for POUM. However, it is possible that a licensee might need to avail themselves of a third-party manufacturer/printer,[173] and so it does not hurt to have that right explicit in the IPIL, even for 3D printing projects. At the other end of the spectrum, ASIC developers are quite unlikely to personally own the equipment needed to instantiate an ASIC design. These licensees will almost certainly need to avail themselves of foundries, packaging houses (sometimes referred to as OSATs), and/or other contractors in the manufacturing chain. Thus, it is important that the IPIL include have-made rights if it is going to be useable for OHW that is tied to multi-step design and manufacturing projects.

As discussed above,[174] a sublicense provision raises issues for any license that relies on contract remedies as an enforcement mechanism. Because at least a portion of the product chain is not covered by copyright protection, the IPIL does not include a sublicense provision and instead relies upon direct licensing throughout the license chain.[175]

---

[172] The rights to have a project manufactured by a third party are referred to as "have-made rights" here to maintain consistency with typical license terminology, but the IPIL uses the "have-instantiated" terminology to maintain internal consistency with the definitions.

[173] In particular, a licensee may desire to use a third-party printing facility that provides the capability to print large objects, use specialized materials, or produce objects more rapidly than consumer models.

[174] *See supra* Section III.E.

[175] In this respect, the IPIL is similar to the GPL, which states:

> Sublicensing is not allowed; section 10 makes it unnecessary . . . . Automatic Licensing of Downstream Recipients. Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

GPLv3, GNU, https://www.gnu.org/licenses/gpl-3.0.html [https://perma.cc/Z473-SR2N].

E. Circumvention

As discussed above,[176] with the right agreement provisions, a trade secret misappropriation claim might be viable and provide some additional protection for licensors, but it would not comply with the OSHWA definition. Nevertheless, one goal of the IPIL is to allow licensing of hard IPs or other hard form materials, which are not source code. Accordingly, the IPIL includes an anti-circumvention provision to maintain the viability of such a claim in circumstances where it is available and to allow for the distribution of hard form materials where desired.[177] Of course, the choice to distribute hard IPs or other materials in hard form is only available to the original licensor, as all subsequent licensees are required to distribute documentation in the form in which they received the licensed materials.[178]

F. Downstream Requirements

The IPIL includes those downstream requirements that are reasonably enforceable in the construct of OHW, but in as unambiguous a form as possible. These terms depend on what form the licensee received OHW materials. If, for example, the licensee receives hard IPs or hard form materials, the licensee is not obligated to disclose any soft form materials unless they modified the hard form materials.[179] In that case, they would be required to disclose. This disclosure requirement takes care of the situation where a licensee simply wants to use the licensed materials to make hardware in an unmodified form. Conversely, for soft IPs or soft form materials, copyleft provisions would be applicable to any modifications made and distributed by the licensee.[180]

G. Termination and Safe Harbor

One of the challenges with open source licensing is that once materials are available open source, there is no means to terminate a licensee, because the licensee can simply obtain another copy of

---

176 *See supra* Section III.B.
177 *See infra* Appendix A, IPIL Section 3.f.
178 *See infra* Appendix A, IPIL Section 3.c.
179 *Id*.
180 *Id*.

the materials and a new license. The IPIL addresses this concern by including a provision that future licensing of particular materials is not permitted without explicit permission from the licensor once a license has been terminated.[181] To enhance the enforceability of this provision, the IPIL states that any such subsequent license attempt will be considered in bad faith and/or fraudulent.[182] This solution does present additional challenges. In particular, obtaining permission for a subsequent license will probably be quite difficult for a particular licensee in a project with any significant number of contributors. However, this obligation will probably only arise in a small minority of cases and only with respect to licensees that have had their licenses terminated for good cause.

The IPIL adds a safe harbor provision to account for licensees that make compliance mistakes, but are otherwise acting in good faith.[183] The safe harbor provides that the license will not automatically be terminated in the first instance of a violation.[184] Instead, upon notice from a licensor of a compliance issue, the licensee can respond that the noncompliance was inadvertent and thus avail themselves of a fourteen-day period to remedy the violation.[185] This provision is most likely going to apply in a situation where a licensee has not provided the required documentation in association with a distribution of modified material or hardware. Thus, fourteen days should be sufficient for the licensee to remedy this deficiency (by posting the required materials) and becoming compliant once again. The goal of this provision is two-fold: (1) to provide licensors with a process to enforce community norms against good-faith licensees without having to file suit; and (2) to provide good-faith licensees an opportunity to remedy the problem and thus get back into compliance without having the license terminated.

---

181 *See infra* Appendix A, IPIL Section 4.

182 *Id*.

183 *Id.*; *see* Carver, *supra* note 2, at 464 (stating "the GPL has been primarily enforced through private negotiation and settlement agreements. This process been successful thus far because most alleged violators have apparently been eager to correct any defects in their compliance.").

184 *See infra* Appendix A, IPIL Section 4.

185 *Id.*

H. Compliance with Open Source Principles

Although not necessarily reflective of every licensee and/or licensor's expectations with respect to an open source license, this Article will use the OSHWA definition as a benchmark by which to measure the provisions of the IPIL.[186] The first requirement of the definition is that hardware must be released with documentation and design files.[187] Additionally, the definition requires that design files be provided in the "preferred format for making changes."[188] The most simplistic approach to addressing this requirement is to simply require that licensees distribute the documentation with each item of distributed hardware. However, this may not be practical for any reasonable scale; thus, a more practical approach may be to simply allow the licensee to make the documentation publicly available on the internet, which is permitted under the definition.[189] The IPIL addresses these requirements in Section 3.c by placing two obligations on a licensee: first, the licensee is required to provide modified documentation to any and all requestors; and second, the licensee is required to post the modified documentation at either the website from which the originals were obtained or another repository used for the posting of OHW materials.[190]

A final requirement in the first section of the OSHWA definition is that "[d]eliberately obfuscated design files are not allowed."[191] Because the IPIL contemplates that materials may be received and distributed under the license in hard form (or as hard IPs), the IPIL cannot comply with this requirement. However, this gap should not result in a significant deviation from licensor expectations because it would be unreasonable for a licensor to provide only hard form materials and expect licensees to provide soft form materials. Nevertheless, it is worth noting that this is an area where the IPIL does not comply with the language of the OSHWA definition.

---

186 *See supra* Section III.B.

187 *Definition (English)*, OPEN SOURCE HARDWARE ASSOC., https://www.oshwa.org/definition [https://perma.cc/C8AC-N8S7] [hereinafter OSHWA].

188 *Id*.

189 *Id*.

190 *See infra* Appendix A, IPIL Section 3.c.

191 *See* OSHWA, *supra* note 187.

The second section of the OSHWA definition addresses scope and requires that the documentation specify what portion of the design is released under the license.[192] Although this situation seems unlikely to routinely arise in practice, it is possible that a contributor could license a project under multiple licenses, including the IPIL, because of the interface provision discussed above.[193] In other words, a licensee may obtain a portion of a larger project under the IPIL and the remainder under one or more other licenses (or choose to license other portions of the project created by the licensee under a separate license[194]). In that situation, or any other where materials are being provided under multiple licenses, the IPIL requires notice of such in Section 3.d.iii and so the IPIL complies with this requirement of the definition.[195]

The third statement in the OSHWA definition relates to necessary software, but it is permissive, not mandatory.[196] Because of the variety of OHW projects intended to be covered by the IPIL, the IPIL does not address this statement. In particular, OHW materials directed to an ASIC (or a portion thereof) may require EDA software to be usable for their intended purpose, but it does not seem reasonable to require the licensee to verify that an acceptable open source version of EDA software is available. Because the requirement is permissive, IPIL is considered to be in compliance with the definition even though there is no specific language in the IPIL directed to this issue.

The fourth OSHWA requirement is that the license must allow modifications and derivative works, distributed under the same license, and allow for the manufacture, sale, and distribution of products and modified design files.[197] The license grant in the IPIL (Section 2) explicitly complies with this portion of the definition by

---

192 *See id*.

193 *See supra* Section V.C.

194 Note that this would only be permissible if these other sections created by the licensee are not modifications of the licensed materials, as defined in the IPIL. *See supra* Section V.C; *see also infra* Appendix A, IPIL Section 1.

195 *See infra* Appendix A, IPIL Section 3.d.iii.

196 *See* OSHWA, *supra* note 187.

197 *Id*.

making all of these activities licensed activities, and requiring (in Section 3.c) that all modifications be licensed under the IPIL.[198]

The fifth OSHWA requirement prohibits restrictions on the sale or "giving away" of the project documentation.[199] The IPIL makes clear in Section 3.a that associated services, such as training, support, and warranty protection, may be provided at any price, but that the licensee cannot charge a fee or royalty for distribution of the modified documentation.[200] The IPIL does not place any requirements for royalty or fee-based sales of documentation. Accordingly, the IPIL complies with this requirement of the OSHWA definition.

The sixth OSHWA requirement is with respect to attribution, but again is permissive, rather than mandatory.[201] Notwithstanding the permissive language in the definition, the IPIL requires that all attributions and rights notices in the materials be carried forward with any further distributions.[202] Accordingly, the IPIL complies with this permissive statement.

The seventh, eighth, and twelfth OSHWA requirements provide that the license must not discriminate against persons or groups or fields of endeavor and must be technology neutral.[203] The IPIL does not include any provisions that would run afoul of these restrictions, and in particular, allows commercialization. Accordingly, the IPIL complies with these requirements.

The ninth OSHWA requirement provides that the license terms must flow down to subsequent acquirers without the need for execution of additional licenses.[204] This requirement is addressed in Section 3.a and thus the IPIL complies with this requirement of the definition.[205]

The tenth and eleventh requirements of the definition state that the license must not be specific to a particular product and that it

---

198 *See infra* Appendix A, IPIL Sections 2, 3.c.

199 *See* OSHWA, *supra* note 187.

200 *See infra* Appendix A, Section 3.a. This is required to comply with the first requirement of the OSHWA definition. *See* OSHWA, *supra* note 187.

201 *See* OSHWA, *supra* note 187.

202 *See infra* Appendix A, IPIL Section 3.b.

203 *See* OSHWA, *supra* note 187.

204 *Id*.

205 *See infra* Appendix A, IPIL Section 3.a.

must not place restrictions on aggregations with other hardware or software.[206] The IPIL does not include any restrictions on particular products and specifically allows interfacing with other materials on an unrestricted basis in the license grants of Section 2.[207] Thus, the IPIL complies with these requirements of the definition.

In sum, the IPIL complies with most of the requirements of the OSHWA definition of Open Source Hardware. The one primary area of noncompliance, the disallowance of the deliberate obfuscation of design files, is driven by the breadth of activities contemplated under the IPIL and so there is not a clear path to bringing the IPIL into compliance without sacrificing compatibility with alternate distribution models, such as distribution in hard form.

I. Enforceability

The IPIL is designed to be enforceable under copyright law for the software forms of an OHW project to the same extent as an OSS license. For example, the restrictions regarding modifications and distributions are stated as conditions on the license, rather than contractual covenants.[208] Accordingly, licensees that fail to comply with the license terms with respect to the software forms, including the copyleft provisions, likely could be held liable under a copyright infringement claim. As discussed above,[209] the availability of a copyright infringement claim probably does not extend to the instantiated hardware.[210] Thus, other remedies are required.

Breach of contract remedies are the most obvious enforcement mechanism for violations associated with the hardware, rather than the software. While the IPIL maintains the viability of contract remedies, the issues discussed above[211] with contract enforcement generally also apply to the IPIL. To the extent a particular licensor

---

206 *See* OSHWA, *supra* note 187.

207 *See infra* Appendix A, IPIL Section 2.

208 *See infra* Appendix A, IPIL Section 3.b., 3.c.

209 *See supra* Section III.A.

210 3D printing might be a special case in which the act of loading the software onto the printer constitutes copyright infringement and might provide copyright remedies. *See* Greenbaum, *supra* note 23, at 277. For purposes of this Article, this special case will not be addressed separately from the overall OHW context.

211 *See supra* Section III.E.

owns a patent covering a portion of the licensed materials, a patent infringement suit could be brought for violations of the IPIL, but as discussed above, this is a relatively unlikely scenario.

Because the IPIL contemplates licensing of hard form materials, or hard IPs, it maintains the viability of a trade secret misappropriation claim for materials licensed in hard form. However, a trade secret claim is not a panacea for the failings of copyright law with respect to hardware. In particular, a trade secret misappropriation claim would not apply to every violation of the license terms and thus would not provide the same scope of coverage as copyright. Instead, the trade secret misappropriation claim would only apply when hard form materials are licensed and the licensee decompiles the materials, decrypts the materials, or uses some other process to determine the trade secret information. Thus, the availability of the trade secret misappropriation claim provides a narrow scope of protection for that subset of licensors who want to make their projects available but retain underlying trade secrets in the project.[212]

In sum, copyright remains the primary enforcement mechanism for the IPIL in the same way as for OSS licenses and some other OHW licenses. Thus, the copyleft aspects of the IPIL are largely enforceable with respect to the software aspects of an OHW project, but not the hardware aspects. Consequently, aggrieved licensors have to look to patent, trade secret, or contract law to enforce violations of the IPIL at the hardware stage.

CONCLUSION

The success of open source software sets an example for the potential future of open hardware. Unfortunately, OHW does not enjoy the same end-to-end benefits from copyright law that OSS receives. Accordingly, the licensing construct for OHW is more difficult and is not well-suited to enforcing community norms through, for example, a copyleft provision. Several licenses have been proposed to account for this disparity, but they have generally not

[212] As an example, a licensor may provide both an "open" and a "proprietary" version of a particular project and thus may choose to release the "open" version only in hard form to maintain the viability of the proprietary version.

addressed the entire design cycle of OHW because they do not address the physical aspects of the OHW projects and they also do not contemplate distribution of hard form materials. Moreover, these licenses do not necessarily reflect the goals of OHW contributors, who may be less concerned with enforcing open source principles and more concerned with wide dissemination and/or use of their contributions and the ability to distribute hard form materials. Commercial licenses do address some of these issues, but commercial vendors are in a better position to avail themselves of contract remedies than a typical OHW contributor and are a fundamentally different licensor than an OHW licensor.

These issues and the desire to craft an OHW license that represents a balanced compromise between licensor and licensee objectives gave rise to the IPIL. The IPIL includes a copyleft provision, which is primarily applicable to the software aspects of an OHW project. The IPIL specifically addresses instantiations of the licensed materials and contemplates that materials may be provided in either hard or soft form. Moreover, the IPIL is clear as to what exactly the licensee has a right to do and what the licensee's obligations are under the license. By taking open hardware licensing in this different direction, the IPIL should provide a good framework for OHW licensing going forward so that both licensees and licensors can reap the benefits of future technological advancement that can benefit from a robust OHW ecosystem.

APPENDIX A

The IP Instantiation License (IPIL)

1. Definitions.

"Affiliate" means any individual or entity acting under the direction of Licensee or acting with the intention of furthering Licensee's interests.

"Associated Services" means setup, training, support services, warranty services, and the like, associated with the Licensed Materials and/or Modifications but in no event includes distribution of Documentation, as required herein.

"Compile" means to use software to transform Soft Form materials into Hard Form, where the resulting Hard Form materials describe a Hardware embodiment of the Soft Form materials. "Compilation" means the resultant materials from Compiling Soft Form materials.

"Documentation" means the Licensed Materials in the format received or obtained by Licensee with all Modifications incorporated into the Licensed Materials in the same format.

"Hard Form" means a description for one or more physical objects in a form that would be recognized by practitioners in the relevant field as not being suitable for human readability and modification, where such descriptor requires one or more transformations and or manufacturing steps to become the physical object(s). Hard Form materials include, for example, mask layout files, compiled source code, encrypted or obfuscated design files, and 3D printer machine files.

"Hard Materials" means any and all files and/or other materials provided to or obtained by Licensee in Hard Form under this License Agreement, other than by Compiling Soft Materials and/or Modifications.

"Hardware" means the physical object(s) produced when Licensed Materials and/or Modifications are Instantiated.

"Instantiate" means to transform Soft Materials and/or Hard Materials into a physical object, whether through a single step or multiple steps and whether or not combined with Secondary

Materials, where the physical object is an embodiment at least partially described by the Soft Materials and/or Hard Materials.

"IPIL" is an abbreviation for IP Instantiation License and means this License Agreement.

"License Agreement" means this agreement and all of the terms and conditions herein, consisting of Sections 1–7.

"License Terms" means all of the terms and provisions of this License Agreement.

"Licensee" means any individual or entity receiving or obtaining the Licensed Materials and availing themselves of the license rights granted herein.

"Licensed Materials" means the Soft Materials and Hard Materials. Any Soft Form and/or Hard Form materials that are provided with an explicit statement that they are licensed under the IPIL, provided on a website or online forum including a statement that such materials are licensed under the IPIL, obtained through a process including a clickwrap or browsewrap license including a statement that such materials are licensed under the IPIL, or distributed in other electronic form with a statement that such materials are licensed under the IPIL, are Licensed Materials.

"Licensor" means, individually and collectively, the creators of, contributors to, and/or intellectual property rights owners of the Licensed Materials.

"Modification" means any materials, whether in Soft Form or Hard Form, that are derived from the Licensed Materials through a process of making edits or additions to the Licensed Materials, and specifically includes derivative works under U.S. copyright law. To the extent the Licensed Materials include interfaces for attachment to, or combination with, other materials, Modifications do not include materials that are attached to or combined with the Licensed Materials through such interfaces. As an example, if the Licensed Materials constitute a digital logic processing circuit including an input/output interface, combining such materials with an RF transceiver circuit through the input/output interface to form a portion of an ASIC would not constitute a Modification of the Licensed Materials.

"Modify" means to create a Modification.

"Rights" means any and all copyright, trademark, patent, and trade secret rights of the Licensor(s) included or embodied in the Licensed Materials, provided that, with respect to patent rights, Rights only include those patent claims owned by a Licensor that are necessarily infringed by those portions of the Licensed Materials contributed by such Licensor and the Hardware resulting therefrom.

"Rights Notice" means a file or embedded data included with Soft Materials and/or Hard Materials that includes attributions, license statements, copyright notices, trademark notices, and/or other indications of intellectual property rights ownership.

"Secondary Materials" means foundry libraries, cell libraries, and the like that are provided separately from the Soft Materials and/or Hard Materials but are necessary or desirable to Instantiate or Compile the Soft Materials and/or Hard Materials. Secondary Materials may be licensed under any license, including proprietary licenses.

"Soft Form" means a description for one or more physical objects in a form that would be recognized by practitioners in the relevant field as being suitable for human readability and modification, where such description requires one or more compilations, transformations, and/or manufacturing steps to become the physical object(s). Soft Form materials include, for example, source code, circuit netlists, 3D printing design files, VHDL files, and soft IPs.

"Soft Materials" means any and all files, drawings, documentation and/or other materials provided to or obtained by Licensee in Soft Form under this License Agreement.

2. License Grant

Subject to the conditions set forth in Section 3 herein, each Licensor hereby grants to Licensee, under all of such Licensor's Rights in the Soft Materials and Hard Materials, respectively, a perpetual, worldwide, non-exclusive, royalty-free license to:

a. With respect to the Soft Materials, (1) reproduce, Modify, simulate, validate, test, integrate with other design materials, and distribute such Soft Materials and Documentation in Soft Form;

(2) Compile the Soft Materials and Modifications into Hard Form; and (3) Instantiate, and have Instantiated, such Soft Materials, including those Compiled into Hard Form in accordance with subsection a.(2), into Hardware, and use, sell, offer for sale, distribute, modify, import, and export such Hardware.

b. With respect to the Hard Materials, simulate, validate, test, and integrate with other design materials the Hard Materials, distribute such Hard Materials in Hard Form, Instantiate and have Instantiated such Hard Materials into Hardware, and use, sell, offer for sale, distribute, modify, import, and export such Hardware.

3. Conditions

As stated in Section 2, the licensed rights herein are subject to the following conditions (and are otherwise irrevocable):

a. Licensee shall not make any Modifications or distribute any Licensed Materials, Documentation, or Hardware except in strict compliance with this License Agreement. The exercise by Licensee of any rights licensed under this License Agreement shall be deemed acceptance of this License Agreement and all of the License Terms herein. Licensee may provide Associated Services and Hardware at any price. However, Licensee may not charge a fee or other royalty for distributing and making available the Documentation, as required by this Section 3.

b. Licensee shall retain and/or reproduce all copyright, patent, trademark, and attribution notices in the Licensed Materials with any copies, excerpts, or other reproductions made by Licensee or its Affiliates, except to the extent that a particular notice does not apply because the associated material has been removed from the Licensed Materials in the excerpt. Licensee may

comply with this condition by retaining and/or reproducing the Rights Notice provided in the Licensed Materials, if provided, in all Documentation. Licensee may, and should, modify the Rights Notice and/or any other attribution statements in the Licensed Materials with rights and/or attributions statements of Licensee appropriate for the Modifications made by Licensee.

c. Licensee shall license all Documentation under the License Terms and shall not license such Documentation under any other license. Licensee may meet this condition by providing a notice within or with such Documentation referring to this License Agreement by name and/or providing a web link to this License Agreement. Any distribution by Licensee or its Affiliates that does not indicate that it is licensed under the License Terms shall be deemed to have been licensed under the License Terms.

d. If Licensee makes any Modifications and subsequently Compiles or Instantiates such Modifications and distributes the Modifications, Compilations, Documentation, or resulting Hardware in any form, regardless of whether such distribution is in conjunction with any payments to Licensee, Licensee shall:

    i. provide the Documentation to any and all requestors within fourteen days of a request, provided that this requirement expires one year after the last-in-time distribution;

    ii. post the Documentation on a website or forum at which open source hardware materials are routinely posted for distribution or post the Documentation on the website or forum from which the

associated Licensed Materials were obtained by Licensee; and

iii. to the extent the distribution includes materials licensed under any license other than this License Agreement, specify in the Documentation which portions of the materials are so licensed and what license applies to such materials.

e. If Licensee or its Affiliates distribute any Hardware, whether for commercial gain or not, Licensee shall include with each distributed Hardware:

i. If the Hardware embodies any Modifications: a statement, preferably including a web address, indicating where the recipient of the distribution can obtain the Documentation; or

ii. If the Hardware embodies only the unmodified Licensed Materials: a statement, preferably including a web address, indicating the source from which Licensee obtained the Licensed Materials.

f. If the Licensed Materials are provided to Licensee only as Hard Materials, Licensee shall not decompile, disassemble, or use any other process to render the Hard Materials into Soft Form or otherwise make the Hard Materials human-viewable or modifiable or determine confidential or trade secret information included in the Hard Materials.

g. If Licensee or its Affiliates institute any litigation against any Licensor (including any governmental administrative proceeding or a cross-claim or counterclaim in a lawsuit) alleging that the Soft Materials or Hard Materials infringe or misappropriate any intellectual property right of the Licensee, then any licenses granted to

Licensee under this License Agreement shall terminate as of the date of such institution.

h. If Licensee exercises the 'have Instantiated' rights in the license grant, Licensee shall ensure that the person or entity providing the instantiation services is obligated to provide the services exclusively on Licensee's behalf, under Licensee's direction and control, and is prohibited from making additional copies, distributions, or Instantiations other than with respect to the services for Licensee.

4. Safe Harbor and Termination

For clarification and avoidance of doubt, any violation of the conditions in Section 3 herein shall immediately terminate the licenses set forth in Section 2 above. Notwithstanding the foregoing, upon notice from a Licensor of a first violation, if the Licensee notifies such Licensor in good faith that such violation was inadvertent, the termination of the licenses will be delayed for fourteen days from the date of the Licensor's notice to give Licensee an opportunity to remedy the violation. Should Licensee remedy the violation within such fourteen-day period, the licenses will not terminate. This opportunity to remedy the violation shall apply solely to the first violation for which Licensee receives notice and all subsequent violations shall be governed solely by the first sentence of this Section 4. With respect to the Licensed Materials, if this License Agreement is terminated for Licensee or its Affiliates pursuant to this Section 4, Licensee shall not be licensed to such Licensed Materials under any subsequent license without express written permission from Licensor referring to the previous breach. Any attempt by such Licensee or its Affiliates to acquire a new license to the Licensed Materials will be deemed to be in bad faith and/or fraudulent.

5. No Warranty

THE SOFT MATERIALS AND HARD MATERIALS ARE PROVIDED "AS-IS, WHERE-IS" AND WITH ALL FAULTS. LICENSORS MAKE NO WARRANTIES OF ANY KIND

WHATSOEVER, EXPRESS, IMPLIED, ORAL, WRITTEN OR OTHERWISE, INCLUDING, WITHOUT LIMITATION, WARRANTIES AS TO NON-INFRINGEMENT, TITLE, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE, OR WARRANTIES ARISING BY CUSTOM OR TRADE USAGE. ANY AND ALL INSTANTIATIONS OR COMPILATIONS OF THE SOFT MATERIALS AND HARD MATERIALS, AND THE DISTRIBUTION OF ANY RESULTING HARDWARE, ARE DONE AT LICENSEE'S OWN RISK.

6. Warranty by Licensee

Licensee may choose to provide warranties, representations, and/or incur liabilities in association with distributions of Documentation and/or Hardware in accordance with the license terms above, provided that, in no event may Licensee or its Affiliates bind any Licensor to any warranty, representation, or liability.

7. Limitation of Liability

TO THE FULLEST EXTENT PERMITTED BY APPLICABLE LAW, IN NO EVENT SHALL LICENSORS BE LIABLE UNDER OR IN RELATION TO THIS LICENSE AGREEMENT FOR ANY DAMAGES, INCLUDING ANY SPECIAL, CONSEQUENTIAL, INCIDENTAL, DIRECT, INDIRECT, PUNITIVE OR EXEMPLARY DAMAGES WHATSOEVER, OR LOST PROFITS, HOWEVER CAUSED, EVEN IF LICENSORS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.