CERN OHL version 2
An Introduction and Explanation

CERN

February 14, 2020

1 Introduction

The CERN-OHL has become one of the most respected open hardware licences. People in the world of open hardware tend to associate it particularly with electronic devices designed at circuit-board level (because that was the original use case, and continues to be one of the most common).

In previous iterations, we'd tried to address the licence's applicability to other fields, and at one point during the drafting process for the next version we considered (and drafted) a set of exceptions designed to allow for weak copyleft, the use of proprietary library files for HDL design, and compatibility with GPLv3 and TAPR. From a legal perspective, these seemed to work, but we are conscious that the licence was becoming complex and cumbersome: we felt we needed a different approach. We went back to first principles, deleted much of the text in the existing CERN-OHL, and questioned the licence's fundamentals, dramatically simplified some of the existing wording, and tried to come up with something simpler, clearer and more useful, but at the same time retaining the same spirit (so that people who had chosen earlier versions of the CERN-OHL which allowed for automatic permission to relicense under later versions wouldn't feel that we had misled them).

We had also been asked if a permissive version of the licence could be produced. We looked at this a number of different ways, and even drafted a permissive exception which could be used with version 2, and then incorporated a permissive variant into version 2 itself. This was fairly cumbersome, not least because people looking for a permissive licence tend to be looking for something simple, and it's a rather jarring to have to read and understand a complex reciprocal licence, and *then* to have to read some more legal wording to turn it into a simpler licence.

Finally, we've adopted the following names for the licences: the CERN Open Hardware Licence - Stronger Reciprocal (the successor to the original CERN-OHL), the CERN Open Hardware Licence - Weakly Reciprocal, and a simpler CERN Open Hardware Licence - Permissive. These are known as CERN-OHL-S, CERN-OHL-W and CERN-OHL-P respectively.

This document sets out some of our thinking, which will, we hope, help you to decide whether the CERN-OHL is the right licence for you, and, if so, which variant.

1.1 The Boundaries - Horizontal and Vertical

There were still some niggling concerns driven by frequent questions: if you incorporate a component licensed under a reciprocal version of the CERN-OHL into any piece of hardware, does that mean that the whole of that piece of hardware has to be released under the CERN-OHL? For an electronic design, do you have to release the design of all the constituent components, such as the chips, the capacitors and so on? Is the firmware or microcode contained in SoCs or microprocessors included in information which needs to be provided? We tried to address some of these issues in previous versions of the licence by requiring details of modified designs to be provided 'at a similar level of abstraction' to the original design. This went some way to addressing the issue, but possibly wasn't as clear as it could have been.

We were conscious of the problem of defining the boundaries where the effect of the reciprocal CERN-OHL stopped. There are two issues: vertically (if I place a reciprocal CERN-OHL licensed component in another assembly, does the whole assembly need to be licensed under the equivalent reciprocal CERN-OHL, and, if so, what if that assembly is placed within an even larger item?); and horizontally: if I interface a reciprocal-licensed CERN-OHL licensed circuit board to another circuit board and distribute the two together, do both of them have to be licensed under the reciprocal CERN-OHL?

1.2 Introducing *Available Components*

Then, in the course of radically simplifying the language of the original CERN-OHL to produce the new reciprocal versions of the CERN-OHL, it occurred to us the answer might lie in being more specific about how we tackle components.

Starting with a circuit board as an example, it almost goes without saying that if all the components are common commodity electronic components, then it shouldn't be necessary to provide details any more specific than you'd find, for example, in the project instructions in an electronics magazine. The design might need three BC108 transistors, a 555 timer, and five 220 Ohm resistors. Perhaps, for the circuit in question, the resistors need to be specified as 0.25 watt resistors, and be accurate to 10 percent tolerance. That is sufficient information for a maker to make the board, and instinctively feels like the correct amount of information to provide.

But what if you're making a circuit board, and some of the components aren't commodity components? In this case, to get all the information you need to make the circuit board, you need the board layout, the circuit diagram, a list of components, and, for those materials that aren't commodity materials, you need sufficient information about their specifications, interfaces and characteristics (such as their power consumption, physical package size and thermal profile), to be able to make one, even if you don't have details of the design itself. For example, the design might specify an AC/DC power converter taking in 220V AC and supplying 1.5 amps at 12V DC on its output, fitting within a cube with 70mm edges, and not emitting more than 2W of heat at full load.

Ideally, the full design for the component (the power converter in this example) would be under the CERN-OHL, but it might be available under another open hardware licence, or the design might even be under a proprietary licence, or it might be the case (as with the power converter) that the specifications are sufficient to enable a reasonably competent maker to source or make one.

As long as you have the ability to make or obtain the relevant component, even at a cost (and remember, you'll have to pay for simple components like capacitors or resistors, so why should more complex or unusual components not come at a price or with conditions attached?), then this shouldn't be a barrier to the design being capable of release under the CERN-OHL, and being regarded as an open hardware design.

We call components which are generally and readily available (even if they have to be paid for, or you have to sign up to an NDA to get them), ***Available Components***. When you provide the Complete Source (the design documentation) for your design under the CERN-OHL, you do not have to provide the Complete Source for each of the Available Components (of course, it's great if you can!). The minimum requirement, (for physical components in CERN-OHL-S, and for any component, physical or digital/virtual, in CERN-OHL-W), is to provide enough information to describe them and their interfaces, to enable them to be sourced or made, or you can, of course, provide the Complete Source for the Available Component under the CERN-OHL (W or S) itself. (You'll notice that this allows the nesting of CERN-OHL components within a larger CERN-OHL design: this is intentional. The nesting can work in both directions: downwards in the direction of atoms, and upwards in the direction of galaxies).

'Available Component' works slightly differently depending on whether the variant of CERN-OHL is the 'W' or the 'S' variant. (And this distinction is one of the few differences between the two reciprocal variants of the CERN-OHL). Remember, if a component is included in a reciprocal CERN-OHL design and it is NOT an Available Component, the Complete Source for that component must be made available as part of the Complete Source of the Product.

The definition reads as follows (-S variant):

> *'Available Component'* means any part, sub-assembly, library or code which:
>
> a) is available to You as Complete Source under a Compatible Licence; or
>
> b) is available to You and any other prospective Licensees
>
> > i) as a physical part with sufficient rights and information (including any configuration and programming files and information about its characteristics and interfaces) to enable it either to be Made itself, or to be sourced and used to Make the Product; or
> >
> > ii) as part of the normal distribution of a tool used to design or Make the Product.

And in the -W variant:

> *'Available Component'* means any part, sub-assembly, library or code which:
>
> a) is available to You as Complete Source under a Compatible Licence; or

b) is available to You and any other prospective licensees
    i) with sufficient rights and information (including any configuration and programming files and information about its characteristics and interfaces) to enable it either to be Made itself, or to be sourced and used to Make the Product; or
    ii) as part of the normal distribution of a tool used to design or Make the Product.

The difference between these lies in b(i). In the -W variant, *any* Component, physical or digital, can be an Available Component, provided that it is available with sufficient rights and information etc. In the -S variant, this only applies to physical Components (i.e digital components will have either to be available under a Compatible Licence, or be part of the normal distribution of tools).

It's important to understand that 'Available Component' can include Components both in the physical domain (resistors, capacitors, ICs, circuit boards) and Components in the digital domain (firmware, libraries).

For example, if you're designing a component which can be implemented by FPGA, you may very well be using tools such as those provided by Xilinx, and to make effective use of these, you may also be using accompanying libraries. If you're configuring an FPGA, the output of the tools will be a bitstream (a digital sequence which is loaded into the FPGA and configures it to have the functionality you are seeking).

If you're designing a development board that contains an FPGA which the user can load with their own bitstream, then you are under no obligation to distribute any bitstream at all, if the FPGA is not loaded with one on distribution. You may choose to load a simple sample bitstream, in which case you should distribute either the bitstream itself, alongside the FPGA, or the FPGA preloaded with a bitstream with sufficient description to enable someone to recreate the functionality of the FPGA as configured. The bitstream itself does not have to be licensed under the CERN-OHL (although it may be), but it must be provided with sufficient rights to enable the recipient of the FPGA to use it to Make the Product.

However, we encourage people to release the HDL (hardware description language) file for the bitstream under the CERN-OHL-S, -W or -P. That essentially makes it a Product (the bitstream) within a Product (the FPGA).

Where a bitstream is released under CERN-OHL-W or -S, what's the Complete Source for the bitstream? It will consist of the custom code that you have written together with details of the libraries you are using to create it. If they are the standard libraries that are provided with Xilinx's ISE or Vivado Design Suite, then they will fulfil the definition of Available Component: they are libraries or code generally available (in this case, at a cost), and are part of the normal distribution of a tool used to design or Make the Product. So you get the bitstream you need, and you are not required to release the Xilinx libraries if you distribute the bitstream. This is intended to balance the legitimate interests of the FPGA companies as well as retaining the essential openness in the design.

What if you want to include in the HDL some code from a third party library which is not part of the normal distribution of a tool used to design or make the Product? If it's already licensed under CERN-OHL (any version) or a

Compatible Licence, you're fine: it's automatically an 'Available Component', and you can distribute it as a Product in its own right by providing access to the Complete Source for it (either modified, or not, as you desire, provided you comply with that version of the CERN-OHL on distribution). (A Compatible Licence is a version of the CERN-OHL, or another licence which lets you treat the Source released under it as being released under the -S or -W variants of the CERN-OHL. This would include permissive licences, such as the Apache 2.0 or Solderpad Licences, and also the CERN-OHL-P).

What's the rationale behind the distinction between CERN-OHL-S and CERN-OHL-W? This only affects Products in the non-physical (digital or virtual) domain (like bitstreams) and means that, for the W-variant, the bitstream can contain code licensed under Compatible Licences, libraries and components normally distributed with the tools you are using, and third party libraries, provided they are generally available and you have the interface information available for them. These may be proprietary libraries, or they may be libraries under other licences (such as LGPL) provided that the interface information information is available (and of course, you must comply with the terms of the third party libraries if you are going to distribute them along with the Source).

For CERN-OHL-S, if the Available Components are in the digital domain, you can't use third party libraries unless they are part of the normal distribution of a tool you are using (or are themselves licensed under a Compatible Licence).

If you're designing an ASIC, similar considerations apply, although you will be using different tools and libraries, potentially provided by the foundry and of course, the ASIC will be supplied as a chip with pre-defined functionality. You will either have to make the ASIC itself available together with the information necessary to interface it to the rest of the project (just as if you were providing a 555 timer), or you will have to provide the HDL used to create the ASIC, together with details of the target silicon. An ASIC can be an Available Component if it's available with full specification and can be sourced. It's not necessary to provide the HDL in these circumstances. However, anyone producing an ASIC can choose to release it, and its associated HDL, under the CERN-OHL. In this case, the HDL is likely to consist of custom code, and libraries, primitives, macros etc. Each of these is a component, and to be CERN-OHL compatible, they have to be Available Components. You will recall that this means that they will either be your code, code under an older version CERN-OHL, other code you are able to release under CERN-OHL-S or -W, or code which is shipped as part of the standard components of a tool you are using, or (in the case of CERN-OHL-W only) other third party libraries which you have the right to ship, and which you can provide with appropriate interface information (but for which you do not have to provide Complete Source).

For those of you familiar with free software licensing, the ability for the -W variant to combine with third party libraries is similar to the way the LGPL works: you can interface LGPL code in a larger work without the whole larger work having to be itself released under the LGPL, provided you provide enough information about the LGPL work to enable it to be modified and re-interfaced to the larger work.

1.3 Moving from CERN-OHL-W to CERN-OHL-S

Can I combine a CERN-OHL-W work and a CERN-OHL-S work? If one of the works counts as an Available Component within the definition of the other, then yes, there's no problem.

If, however, you want to merge two designs into one, then you can, but only if the final work is licensed under CERN-OHL-S, and all of the Available Components in the design still qualify as Available Components within the CERN-OHL-S definition. So, for example, if the CERN-OHL-W design consists of, say, HDL, and you have all of the information needed to interface and use library X (in other words, you're relying on the definition of "Available Component" in 1.7(b)(i)), then, because a code library isn't a physical component, it won't qualify as an Available Component in CERN-OHL-S, and you'll be unable to relicense the design under CERN-OHL-S.

1.4 A note on 'available'

We use the term 'available' quite a lot. What does it mean? In our view, a component is available where it can be obtained by anyone who wants to Make the Product, even if to obtain it, they have to pay for it, obtain a licence to it, or even enter into a non-disclosure agreement to get hold of it. So long as the next maker can get hold of the component and is not discriminated against when they are obtaining it, then it's potentially an available component. There may be reasons why people are unable to get hold of a component which are not the fault of the supplier of the component: for example, the component may be subject to export control because it contains technology (such as cryptography) which cannot be exported to certain countries. Our view is that if the restriction is not the choice of the supplier, then the component is still an available component. We have not expanded the definition of 'available' to cover this, because we feel that this is something which will develop over time, and any ambiguity will tend to drive people in the direction of more openness, which we applaud.

In many ways, our approach is more open than the OSHWA certification program, which anticipates that a design can still be certified as open source hardware even if the components it contains are closed, provided that those components aren't under the licensor's control. Under CERN-OHL, closed components can be incorporated only where all the relevant interface information is provided.

In the same way that it's possible for software to be released under an OSI-approved licence and not itself to comply with the open source definition (although this is fairly unusual) it's possible for hardware to be released the CERN OHL and not comply with the OSHWA definition. For example. the OSHWA definition requires that the design documentation is made publicly available, although (see below) in common with many free and open source software licences, the CERN-OHL allows designs to be distributed privately.

1.5 Nesting

Turning back to the definition of 'Available Component', you may ask why we included the wording allowing a component to be described as an Available Component, if it is itself available under the CERN-OHL? Surely, if it's not an

Available Component, it must be available under CERN-OHL anyway, so why allow this option?

The reason is that we regard it as entirely possible for a design to consist of a number of different nested levels (e.g. cabinet, circuit board, electronic components, firmware) all or some of which can be licensed under CERN-OHL (potentially different variants or versions at different levels).

Without this distinction, you could have a circuit board design including the code for an FPGA bitstream at the same level, and all part of the same Product, but it would be untidy and make it difficult, for example, to strip out the irrelevant notices if someone wanted to use the FPGA design separately. It makes it more difficult to indicate to users where they can find the Complete Source and where it has to be indicated on the product. For this reason, it's much neater to allow CERN-OHL designs to incorporate other CERN-OHL designs as components, rather than for them to become one monolithic design.

1.6 Private Distribution

It comes as a surprise to some people, but the GPL family of licences contain no obligation on someone distributing code to make the code public. They do require the distributor to make the source code available to the recipient, and the source itself has to be available under the GPL. Of course, it means that the recipients are able to make the code public if they want to (and it's a breach of the GPL to try to prevent, whether by contract or otherwise, recipients from exercising their rights under the GPL, including this one).

In contrast, earlier versions of the CERN-OHL enforced the idea of requiring a recipient to make the design documentation available to the public if the recipient made changes to the documentation and then distributed either it or a Product Made using the design.

We thought hard about this, and decided, in the new version, to allow private distribution, in the same way that the GPL family of licences do. We retain the idea of a 'Source Location' (which is a publicly accessible place like an online repo) which you can insist is used in the attached Notice file, but if you don't insist on using a Source Location, you or any downstream licensee can privately distribute the code. Of course, any downstream recipient is permitted to use and specify a Source Location if they wish.

1.7 GPLv3 Compatibility

We thought hard about compatibility with other licences. An earlier version of the draft contained a mechanism for options, and a specific option which allowed compatibility with GPLv3. There's an easy way to allow compatibility, of course: you can insert a clause which explicitly allows relicensing, but this necessarily means that one of the licences will have characteristics which are more forgiving than the other, and users can just pick whichever licence they want. If the original licensor wanted to do this in the first place, they could simply dual license under CERN-OHL and GPLv3.

We also found that the mechanism we chose was pretty cumbersome, especially since to make it work, we needed also to add an additional permission to GPLv3. Since we're essentially asking people to relicense their GPLv3 components anyway (albeit within the mechanism), we felt we should be more explicit about this.

Because we've adopted Available Components in the CERN-OHL, and Available Components can be code, such as libraries, this fits most neatly into LGPL compatibility. We'll expand this document over time to explain how this works.

1.8 The licence to the licence(s)

Both versions of the licence state that it may only be used in unmodified form. We wanted to clarify that. Clearly, you can apply the CERN-OHL to your hardware designs, and we encourage you to do so. However, we do not want you to modify the licence itself, as this would confuse people, and may also give people the impression that the licence that you have altered is in some way endorsed by CERN.

On the other hand, we are keen to see the licence disseminated as widely as possible, and that people are not hindered from learning about the licence. For this reason, everyone also has the following additional permissions with respect to the text of the CERN-OHL.

1. You may faithfully translate the licence into any other language, and distribute or make available ("convey") the translation, provided that you add a note to your translation in that language saying that the text is a translation, and is not a legal text, is not approved or endorsed be CERN, and that the only definitive legal text is the version in the English language released by CERN at [CERN URL], and that your translation of the licence may only be further conveyed if the notice you have added is retained in full.

2. You may use, modify and convey extracts from the licence (if a "fair use" or "fair dealing" type exception does not apply already) for the purpose of explanation of the licence, teaching or advising on the effect of the licence, provided that it is clear to a recipient that the extract is only an extract of the licence, that CERN is credited as the source of the licence, and that that extract is not being used to create any legal relationship, rights or obligations.