

Frequently Asked Questions

Version 1.5

Revision History

Revision	Description of Change	Date
v1.1	Initial creation for OpenCPI 1.1	3/2017
v1.2	Updated for OpenCPI Release 1.2	8/2017
v1.4	Updated for OpenCPI Release 1.4	9/2018
v1.5	Updated for OpenCPI Release 1.5	4/2019

1 General Questions

Is the RPM suite a standalone install?

Yes, the RPMs distributed by the ANGRYVIPER team incorporates and extends the Free / Open Source Project “OpenCPI.” Any OpenCPI installation documents that still exist are for reference and legacy users. All other OpenCPI documentation still applies and should be referenced. Do **not** attempt to install OpenCPI from source at the same time as the RPM distribution.

Where can I go for more help?

All documentation is available at github.io and there is a public mailing list (with archive) at opencpi.org.

2 Install-Specific Questions

Does it matter what version of CentOS is used?

Both CentOS 6¹ and CentOS 7 are supported as long as the proper version of the RPM is used. Local hardware support (*e.g.* PCIe-based platforms) is officially supported on both OS releases starting with Version 1.1. CentOS 6, while not officially deprecated, does not get as much testing by the core team, so may have unnoticed issues.

CentOS 7 is now considered a “rolling release” so they say, for example, that at a point in time² it is “CentOS 7 (1810)” but *not* specifically “7.6”. The OpenCPI RPMs are built using “7 (1804)” which is informally 7.5, so if you are running a version older than 7.5, you need to upgrade the target OS or use the source distribution.

3 General Usage Problems / Questions

Make error: “*** isim not an available HDL platform. Stop.”

Either the Core Project was never built, or it is not properly registered. This is explained in the *Getting Started Guide*.

```
/opt/opencpi/cdk/centos7/bin/ocpigen: /opt/Xilinx/.../libstdc++.so.6: version 'GLIBCXX.3.4.15' not
found (required by /opt/opencpi/cdk/centos7/bin/ocpigen)
```

If the OpenCPI tools, *e.g.* `ocpigen`, are reporting problems with a `libstdc++` *within a Xilinx tool path*, that means your environment has imported the Xilinx tool’s configuration script, *e.g.* `settings64.sh` (Section 4). If you didn’t manually import it, ensure you don’t have a line that does elsewhere, *e.g.* in `~/.bashrc`.

I am trying to run a demo application with “ocpirun” and artifacts are not being found.

The usual causes of this are:

- Core Project was not built for the target platform
 - Consult the *Getting Started Guide*
- `OCPI_LIBRARY_PATH` was not properly set
 - View the artifacts being checked by adding “-1 8” on the `ocpirun` command line to increase the logging level

HDL Workers are failing Unit Tests that passed before 1.4.

The most likely cause is that “backpressure” is now automatically asserted by default; see the *Component Development Guide* for details.

¹CentOS 6 RPMs are built using 6.10.

²<https://wiki.centos.org/About/Product> and https://wiki.centos.org/FAQ/General#How_does_CentOS_versioning_work?

My application's I and Q seem wrong after moving to 1.4.

See below and the *Release Notes*.

How do I handle iqstream_protocol's I and Q data ordering in HDL workers?

As noted in the *HDL Development Guide*, when a Protocol contains a Struct Argument, the first Argument Member defined in the Protocol's XML is *always* in the *least significant* bits of the resulting Port. A good example of the importance of this is the scenario where a Port's default data width is overridden (in the OWD) to present all of a Struct's Members in parallel within a single clock cycle. For example, `iqstream_protocol`'s default ordering is a 16-bit interleaved I/Q data: " $I_{t=0}, Q_{t=0}, I_{t=1}, Q_{t=1}, I_{t=2}, \dots$ ". However, if the Worker configures its data Ports to be a width of 32 bits, then the I/Q data is presented as a parallel I/Q sample pair with "I" in the *lower* 16 bits and "Q" in the *upper* 16 bits, *i.e.*:

$$\begin{aligned} I_0 &= InPort_0[15 : 0]; Q_0 = InPort_0[31 : 16] \\ I_1 &= InPort_1[15 : 0]; Q_1 = InPort_1[31 : 16] \end{aligned}$$

A Worker to consult as an example is "`iqstream_max_calculator.hdl`."

4 Xilinx-Specific Questions

Are there any other setups I need to perform on the Xilinx Vivado or ISE side?

No, we abstract away a lot of the requirements if you simply install it in `/opt/Xilinx` and point the OpenCPI-specific setup variables to it (see `/opt/opencpi/cdk/env.d/xilinx.sh.example` and the *RPM Installation Guide*).

Additionally, importing the Xilinx setup scripts, *e.g.* "`source/opt/Xilinx/14.7/ISE_DS/settings64.sh`" or "`source/opt/Xilinx/Vivado/2017.1/settings64.sh`", can cause other problems (Section 3) and **should not be performed**.

The ZedBoard comes with a license, but it is for the Vivado tools.

Xilinx's "WebPack" Vivado or ISE license is sufficient to do anything with the ZedBoard.

ISE Note: As for purchasing, you can "rollback" a Vivado license by contacting Xilinx and they will issue you an ISE license with the same expiration with a gentleman's agreement that you won't use both at the same time.