

OpenCPI

Release Notes

OpenCPI Release: v2.4.6

Revision History

Revision	Description of Change	Date
v1.0.1	Initial creation	05/2016
v1.1.0	Updates for 1.1 release	03/2017
v1.2.0	Updates for 1.2 release	08/2017
v1.3.0	Updates for 1.3 release	01/2018
v1.3.1	Updates for 1.3.1 release	03/2018
v1.4.0	Updates for 1.4.0 release and moved previous to appendices	10/2018
v1.5.0	Updates for 1.5.0 release	04/2019
v1.6.0	Updates for 1.6.0 release	01/2020
v1.7.0	Updates for 1.7.0 release	06/2020
v2.0.0	Updates for 2.0.0 release	10/2020
v2.1.0	Updates for 2.1.0 release	03/2021
v2.2.0	Updates for 2.2.0 release	07/2021
v2.3.0	Updates for 2.3.0 release	09/2021
v2.4.0	Updates for 2.4.0 release	10/2021

Table of Contents

I Enhancements	1
I-1 Training Improvements	1
I-2 Instructional Video Links Added	1
I-3 Air-gapped Installation Section Added	1
I-4 New <code>triggered_sampler</code> Components	1
II Changes and Notes	1
II-1 <code>complex_mixer</code> Component Moved	1
II-2 Updates to Runtime C++ Files	1
II-3 GUI Changes	2
IIIMajor Known Deficiencies	2
III-1 Previously Documented Deficiencies	2
Appendices	3
A Release 2.3.0	3
A-I Enhancements	3
A-I.1 PicoEVB HDL Platform	3
A-I.2 <code>test_internal_assy</code> Assembly	3
A-I.3 New <code>analog</code> OSP	3
A-I.4 New <code>sdr</code> COMP	3
A-I.5 <code>capture_v2</code> Unit Test	4
A-II Changes and Notes	4
A-II.1 <code>ocpidev</code>	4
A-II.2 <code>ocpidoc</code>	5
A-II.3 GUI Changes	5
A-IIIMajor Known Deficiencies	5
A-III.1 Previously Documented Deficiencies	5
B Release 2.2.0	5
B-I Enhancements	5
B-I.1 Xilinx Vivado 2020.2 Support	5
B-I.2 Ubuntu 20.04	5
B-I.3 “makeless” Development	5
B-I.4 Inferred Project Dependencies	6
B-I.5 “On Demand” Worker Configuration/Building	6
B-I.6 Multiple Device Instances	7
B-I.7 <code>zcu104</code> Regression Fixed	7
B-I.8 New Documentation Generator	7
B-I.9 <code>RCCResult</code> Return Values	7
B-I.10 <code>ettus</code> OSP	7
B-I.11 New GUI Beta	7
B-I.12 <code>RCC</code> Proxy Workers	7
B-I.13 <code>zcu106</code> Platform Support	7
B-I.14 <code>complex_short_time_sample</code> (CSTS) Workers/Apps	8
B-II Changes and Notes	8
B-II.1 <code>ocpidev</code>	8
B-II.2 Decoupling of <code>OCPLCDK_DIR</code> and <code>OCPLROOT_DIR</code> Environment Variables	8
B-II.3 <code>data_sink_qdac</code> and <code>data_src_qdac</code> Workers	8
B-II.4 Python ACI Update	8
B-IIIMajor Known Deficiencies	8
B-III.1 Previously Documented Deficiencies	8

C	Release 2.1.0	9
C-I	Enhancements	9
C-I.1	GNU Radio Installation Support	9
C-I.2	ADI mykonos Library	9
C-I.3	FSK Application on Matchstiq Z1	9
C-I.4	ComplexShortWithMetadata-prot.xml Sequence Length	9
C-I.5	Digital Radio Controller	9
C-I.6	Proxy Workers and Slaves Model	9
C-I.7	Proxy Worker Capabilities	9
C-I.8	Scalable Software-side Buffer Count Without Memory Bloat	10
C-I.9	Timed Transmission	10
C-I.10	Performance Improvements	10
C-I.11	OCPI_ROOT_DIR Environment Variable	10
C-I.12	e3xx OSP	10
C-I.13	timestamperscdcd-spec.xml Properties	11
C-II	Changes and Notes	11
C-II.1	ocpigen	11
C-II.2	ocpiremote Option Standardization	11
C-II.3	platform_pkg Updates	11
C-III	Major Known Deficiencies	11
C-III.1	Previously Documented Deficiencies	11
D	Release 2.0.0	11
D-I	Enhancements	11
D-I.1	Deprecations	11
D-I.2	Binary Distribution Builds	11
D-I.3	CentOS 8	12
D-I.4	Digital Radio Controller for E31x	12
D-I.5	Man Pages	12
D-I.6	ocpiadmin	12
D-I.7	Optional Slave Support	12
D-I.8	Proxy Slaves	12
D-I.9	Selection Expressions	13
D-I.10	Ubuntu 18.04	13
D-I.11	Xilinx RCC Platforms	13
D-II	Changes and Notes	13
D-II.1	Component Library Searching Algorithm	13
D-II.2	ocpigen	13
D-II.3	PlutoSDR RCC Platform Support	13
D-II.4	Primitive Libraries util and bsv	13
D-II.5	Test Bitstream testbias	13
D-II.6	Tutorials	13
D-III	Major Known Deficiencies	14
D-III.1	Previously Documented Deficiencies	14
E	Release 1.7.0	14
E-I	Enhancements	14
E-I.1	ADC/DAC Clocking Architecture	14
E-I.2	Built-in platform Project	14
E-I.3	CIC Decimator RCC work-a-like	14
E-I.4	Clock Generator Primitive	14
E-I.5	Man Pages	14
E-I.6	ocpiremote	14
E-I.7	PlutoSDR HDL Platform Support	15
E-I.8	Protocol complex_short_samples	15
E-I.9	Ubuntu 16.04 Support	15
E-I.10	zcu104 HDL Platform Support	15

E-II	Changes and Notes	15
E-II.1	<code>ComplexShortWithMetadata</code> Changes	15
E-II.2	MacOS 10.14 and 10.15	15
E-II.3	Optional HDL Ports with <code>clockdirection=in</code>	15
E-II.4	Python Scripts	15
E-II.5	SCons 3	16
E-II.6	SD Card Export Model	16
E-III	Major Known Deficiencies	16
E-III.1	HDL Platform <code>zed_lise</code> Build Error	16
E-III.2	Remote Development using Network Mode Scripts	16
E-III.3	RPM Support	16
E-III.4	Previously Documented Deficiencies	16
F	Release 1.6.0	16
F-I	Enhancements	16
F-I.1	Support for Workers with Separate Control and Data Clocks	16
F-I.2	Issues Database for OpenCPI is now Publicly Accessible	17
F-I.3	Source Installation Simplified	17
F-I.4	Time Service Interfaces were Upgraded for better Timestamping Support	17
F-I.5	New Features for Simplifying ADC/DAC Devices Workers and Enabling Timestamping	17
F-II	Changes and Notes	17
F-II.1	Repository of OpenCPI moved to GitLab.com	17
F-II.2	Project with Content for Tutorials is in the Core Repository	17
F-II.3	MacOS Catalina (10.15) is now Supported for Software Only (no FPGA) Development.	17
F-II.4	The <code>gpsd</code> package is now a prerequisite for all OpenCPI systems.	17
F-II.5	Optional Environment Settings for a Source Installation are in a New File.	18
F-III	Major Known Deficiencies	18
F-III.1	RPM Installation is not Supported in Release 1.6.0	18
F-III.2	Previously Documented Deficiencies	18
G	Release 1.5.0	19
G-I	Enhancements	19
G-I.1	New Digital Radio Controller Front-End Control Paradigm	19
G-I.2	New Time-Based Protocol (" <code>ComplexShortWithMetadata-prot</code> ") (Technology Preview)	19
G-I.3	New Time-Based Assets Project (" <code>assets_ts</code> ") (Technology Preview)	19
G-I.4	New HDL Worker Port Interfaces	19
G-I.5	<code>ocpidev</code> Improvements	19
G-I.6	New "Zed" RCC Platform: <code>xilinx13.4</code>	19
G-I.7	<code>gdb</code> and <code>rsync</code> Provided on <code>xilinx13.3</code> and <code>xilinx13.4</code> RCC Platforms	20
G-II	Changes and Notes	20
G-II.1	Deprecation of the "Readable" Attribute	20
G-II.2	Deprecation of Zipper RF Interface Board	20
G-II.3	E3xx PPS Discipline	20
G-II.4	" <code>new_project_source</code> " script removed	20
G-II.5	RCC Worker Parameter Macros Stricter	20
G-II.6	<code>ocpidev</code> Changes	20
G-II.7	" <code>myClock</code> " HDL Code Deprecated / Removed	20
G-II.8	Infrastructure OWDs Moved	20
G-II.9	RPM Upgrade Notes	21
G-II.10	HDL Wrapper Changes	21
G-III	Major Known Deficiencies	21
G-III.1	" <code>udev</code> " Radio Rules Sometimes Does Not Work	21
G-III.2	Emulators Not Constrained To Specific Containers In Unit Tests	21
G-III.3	Provided LiquidDSP Build May Link Incorrectly	21
G-III.4	Previously Documented Deficiencies	21

H	Release 1.4.0	23
H-I	Enhancements	23
H-I.1	Support for Ettus E310 Platform	23
H-I.2	Component Unit Test Enhancements	23
H-I.3	Multi-Slave Proxy Support	23
H-I.4	Transmit Blanking / Powerdown for DACs	23
H-I.5	ocpidev Improvements	23
H-I.6	VHDL-Native Pattern and Capture HDL Workers	24
H-I.7	New Documentation RPM	24
H-I.8	Zynq Parts Defined	24
H-II	Changes and Notes	24
H-II.1	I/Q Data Ordering	24
H-II.2	Upgrade Notes	25
H-II.3	Significant SD Card Layout Changes	25
H-II.4	Intel Quartus and Quartus Pro Version Change	25
H-II.5	New Input Port ZLM-Handling Primitive	25
H-II.6	Change in QDAC/QDAC_TS Component Spec	25
H-II.7	Change in ocpidev Default Output and Scope	25
H-II.8	Rounding and Truncation in qdac-spec Workers	25
H-II.9	HDL Workers Failing Unit Tests That Previously Passed	26
H-II.10	New Framework Build System	26
H-II.11	New RCC Platform Configuration	26
H-III	Major Known Deficiencies	26
H-III.1	ZedBoard IP Required Even In Standalone Mode	26
H-III.2	Connection Direction In GUI Assumes XML order	26
H-III.3	OnlyPlatforms/ExcludePlatforms Does Not Work For Individual Unit Test Cases	26
H-III.4	Some HDL Workers Will Need To Be Updated When props_in.raw.address Width Fixed	26
H-III.5	Default python Must Be Python2	26
H-III.6	Some ocpi.assets Workers Have Flow Control Issues	26
H-III.7	Previously Documented Deficiencies	27
I	Release 1.3.1	28
I-I	Enhancements	28
I-II	Changes and Notes	28
I-III	Major Known Deficiencies	28
I-III.1	PCI Platforms Can Only Transmit Using One Channel	28
J	Release 1.3	29
J-I	Enhancements	29
J-I.1	OSS Release	29
J-I.2	Python and GNU Radio Integration (Technology Preview)	29
J-I.3	AD9361 Support	29
J-I.4	GUI Improvements	29
J-I.5	Project Registry	30
J-I.6	Project Migration	30
J-I.7	ocpidev Improvements	30
J-I.8	RCC Platforms	30
J-I.9	Remote Containers (Technology Preview)	30
J-I.10	OpenCL Containers (Technology Preview)	30
J-II	Changes and Notes	30
J-II.1	Major Project Restructuring	30
J-II.2	Upgrade Notes	31
J-II.3	Prerequisite RPMs Built With Newer Language Standards	31
J-II.4	Legacy Platform Names Continue to be Deprecated	31
J-II.5	OCPLXILINX_VIVADO_DIR Variable Changed	31
J-II.6	ocpidev and Requesting RCC Platform To Match HDL Platform	31
J-II.7	zero_padding Component in ocpi.assets deprecated	31

J-III Major Known Deficiencies	31
J-III.1 Previously Documented Deficiencies	31
K Release 1.2	33
K-I Enhancements	33
K-I.1 Vivado Support for Applicable Xilinx-Based Platforms	33
K-I.2 New command option “ocpidev build”	33
K-I.3 GUI Now Delivered as RPM	33
K-I.4 Artifact Version Checking	33
K-I.5 Miscellaneous Improvements	33
K-II Changes and Notes	33
K-II.1 rx-spec and tx-spec Support Property Interdependence	33
K-II.2 Vivado Support	34
K-III Major Known Deficiencies	34
K-III.1 Driver Failing to Allocate Memory on Zynq-Based Platforms	34
K-III.2 HDL Workers Cannot Send SOM Without Valid Data	34
K-III.3 SDP Cannot Handle More Than One ZLM Per Application Run	34
K-III.4 File_Read and File_Write Cannot Handle ZLM Mid-file	34
K-III.5 PCI-Based Platforms Not Always Detected in Remote Tests	34
K-III.6 Built Projects Cannot Be Moved	34
K-III.7 Properties Before Raw Properties Cannot Have Parameterized Lengths	34
K-III.8 XML Parser Mixed-Case Issue	34
K-III.9 Simulation of Primitive Cores Needs Makefile Modified	34
K-III.10 Lime RX/TX Proxy Tests on CentOS6	35
K-III.11 Previously Documented Deficiencies	35
K-IV Addenda to Previous Release Notes	35
L Release 1.1	36
L-I Enhancements	36
L-I.1 Debug Symbols for Framework Now Available	36
L-I.2 Kernel Driver Memory Arguments	36
L-I.3 HDL File_Read and File_Write CWD	36
L-II Changes and Notes	36
L-II.1 Simulators Renamed and Work Out-of-the-Box	37
L-II.2 HDL Width Adapters Not Required	37
L-II.3 Platform Name Changes	37
L-II.4 RCC Cross-compilation No Longer Uses OCPI_CROSS_HOST	37
L-II.5 Updating a 1.0.x Unit Test for the 1.1 Framework	37
L-II.6 Prerequisite RPM Changes	38
L-II.7 Deprecated RCC controlOperations	38
L-II.8 64-Bit Build Host Issues (“Value too large for defined data type”)	38
L-III Major Known Deficiencies	38
L-III.1 Maximum Message Length in PCI-Based Platforms	38
L-III.2 MAC Address on ZedBoard when Using the xilinx13.3 Software Platform (use xilinx13.4 to avoid the issue)	39
L-III.3 Platform Development Guide	39
M Release 1.0.1	40
M-I OpenCPI HDL Assembly Filename Constraint	40
M-II 32-bit Architecture File Size Limit	40
M-III Eclipse Project Description File	40
M-IV Refreshing the AV GUI	40
M-V Non-default Xilinx Installation Locations	40
M-VI Zero-Length Message Finite State Machines	40

Overview

This document contains corrections, changes, and enhancements made to OpenCPI since the previous release. It can never encapsulate *every* change or bug fix, but attempts to assist previous users in getting up and running as swiftly as possible. The *general* format of this document is to answer user questions:

- Section I (“Enhancements”) describes “*What’s New?*” about the current release.
- Section II (“Changes and Notes”) is “*What do I need to be aware of as a **previous** user of the Framework?*”
- Section III (“Major Known Deficiencies”) is for *all* users to answer “*What bugs should I be aware of?*”

The Appendices (beginning on page 3) contain the Release Notes from previously released versions of the Framework, roughly following the same format, e.g. J-I, being Section I of the v1.3 Appendix, is the “Enhancements” that were added to Release 1.3.

I Enhancements

I-1 Training Improvements

This release provides updates to Tutorials 1 through 4 to demonstrate the use of the OpenCPI GUI and to bring the tutorials up to OpenCPI “best practices”. This release also adds several new tutorials, both “software only” (running on the xsim simulator) and “hardware” (running on the plutosdr platform). The OpenCPI briefings have also been updated to OpenCPI best practices and two additional briefings have been added (one for an overview of the Application Control Interface, and one to introduce tools and techniques for debugging).

I-2 Instructional Video Links Added

Links to instructional OpenCPI videos on YouTube have been added to opencpi.gitlab.io web page.

I-3 Air-gapped Installation Section Added

The OpenCPI Installation Guide provides instructions for performing an air-gapped installation.

I-4 New `triggered_sampler` Components

Adds new `triggered_sampler` components allowing for applications to gate a settable number samples upon a trigger event, which can be either a write to a property, input data on the optional input port, or a metadata message on the primary input. The following protocols are supported: `bool`, `char`, `float`, `long`, `short`, `uchar`, `ulong`, `ushort`, `complex char`, and `complex short`.

II Changes and Notes

II-1 `complex_mixer` Component Moved

Applications in the tutorial project now utilize the `complex_mixer` component in “`opencpi/projects/assets/components/dsp_comps`”.

II-2 Updates to Runtime C++ Files

Geared towards developers moreso than users, there have been updates made to the `c++` runtime code base. The focus of these changes is reorganization for the sake of consistency in file and namespace naming conventions.

II-3 GUI Changes

The following changes have been made to the new GUI since the previous release:

- The user now has the ability to run a single test case when running unit tests.
- When using `ocpiadmin install` users are now able to specify the minimal install.
- The user now has the ability to run the install script for the GUI and it will create an `ocpigui` command for the user to use to launch the GUI.
- With the `ocpigui` command, users can now use `--project-path` to specify the project path for the project explorer to use and `--ocpi-path` to specify the path to the built `opencpi` project.

III Major Known Deficiencies

III-1 Previously Documented Deficiencies

The following deficiencies are still present in 2.4:

- 32-bit Architecture File Size Limit (cf. M-II)
- OpenCPI HDL Assembly Filename Constraint (cf. M-I)
- Maximum Message Length in PCI-Based Platforms (cf. L-III.1)
- PCI-Based Platforms Not Always Detected in Remote Tests (cf. K-III.5)
- PCI Platforms Can Only Transmit Using One Channel (cf. I-III.1)
- Connection Direction In GUI Assumes XML order (cf. H-III.2)

Appendices

The Release Notes from previous releases are included here as reference for still-existing bugs, relevant upgrading notes for users of very old versions, etc.

A Release 2.3.0

A-I Enhancements

A-I.1 PicoEVB HDL Platform

This release adds preliminary (as in “beta” – as in “**not everything works**”) support for the “PicoEVB” HDL platform, which is a PCIe-based FPGA device. Significantly, this is the first PCI device that OpenCPI has supported in many years, and it is the first to use the modern dataplane infrastructure (SDP) with PCI Express. It can be used as an example for supporting future PCIe-based FPGA platforms.

The “PicoEVB” is a low-cost FPGA device from RHS Research™. The board is designed around the Xilinx Artix 7 (XC7A50T) FPGA with a PCIe x1 interface. The “PicoEVB” will work in the following slots:

- M.2 2230 Key A
- M.2 2230 Key E
- M.2 2280 Key M
- Full length mini PCIe (with carrier board)
- PCIe x1, x4, x8, or x16 slot (with carrier board)

More information about the “PicoEVB” platform can be found at the RHS Research website (<https://rhsresearch.com/>). Note the OpenCPI HDL (FPGA) platform for the “PicoEVB” is called “picoevb”. Additional details may be found in the “PicoEVB Getting Started Guide”.

The following list highlights the features included with support of the “PicoEVB”:

- Updated interface for PCIe-based Control and Data Planes with SDP support (replaces uNOC implementation)
- Added method for automatic generation of Xilinx PCIe IP as part of the build process
- Added assembly for testing HDL platforms without embedded processors

A-I.2 test_internal_assy Assembly

The new `test_internal_assy` assembly is an example of an HDL assembly that generates and receives data internally. It may be used for testing and validating new HDL platforms.

A-I.3 New analog OSP

The new “analog” OSP provides OpenCPI support for Analog Devices, Inc. products. This release provides initial OpenCPI support of ADRV9361-Z7035 attached to ADRV1CRR-BOB.

A-I.4 New sdr COMP

This is the initial OpenCPI release of the contributed “`ocpi.comp.sdr`” project (used to be known as “`ocpi.sdr`”).

In the months since the project was originally created/contributed, the following enhancements, bugfixes, and features have been added and are present in this release:

- Golay Encoder and Decoder components have been added that can be used for forward error correction.
- Convolutional interleaver and convolutional de-interleaver components have been added which can be used to reduce burst errors when transmitting bursts of data.
- The deprecated readable attribute has been removed from the value property for the `equal_to_ulb` component.

- A limit on the number of messages used at the verification stage of unit testing for generator components has been implemented by truncating the generated output files such that they contain no more than a set number of messages.
- The `cic_interpolator_xs` HDL worker has been modified to allow for the `up_sample_factor` property to be set during run time without the worker entering a broken state.
- All primitives and components have been updated to protocol version 2.
- Added more support and standardized the testing for components with multiple output ports.
- Improved the HDL resource efficiency of `fir_filter_scaled_xs` and `fir_filter_scaled_s` by using the `number_of_taps` property to set the array length of taps at build time.
- Improved the HDL resource efficiency of `symbol_mapper_uc_l` by using the `array_size` property to set the array length of symbols (previously named values) at build time.
- Improved performance of RCC worker `counter_up_down_b_uc` (previously `counter_b_uc`) by updating the volatile property `counter_value` (previously `count`) after processing each message instead of each sample in the RCC worker.
- Added checking of description tag text for both child elements and attributes in xml files to `ocpi_linter`.
- Renamed the project package name from `sdr` to `comp.sdr` and updated all `example_app.xml` files accordingly.
- Renamed `counter_b_uc` to `counter_up_down_b_uc`.
- Modified HDL opcode and standard logic vector conversion functions to use `'pos` and `'val` attributes instead of case statements.
- Modified all HDL worker signal names to lowercase.
- Added `bias_xs` component with RCC and HDL workers.
- Added `carrier_generator_s` component with RCC and HDL workers.
- Added `cic_decimator_xs` RCC worker.
- Added `constant_l` component with RCC and HDL workers.
- Added `constant_ul` component with RCC and HDL workers.
- Added `counter_up_down_b_ul` component with RCC and HDL workers.
- Added `counter_up_resettable_b_ul` component with RCC and HDL workers.
- Added `fir_filter_scaled_s` RCC worker.
- Added `fir_filter_scaled_xs` RCC worker.
- Added `multiplexer_l` component with RCC and HDL workers.
- Added `multiplexer_ul` component with RCC and HDL workers.
- Added `ramp_generator_l` component with RCC and HDL workers.
- Added `register_b` component with RCC and HDL workers.
- Added `register_l` component with HDL worker.
- Added `register_ul` component with HDL worker.

A-I.5 capture_v2 Unit Test

A new script has been created that generates an output file for the `capture_v2` unit test's properties in a more readable format. The input to the script is a property dump file when using the `dumpFile` option when running an app, or a `.log` file generated by a unit test. It is located in `"opencpi/projects/assets/scripts/format_capture_v2_property_dump.py"`.

A-II Changes and Notes

A-II.1 ocpidev

The `"-t"` option is no longer available. Test assets cannot be created simultaneously with component assets.

A-II.2 ocpidoc

Although a great deal of progress has been made on finalizing the new documentation generator for OpenCPI assets, it should still be considered as “preliminary” in this release.

A-II.3 GUI Changes

The following changes have been made to the new GUI since the previous release:

- When building OpenCPI assets or running unit tests, RCC and HDL targets can now have multiple platforms selected for a single operation.
- The GUI’s “right click” menu is now context sensitive with respect to the OpenCPI asset being clicked on, and will only show options/operations relevant to that asset.

A-III Major Known Deficiencies

A-III.1 Previously Documented Deficiencies

The following deficiencies are still present in 2.3:

- 32-bit Architecture File Size Limit (cf. M-II)
- OpenCPI HDL Assembly Filename Constraint (cf. M-I)
- Maximum Message Length in PCI-Based Platforms (cf. L-III.1)
- PCI-Based Platforms Not Always Detected in Remote Tests (cf. K-III.5)
- PCI Platforms Can Only Transmit Using One Channel (cf. I-III.1)
- Connection Direction In GUI Assumes XML order (cf. H-III.2)

B Release 2.2.0

B-I Enhancements

B-I.1 Xilinx Vivado 2020.2 Support

Xilinx Vivado 2020.2 is supported as of this release on all OpenCPI development platforms.

B-I.2 Ubuntu 20.04

Ubuntu 20.04 is now a supported OpenCPI development platform. N.B.: The minimum Xilinx tool version that will work with this platform is 2020.2.

With the introduction of “ubuntu20_04” as a supported OpenCPI development platform, the “ubuntu16_04” platform is officially deprecated as of this release. Ubuntu 16.04 is no longer supported by Canonical (no updates available), but we will continue to support OpenCPI on Ubuntu 16.04 until the next major release (3.0.0).

B-I.3 “makeless” Development

Over the last year or so we have been moving toward making OpenCPI “makeless”. “Makeless” means we have been slowly removing “make” and “make files” from the developer experience while allowing previous Makefiles and make commands to continue working.

The motivation for this is twofold:

- Reduce learning required of users (no need to know about “make”) in particular consideration of the error-prone nature of make where spelling errors in variable names are not flagged. Basically it is too easy to make mistakes, hard to help users when they make them, and “make” represents an added learning burden to use OpenCPI.

- Enable us to modernize and optimize the OpenCPI “build engine”, with the freedom to move the inner machinery away from using “make” without impacting users. There are major opportunities to improve the efficiency and parallelism of the build process that are easier to exploit if we are not tied to the “make” infrastructure. While some of these optimizations are possible in the “make” environment, they are brittle and hard to maintain, and require advanced knowledge of “make” that is hard to find.

This has been accomplished by moving what was previously variable settings in Makefiles into additional top-level attributes in XML files. In most cases there was already an XML file in the relevant directory so this eliminated the Makefile, and added attributes to the XML file.

So where there are already XML files for assets (workers, tests, assemblies, applications, platforms), there is a migration of Makefile variable settings into top-level attributes of the existing XML files. All documented “make” variable settings are now XML attributes.

Where there were *only* Makefiles, they have been replaced with XML files, with attributes corresponding to the previously documented Makefile variable settings.

So for all directories where there were previously only Makefiles, there are now XML files. Also, all *.mk files in projects are superceded by the contents of the XML files.

The names of the XML files are the name of the directory minus any suffix. The exceptions to this simple rule are:

- test directories have <name>-test.xml as they always did.
- project directories have Project.xml files, and no Project.mk or Makefiles

New XML files are for these asset types:

- projects
- HDL primitives
- component libraries

New XML files also replace Makefiles in the “plural” directories:

- components (when there are libraries under components/)
- hdl/primitives
- hdl/platforms
- applications

One asset type still has a Makefile and a <platform>.mk file: RCC platforms.

The builtin projects of OpenCPI are now a mix of “makeless”, to test the new scheme, and “legacy” to ensure that old Makefiles still work.

In the 3.0 release, support for all the Makefiles may be removed.

There are a number of cases where Makefiles are still required to do complex things, but in the future we may use a different approach to “customizing the build process” and remove “make” entirely.

B-I.4 Inferred Project Dependencies

A project with hdl assemblies can now be built for a target platform without knowing the project dependencies of the project where the target platform lives. This was always the intent, but was not previously enabled/implemented.

B-I.5 “On Demand” Worker Configuration/Building

When building HDL assemblies, containers, and platform configurations, workers (and parameter configurations of workers) can be built on demand, as needed at the time the assemblies (of the three types) are built. Building HDL primitives and HDL platforms is still a prerequisite to building these assemblies. At installation time, a “--minimal”

option can be supplied to “`ocpiadmin install platform`” which will only build the minimum: primitives and platforms. Then, when building assemblies or platform configurations or even unit tests, a “`--workers-as-needed`” option can be supplied to the “`ocpidev build`” command to automatically build the required workers of the required configurations. Note that a worker configuration does not have to be predefined to be built. The worker configurations in “`*-build.xml`” files are primarily required to define what will be unit tested.

B-I.6 Multiple Device Instances

This release contains new support for platform developers with platforms that have multiple instances of the same device, and enables applications and slave assemblies to specify the exact device when instantiating a device worker.

B-I.7 zcu104 Regression Fixed

The `zcu104` regression in v2.1.0 and v2.1.1 that required the workaround of setting `OCPI_DMA_CACHE_MODE` to zero on the embedded system has been fixed, i.e., as of this release, the workaround is no longer necessary.

B-I.8 New Documentation Generator

This release marks the initial announcement of the new `ocpidoc` tool.

`ocpidoc` is a documentation generator for OpenCPI assets. Writing documentation for OpenCPI assets (e.g., components) can be burdensome, with common information required in some cases. To reduce this burden, a standard way of writing asset documentation is included so that:

- the required information is captured; and
- for parts that can be automatically documented, information is included from, e.g., OpenCPI component specification XML files.

Note this is a preliminary implementation of the document generation capability which will be finalized in the upcoming v2.3.0 release. Also, the use of the `ocpidoc` tool will be changed so that, when necessary, it runs underneath `ocpidev` and will not be directly invoked by users.

The `Sphinx` doc tools version required by `ocpidoc` is more recent than provided in the package repositories of any current OpenCPI-supported development platform. This dependency is handled in a distro-independent manner that is sandboxed from, and mostly invisible to, users.

B-I.9 RCCResult Return Values

The new-in-this-release `RCCResult` return values `RCC_FINISHED` and `RCC_ADVANCE_FINISHED` will replace `RCC_DONE` and `RCC_ADVANCE_DONE` in the next major release (3.0.0), i.e., `RCC_DONE` and `RCC_ADVANCE_DONE` are deprecated as of this release.

B-I.10 ettus OSP

The `ettus` platform project (`opencpi/osp/ocpi.osp.ettus`) is new in this release, and provides preliminary support for the Ettus N310 v4.0.0 RCC platform.

B-I.11 New GUI Beta

A preliminary version of a new GUI for OpenCPI development is officially available as of this release. The associated project is `opencpi/ie-gui`. Feedback is definitely encouraged.

B-I.12 RCC Proxy Workers

RCC proxy workers can now have HDL platforms as explicit slaves in case the proxy needs to have access to aspects of the HDL platform.

B-I.13 zcu106 Platform Support

This release contains preliminary support for the `zcu106` (Zynq Ultrascale ZCU106 Xilinx Development Board) platform.

B-I.14 `complex_short_time_sample` (CSTS) Workers/Apps

Initial support for new CSTS workers and applications is present in this release. Eventually, new CSTS versions of “ComplexShortWithMetadata” (CSWM) workers and applications will be available. As they become available, using them is as simple as specifying “_csts” in the names of the workers/applications – they will use the new protocol automatically. Otherwise identical names without “_csts” are the CSWM versions.

In some future release, there will be a switch to use CSTS by default, meaning the CSWM worker versions will be moved to the “inactive” project and a “_cswm” naming convention added. At the same time, the “_csts” naming convention will be removed, i.e., the CSTS versions will be the default.

At present, it is intended the CSWM versions of workers will continue to be available for use, but the CSWM versions of applications may not.

B-II Changes and Notes

B-II.1 `ocpidev`

The `ocpidev` program is in the process of being completely rewritten in Python (for reasons of maintainability), and the rewrite is mostly complete with this release. The associated changes are intended to be transparent to the user, i.e., no changes in functionality or usage due to the rewrite.

With respect to other `ocpidev` changes affecting the unit test CLI, note the following:

- “`ocpidev build`” is used for generating and building.
- “`ocpidev run`” is used for executing and verifying.
- The “mode” stuff is gone, and “`ocpidev run`” does not build tests anymore.

B-II.2 Decoupling of `OCPI_CDK_DIR` and `OCPI_ROOT_DIR` Environment Variables

The location of the “root” and “cdk” directories depends on the installation and mode. Previously, `OCPI_CDK_DIR` had to be a direct sub-directory of `OCPI_ROOT_DIR`. Effective with this release, `OCPI_CDK_DIR` and `OCPI_ROOT_DIR` have been decoupled, and deviations from the “normal/usual” directory structure only require changing one line in the relevant setup script.

B-II.3 `data_sink_qdac` and `data_src_qdac` Workers

The platform HDL device workers `data_sink_qdac` and `data_src_qdac` previously used the “ComplexShortWithMetadata” protocol. They now use the “`complex_short_timed_sample`” protocol.

B-II.4 Python ACI Update

The annoying need to prefix strings with “b” to force treatment of same as bytes objects (vs. Unicode str objects – the Python 3 default) has been eliminated.

B-III Major Known Deficiencies

B-III.1 Previously Documented Deficiencies

The following deficiencies are still present in 2.2:

- 32-bit Architecture File Size Limit (cf. M-II)
- OpenCPI HDL Assembly Filename Constraint (cf. M-I)
- Maximum Message Length in PCI-Based Platforms (cf. L-III.1)
- PCI-Based Platforms Not Always Detected in Remote Tests (cf. K-III.5)
- PCI Platforms Can Only Transmit Using One Channel (cf. I-III.1)
- Connection Direction In GUI Assumes XML order (cf. H-III.2)

C Release 2.1.0

C-I Enhancements

C-I.1 GNU Radio Installation Support

A new convenience script, `scripts/install-gnuradio.sh`, is available to install OpenCPI's modified version of GNU Radio 3.7 and GNU Radio Companion. Supported host operating systems are CentOS 7 and Ubuntu 18.04.

C-I.2 ADI mykonos Library

For those writing device workers (proxies) for the ad9371 (a.k.a. mykonos), the `mykonos` library has been added as a prerequisite library.

C-I.3 FSK Application on Matchstiq Z1

The matchstiq z1's lime device workers are now able to run in baseband loopback mode for the FSK application.

C-I.4 ComplexShortWithMetadata-prot.xml Sequence Length

`ComplexShortWithMetadata-prot.xml` has been updated to use a sequence length of 4096 instead of 4092 for the "iq" argument of the samples operation. Workers or primitives assuming a sequence length of 4092 should be updated to use 4096 instead.

C-I.5 Digital Radio Controller

The new Digital Radio Controller (DRC) component is used when an application needs to use radio hardware in the system, for the purpose of controlling it and streaming sample data to and from it. The digital radio functionality in a system usually has antennas for transmitting and receiving RF signals, and channels which convert the RF signals to and from base-band digital samples that are produced and consumed by the application.

See the "OpenCPI Application Development Guide", the "OpenCPI Platform Development Guide", and the new "Digital Radio Controller" briefing for more information.

C-I.6 Proxy Workers and Slaves Model

An enhanced model of proxy workers and slaves has been implemented which renders previous methods obsolete and deprecated. The new model is that a proxy worker's slaves are automatically included in the application when that worker is chosen for some component instance in the application. A worker author specifies the slaves of a proxy in the form of a subassembly, which can have instance properties and connections. External ports of the subassembly are associated with ports of the proxy such that application connections to the proxy's ports may be delegated to the external ports of the slave subassembly. In this way, the application does not care about the slaves and does not mention the slaves. This allows applications to be more portable since they do not need to know about slaves at all. This is particularly useful for the newly released DRC component specification used for controlling radios, since different DRCs use different slaves.

Workers in the slave assemblies (as well as HDL assemblies and the optional "worker" attribute of app instances) may now use fully qualified package-ids.

C-I.7 Proxy Worker Capabilities

As part of a package of enhancements for the capabilities of proxy workers, delegated ports have been upgraded to support delegating a proxy port that has a count (that is an array of ports) such that the underlying slave assembly can connect to the individual ports in the proxy's delegated array ports. The immediate use-case is when DRC proxies support multichannel radios and thus have RX and TX ports that are arrays of ports for supporting the multiple DRC channels.

C-I.8 Scalable Software-side Buffer Count Without Memory Bloat

The Scalable Data Plane (SDP) is now 64 bits wide on zynq platforms.

The CPU-side number of DMA buffers on HDL->RCC connections can now be large – up to 255.

The amount of DMA memory allocated can now be specified as an option to the `ocpidriver` and `ocpiremote` commands.

A simple `test_source` component, and associated RCC and HDL workers, have been added and are useful for performance testing.

The `file_write` (RCC) worker now reports bytes/second and has a data integrity checking option when writing what is generated by the `test_source` component, as well as an option to suppress writing the data at all.

C-I.9 Timed Transmission

A time gating HDL worker has been added to support timed transmission. It allows samples to flow through until it receives a time opcode and timestamp for a scheduled time to transmit. When it receives the time opcode and timestamp it will stop samples from flowing through until the scheduled time to transmit and then it will let samples flow through again. It also allows setting a time correction, that is an unsigned fixed point fraction of second, which is subtracted from the scheduled time to open the gate early so that samples flow at the right time.

C-I.10 Performance Improvements

All zynq platforms always used the 64 bit AXI DMA interfaces in the zynq chip, but now the SDP/DMA FPGA modules are also 64 bits wide whereas they used to be 32 bits wide. This allows the DMA to have higher throughput and no idle cycles when producing continuous 32 bit data.

With the exception of zynq ultra, zynq platforms now have the option of using cache coherent DMA.

Options may now be specified to build both the framework itself and software assets (RCC workers, ACI programs) with compiler optimizations, alongside the existing default build for best debugging. I.e., both the default debug builds as well as the optimized builds coexist in the source tree installation, and both can be built and used. The “`--optimize`” option applies to the `opencpi-setup.sh` script (to use the optimized framework build) as well as the `ocpiadmin install platform` and the `ocpidev build` commands (to perform an optimized build). An alternative way to enable building with optimization is to use a “`-o`” suffix at the end of software platform names: the idea is that anywhere in the CLI you have a software platform, you can qualify it with build options and it creates a “pseudo platform” with the specified option, which does not interfere with the default platform (without the suffix). This applies to `ocpiadmin install platform`, `ocpidev build --rcc-platform`, and `ocpiremote load --rcc-platform`. When running applications with RCC workers, when both debug and optimized RCC workers are built (i.e. both artifacts are available in `OCPI_LIBRARY_PATH`), preference is given to the worker whose optimization setting matches the optimization setting of the container in which it is executing.

`ocpiremote` supports the “`--optimize|--optimized`” option to specify that the optimized RCC platform build products should be used.

C-I.11 OCPI_ROOT_DIR Environment Variable

Files in scripts/programs may now be referenced relative to the OpenCPI “root” directory, as well as the “cdk” directory. The following scripts have been modified to take advantage of the new `OCPI_ROOT_DIR` environment variable: `opencpi-setup.sh`, `zynq-setup.sh`, `zynq-net-setup.sh`, and `ocpiserver.sh`.

C-I.12 e3xx OSP

The `e31x` platform project is now based on `xilinx19.2_aarch32` instead of `xilinx13.4`.

C-I.13 timestamper_scdcd-spec.xml Properties

The `timestamper_scdcd` spec file properties now include “`samples_per_timestamp`” (replaces “`min_num_samples_per_timestamp`”) and “`sampling_interval`”.

C-II Changes and Notes

C-II.1 ocpigen

64-bit initial values for 64-bit HDL worker parameters are now handled correctly.

C-II.2 ocpiremote Option Standardization

The `ocpiremote` command options have been updated to match those used in other commands. Specifically, `ocpiremote` now understands “`--rcc-platform`”, “`--hdl-platform`”, and “`--verbose`”. While “`--sw-platform`” and “`--hw-platform`” are still allowed, these options are now deprecated.

C-II.3 platform_pkg Updates

The “`usingPPS`” signal was added to the `time_base_out.t` record in `projects/core/hdl/primitives/platform/platform_pkg.vhd`. Platform workers need to drive this signal either high or low depending on their configurations and whether a PPS is being used.

The “`time_base_out.t.reset`”, “`time_base_out.t.ppsIn`”, “`time_base_in.t.ppsOut`”, and “`time_service.t.reset`” signals have been deprecated and should no longer be used. Platform workers should no longer drive “`timebase_out.reset`”, “`timebase_out.ppsIn`”, or “`timebase_in.ppsOut`”.

C-III Major Known Deficiencies

C-III.1 Previously Documented Deficiencies

The following deficiencies are still present in 2.1:

- 32-bit Architecture File Size Limit (cf. M-II)
- OpenCPI HDL Assembly Filename Constraint (cf. M-I)
- Maximum Message Length in PCI-Based Platforms (cf. L-III.1)
- PCI-Based Platforms Not Always Detected in Remote Tests (cf. K-III.5)
- PCI Platforms Can Only Transmit Using One Channel (cf. I-III.1)
- Connection Direction In GUI Assumes XML order (cf. H-III.2)

D Release 2.0.0

D-I Enhancements

D-I.1 Deprecations

The following have been marked as “deprecated” and support has been removed in OpenCPI 2.x.

- CentOS 6 reaches end-of-life on 30 Nov 2020 and will no longer receive updates or security patches by the CentOS maintainers. Because of this, OpenCPI has dropped support for CentOS 6 as a development platform and an RCC runtime platform.

D-I.2 Binary Distribution Builds

For parties interested in providing OpenCPI binary packages for distribution, the `OCPI_DISTRO_BUILD` environment variable has been added. If set to “1”, OpenCPI build/installation scripts will produce binaries that can be run on any supported x86_64 platform.

D-I.3 CentOS 8

Support for CentOS 8 as an RCC only development platform has been added. HDL support will be made available when vendor tools support CentOS 8. CentOS 7 is still the recommended development platform for new and existing OpenCPI projects.

D-I.4 Digital Radio Controller for E31x

Adds digital radio controller (DRC) implementation for E31x platform. FSK app `fsk_dig_radio_ctrl` is updated to show RX, TX, TXRX and is capable of being run using both `ocpirun` and `ACI` app.

D-I.5 Man Pages

Additional man pages for `ocpidev` nouns are available. This includes man pages for: `ocpidev-application`, `ocpidev-assembly`, `ocpidev-card`, `ocpidev-component`, `ocpidev-device`, `ocpidev-library`, `ocpidev-platform`, `ocpidev-primitive`, `ocpidev-protocol`, `ocpidev-registry`, `ocpidev-slot`, `ocpidev-subdevice`, `ocpidev-test`, `ocpidev-worker`.

New man pages include: `ocpiadmin`, `ocpidriver`, `ocpihdl`, `ocpiremote`, `ocpirun`, `ocpserve`, `ocpview`

When the OpenCPI environment is setup using `opencpi-setup.sh`, the man pages can be viewed using the `man` command, e.g. `man ocpidev`.

Man pages are also available in HTML form at `$OCPI_CDK_DIR/doc/man/html` including cross references between man pages.

D-I.6 ocpiadmin

A new command line tool `ocpiadmin` has been added. This new tool replaces `scripts/install-platform.sh` and `scripts/deploy-opencpi.sh` on source-installed OpenCPI development platforms. Usage of `ocpiadmin` requires the OpenCPI environment be setup.

Example usage of `ocpiadmin`:

```
# Required
source cdk/opencpi-setup.sh -s

# Install xsim platform
ocpiadmin install platform xsim

# Install e31x platform
ocpiadmin install platform e31x -p ospi.osp.e3xx

# Additional help
ocpiadmin install --help
```

D-I.7 Optional Slave Support

Optional slaves means these things:

- the proxy developer can declare that slaves are optional (the app can leave some out)
- the proxy developer can use a runtime query to see if a slave it declared as optional, is in fact present or not.
- the app developer can specify only a subset of the slaves of a proxy
- the attributes in the slave element in the OAS use “slave” and “instance” rather than “name”

D-I.8 Proxy Slaves

When a proxy instance in an application specifies no slaves, the application will be searched for possible slaves that belong to that proxy (that are the right worker). If this is ambiguous there will still be an error.

D-I.9 Selection Expressions

The “worker” variable is now available in selection expressions, and the “worker” attribute of application instances can now include the authoring model suffix.

D-I.10 Ubuntu 18.04

Support for Ubuntu 18.04 as a full development host has been added. Ubuntu 18.04 is the only platform that supports OpenCPI integration with GNU Radio’s “GNU Radio Companion”. Developers and users of OpenCPI looking to use Ubuntu as a development host platform instead of CentOS are highly encouraged to use Ubuntu 18.04 instead of Ubuntu 16.04.

D-I.11 Xilinx RCC Platforms

Support for additional Xilinx platforms as been added. New additions include: `xilinx13.4_arm`, `xilinx17.1_aarch32`, `xilinx17.1_aarch64`, `xilinx17.2_aarch32`, `xilinx17.2_aarch64`.

These new platforms, along with existing platforms, can be built using the new `ocpiadmin` tool.

D-II Changes and Notes

D-II.1 Component Library Searching Algorithm

Component Libraries are searched per project, with all indicated libraries searched in a project before moving to the next project. Previously, this was “library first” which caused confusing results. If a worker in an assembly is found in a library, it is chosen even if not built for the target platform. This is to maintain consistency between library searching that is target-independent with library searching by tools that is target-dependent.

D-II.2 ocpigen

The “built-in” properties are now “hidden” from the property dump display unless explicitly enabled.

D-II.3 PlutoSDR RCC Platform Support

PlutoSDR RCC platform has been updated to use new version 0.32. Installation of the platform no longer builds pluto.frm image as pre-built image by Analog Devices can be used directly. Cross compiler tools have also been updated to use Xilinx version 2019.2 and no longer require a separate installation of Xilinx 2018.3 SDK. NFS is not supported in this version of the kernel which implies drop of support for network mode; standalone and remote modes are still supported.

D-II.4 Primitive Libraries `util` and `bsv`

The `util` and `bsv` primitive libraries are not automatically made available to workers, to reduce namespace pollution. Primitive libraries are transitive - if a worker needs a library, the assembly automatically knows that and you do not have to mention the primitive library for the assembly.

The `bsv` library usage is deprecated and it is now scoped in VHDL by the name of the package it exists in, so that a typical usage in VHDL is: `library ocp_core_bsv; use ocp_core_bsv.all`

In general, primitive libraries can be labeled as being qualified by the package id of the project it is in, so that a developer does not need to now if the library name is used elsewhere.

D-II.5 Test Bitstream `testbias`

The test bitstream `testbias` is always exported and exists at the top of the framework’s directory for the HDL platform. This same file is made available for SD cards as well as when using `ocpiremote` to run a remote system.

D-II.6 Tutorials

More of the assets created in the tutorials are now in the tutorial project.

D-III Major Known Deficiencies

D-III.1 Previously Documented Deficiencies

The following deficiencies are still present in 2.0:

- 32-bit Architecture File Size Limit (cf. M-II)
- OpenCPI HDL Assembly Filename Constraint (cf. M-I)
- HDL Platform `zed_ise` Build Error (cf. E-III.1)
- Maximum Message Length in PCI-Based Platforms (cf. L-III.1)
- PCI-Based Platforms Not Always Detected in Remote Tests (cf. K-III.5)
- PCI Platforms Can Only Transmit Using One Channel (cf. I-III.1)
- Connection Direction In GUI Assumes XML order (cf. H-III.2)

E Release 1.7.0

E-I Enhancements

E-I.1 ADC/DAC Clocking Architecture

A new clocking and timestamping model has been implemented for ADC and DAC devices that is expected to be rolled out across all HDL platforms and devices for 2.0. It simplifies some device workers, minimizes clock-domain crossing and time-jitter for sample data, as well as supports the `ComplexShortWithMetadata` protocol more directly. All new device support for ADC/DAC/transceiver devices should use this model. It is described in the OpenCPI Platform Development Guide, section 5.5.9. For 1.7 it is implemented on the e31x and zed+fmcomms platforms.

E-I.2 Built-in platform Project

The new built-in `platform` project contains reference platforms and support for generic devices and cards. The `zed` and `zed_ise` platforms have been moved to this project, with future reference platforms expected to follow. Most device support in the built-in `assets` project is expected to migrate to the `platform` project for 2.0. Any OSP project which relies on such generic support or reference platforms should include `ocpi.platform` as a project dependency. See the README file in this new built-in project for more details.

E-I.3 CIC Decimator RCC work-a-like

An RCC work-a-like of the `CIC.Dec.hdl` worker has been added to the built-in project “assets”.

E-I.4 Clock Generator Primitive

A new clocking library has been added to `core/hdl/primitives`. This is a preliminary view of how OpenCPI will handle vendor clock generator support.

E-I.5 Man Pages

Man pages are available for the `ocpidev` command and its subcommands. When the OpenCPI environment is setup using `opencpi-setup.sh`, the man pages can be viewed using the `man` command, e.g. `man ocpidev`.

E-I.6 `ocpiremote`

`ocpiremote.py` was added to remove, or reduce, the need for setting up NFS for the purposes of remote application execution and development. When the OpenCPI environment is setup using `opencpi-setup.sh`, running `ocpiremote --help` will list available subcommands. Additional help for each subcommand is available via `ocpiremote <subcommand> --help`.

E-I.7 PlutoSDR HDL Platform Support

Support for plutosdr platform based on adi_plutosdr0_31 RCC platform has been added. The plutosdr platform is the cheapest starter platform for OpenCPI, featuring a Zynq 7010 FPGA and AD9363 transceiver chip on the same board.

Note: Xilinx SDK 2018.3 is required to build the framework, RCC artifacts, and ACI executables for the plutosdr platform.

E-I.8 Protocol `complex_short_samples`

A new protocol, `complex_short_samples`, has been added which is intended to be used for streaming complex data and metadata commonly associated with SDR applications. For expansion of future capabilities, this protocol includes an operation for user-defined metadata. This protocol is expected to replace `ComplexShortWithMetadata` in future releases.

E-I.9 Ubuntu 16.04 Support

Users now have the option of using Ubuntu 16.04 as their host development platform. The level of support for Ubuntu 16.04 is the same as CentOS 7.

E-I.10 zcu104 HDL Platform Support

Support for the zcu104 platform based on xilinx19.2 aarch64 RCC platform has been added. The zcu104 platform is like the zed platform: the cheapest “reference” platform for the zynq-ultrascale processor line, that also can support the ad9361.

Note: Vivado 2019.2 is required to build artifacts for the zcu104 platform. Vivado 2019.2 is not supported on CentOS 6.

E-II Changes and Notes

E-II.1 `ComplexShortWithMetadata` Changes

Removed `user` opcode and further clarified the descriptions of the `sync` and `end_of_samples` opcodes. This protocol is finalized, meaning little to no future changes.

E-II.2 MacOS 10.14 and 10.15

Both MacOS 10.14 (Mojave) and MacOS 10.15 (Catalina) have been requalified as RCC-only platforms, which can still be used as a client for remote containers. Support for these RCC-only platforms relies on MacPorts already being installed on the user’s system.

E-II.3 Optional HDL Ports with `clockdirection=in`

Optional HDL worker ports that have the “clockdirection” attribute set to “in” are now supported.

E-II.4 Python Scripts

Removed all Python 2 related packages and dependencies for OpenCPI. On CentOS 6, there are still python 2 related packages being installed as the `gpsd` prerequisite requires python 2 for SCons.

The shebangs in all OpenCPI python scripts have been changed from `#!/usr/bin/env python3.4` to `#!/usr/bin/env python3` allowing for greater portability across operating systems that provide different python 3.x versions.

All OpenCPI python scripts have had their execute bit set so they can be executed directly without having to explicitly invoke a python interpreter.

Note: OpenCPI requires at least python 3.4 and does not use any newer features added in later versions of python 3. This minimal version requirement is based on what is available for CentOS 6. OpenCPI installation on CentOS 7 installs and uses python 3.6 and related packages.

E-II.5 SCons 3

Related to python 3 changes, the SCons dependency was updated from SCons 2 to SCons 3 on CentOS 7. Due to how SCons is packaged for CentOS 7, one cannot have both `scons` and `scons3` RPMs installed due to conflicts with the RPMs. SCons is used for building the `gpsd` prerequisite.

E-II.6 SD Card Export Model

The process of generating SD cards for embedded systems was revamped and simplified. In particular the syntax for the platform exports files was changed/enhanced. Developers of platforms should read the appropriate section in the OpenCPI Platform Development Guide. The change reduces the requirements to change the embedded system's root file system from the manufacturers default.

E-III Major Known Deficiencies

E-III.1 HDL Platform `zed_ise` Build Error

Known issue with building HDL platform `zed_ise`. See #1113 for status of this issue.

E-III.2 Remote Development using Network Mode Scripts

Known issue related to using network mode scripts for remote development (ex. `mynetsetup.sh`). Status can be tracked here: #1117, #1204

E-III.3 RPM Support

RPMs are not supported in OpenCPI 1.7.0. Support will be added in OpenCPI 2.0.0.

E-III.4 Previously Documented Deficiencies

The following deficiencies are still present in 1.7:

- 32-bit Architecture File Size Limit (cf. M-II)
- OpenCPI HDL Assembly Filename Constraint (cf. M-I)
- Maximum Message Length in PCI-Based Platforms (cf. L-III.1)
- MAC Address on ZedBoard when Using the xilinx13.3 Software Platform (use xilinx13.4 to avoid the issue) (cf. L-III.2)
- PCI-Based Platforms Not Always Detected in Remote Tests (cf. K-III.5)
- PCI Platforms Can Only Transmit Using One Channel (cf. I-III.1)
- Connection Direction In GUI Assumes XML order (cf. H-III.2)

F Release 1.6.0

F-I Enhancements

F-I.1 Support for Workers with Separate Control and Data Clocks

Each worker data port can now declare whether the clock domain of the port's signals is:

- The same as the control port and clock (as before, which is still the default)
- Based on an output clock driven by worker code, indicated using the `clockDirection='out'` attribute
- Based on an input clock accepted into the port, indicated using the `clockDirection='in'` attribute
- The same as some other port of the worker, indicated using the `clock='other-port-name'` attribute

One important use case for this is for all data communication and data processing to happen in a clock domain different from the control clock domain. When worker ports are connected that are in different clock domains, a clock-domain-crossing adapter is inserted which is conservative with respect to the relative clock rates and phasing. For more information, consult the OpenCPI HDL Development Guide.

F-I.2 Issues Database for OpenCPI is now Publicly Accessible

Prior to this release, the issues/bug database used by the active OpenCPI developers and maintainers was private and not visible. Now the OpenCPI developers and maintainers use the GitLab issues database that is associated with the new public GitLab repository at gitlab.com/opencpi/-/issues.

F-I.3 Source Installation Simplified

When installing from source, various steps of installation, including the AV GUI installation and cross-building for embedded software and FPGA platforms, requires fewer steps and less complexity. Installation of an FPGA simulator has also been simplified since the Vivado WebPack package no longer requires any licensing at all. See the OpenCPI Installation Guide.

F-I.4 Time Service Interfaces were Upgraded for better Timestamping Support

For platform developers, the properties of the `time_server` worker instantiated in all platform configurations have been updated and clarified. The VHDL signals associated with the timebase and timeservice platform port types have new signals. The time service now supports the indication as to whether the time values come from valid GPS-based time. Platforms with GPS receivers connected to CPU serial ports (e.g. Ettus e310) enable time values provided on worker time interfaces to be GPS-based time values. This is a technology preview until the next release when most supported platforms will use this capability to provide GPS-based timestamping to workers.

F-I.5 New Features for Simplifying ADC/DAC Devices Workers and Enabling Timestamping

For platform developers, special workers have been developed to allow the data ports of ADC and DAC device workers to remain in the sample clock domain and not be involved in timestamp injection. These are a technology preview until the next release when they will be used for supported platforms. See the specs and workers for `data_src_adc` and `data_sink_dac` in `projects/assets/hdl/device/`. Similarly there is now a new timestamping worker that receives inputs in the sample clock domain. See `timestampersdc` in `projects/assets/components/util_comps`.

F-II Changes and Notes

F-II.1 Repository of OpenCPI moved to GitLab.com

The git repository site where OpenCPI is hosted and actively maintained is now gitlab.com. The documentation for releases is found at opencpi.gitlab.io.

F-II.2 Project with Content for Tutorials is in the Core Repository

The project used for tutorials which was previously in a separate training repository is now in the core OpenCPI repository under `projects/tutorial`.

F-II.3 MacOS Catalina (10.15) is now Supported for Software Only (no FPGA) Development.

Software only development (no FPGA tools or simulators) is supported on MacOS Catalina. It relies on the `macports` package manager already being installed on the system, from [macports.org](https://www.macports.org).

F-II.4 The gpsd package is now a prerequisite for all OpenCPI systems.

OpenCPI now has a core dependency on this package ([gpsd.gitlab.io/gpsd](https://gitlab.com/gpsd)) and it will be built for all development hosts and embedded systems and used in a way that will not conflict with systems that already have `gpsd` installed.

F-II.5 Optional Environment Settings for a Source Installation are in a New File.

All environment settings for OpenCPI (which are all optional), should be set in the `user-env.sh` file at the top level of the installation. This file is automatically read when the OpenCPI environment is established.

F-III Major Known Deficiencies

F-III.1 RPM Installation is not Supported in Release 1.6.0

In the initial 1.6.0 release of 1.6, RPM installation is not supported. Only source installation is supported, as described in the OpenCPI Installation Guide. RPM installations will be supported by the next minor release.

F-III.2 Previously Documented Deficiencies

The following deficiencies are still present in 1.6:

- 32-bit Architecture File Size Limit (cf. M-II)
- OpenCPI HDL Assembly Filename Constraint (cf. M-I)
- Maximum Message Length in PCI-Based Platforms (cf. L-III.1)
- MAC Address on ZedBoard when Using the xilinx13.3 Software Platform (use xilinx13.4 to avoid the issue) (cf. L-III.2)
- PCI-Based Platforms Not Always Detected in Remote Tests (cf. K-III.5)
- PCI Platforms Can Only Transmit Using One Channel (cf. I-III.1)
- Connection Direction In GUI Assumes XML order (cf. H-III.2)

The following previously-documented deficiencies have been fixed in this release:

- Properties Before Raw Properties Cannot Have Parameterized Lengths (cf. K-III.7)
- Lime RX/TX Proxy Tests on CentOS6 (cf. K-III.10)
- Some HDL Workers Will Need To Be Updated When `props_in.raw.address` Width Fixed (cf. H-III.4)

G Release 1.5.0

G-I Enhancements

G-I.1 New Digital Radio Controller Front-End Control Paradigm

(Technology Preview) This release includes a new OpenCPI Specification File (OCS) which represents the next-generation interface for Radio Frequency (RF) command and control. This generation improves upon the previous by removing RF vs. Base Band (BB) specificity, allowing for runtime querying of acceptable settings, and supporting multiple RF channels. Each of these marks an important step towards supporting truly hardware-agnostic RF applications.

For more information, consult `Dig_Radio_Ctrlr.pdf`.

G-I.2 New Time-Based Protocol (“ComplexShortWithMetadata-prot”) (Technology Preview)

One of the major improvements as of 1.5 is to have timestamped data flow from end-to-end using the “RxApp” on an E3xx example platform. To achieve this, “ComplexShortWithMetadata-prot” was created. The source to this Protocol (`<core>/specs/`) has various comments explaining the new opcodes and arguments as well as example usage.

G-I.3 New Time-Based Assets Project (“assets_ts”) (Technology Preview)

The workers required to achieve the previously mentioned goal of timestamped data within “RxApp” were derived from those within “assets” but are now in their own Project, “assets_ts”. This Project mostly exists to work around current limitations within the Framework; whether or not it will continue to exist or if the timestamped versions will return to “assets” is still to be determined.

G-I.4 New HDL Worker Port Interfaces

A new HDL Worker port interface is now available that considerably reduces code complexity when a Worker “doesn’t care” about message boundaries, *e.g.* when streaming sequences of data. This interface is currently requested by noting in the OWD that it is “`version=2`” and this interface will become the default in OpenCPI 2.0. Additional information can be found in the *HDL Development Guide*.

Some Workers Migrated Some of the HDL Workers’ source have been updated to use this new “`version=2`” interface to exemplify its usage, while the “version 1” (pre-1.5) Worker source code has been migrated to the “inactive” Project. This *should not affect* any applications that previously used these Workers, except the Properties explicitly affecting buffer sizes are now silently ignored. The following Workers have been upgraded in this way:

<code>agc_real.hdl</code>	<code>cic_int.hdl</code>	<code>mfsk_mapper.hdl</code>
<code>cic_dec.hdl</code>	<code>dc_offset_filter.hdl</code>	

G-I.5 ocpidev Improvements

The `ocpidev` tool has been significantly improved, including:

- ability to show utilization of various HDL Workers and Assemblies
- `show component` shows information about Components, including Properties and Ports
- `show worker` information about Workers, including Properties and Ports

See G-II.6 for `ocpidev` command changes.

G-I.6 New “Zed” RCC Platform: `xilinx13.4`

With the introduction of the RCC Platform `xilinx13.4` to support the E310 in 1.4, with release 1.5 there are now RPMs available pairing the Zed with both `xilinx13.3` and `xilinx13.4`. The RPMs follow the naming convention `opencpi-hw-platform-zed-<RCCPlatform>`. See the *Zed Getting Started Guide* for more information.

G-I.7 gdb and rsync Provided on xilinx13.3 and xilinx13.4 RCC Platforms

Both of these tools are very helpful with development and an ARM-compiled version is now provided in the SD card images provided by OpenCPI. Out of the box, **rsync** must be run *from the radio* targeting the development host; more information is available in the applicable *Getting Started Guides*.

G-II Changes and Notes

G-II.1 Deprecation of the “Readable” Attribute

The “**readable**” Property Accessibility Attribute has been deprecated and replaced with “**readback**.” *However*, its use case is very rare and specific; by default all Properties are “readable” in the usual context of the term. Additional information can be found in the *Component Development Guide*.

G-II.2 Deprecation of Zipper RF Interface Board

The Myriad RF Zipper Interface Board has been unavailable for purchase for a considerable time. While OpenCPI still *theoretically* supports it, limited available quantities have resulted in its removal from the list of officially supported (and therefore tested) platforms. Support infrastructure has been moved to the “inactive” project.

G-II.3 E3xx PPS Discipline

The “e3xx” Platform Worker timebase interface’s PPS disciplining was previously limited to only the E310/E312’s “SYNC” SMB connector (GPS PPS disciplining was not possible). The source selection is now exposed as a parameter on the Platform Worker.

G-II.4 “new_project_source” script removed

The utility script noted in J-II.1 has been replaced by the more versatile “ocpi-copy-projects” script. This new utility is in the user’s path and has a “--help” option.

G-II.5 RCC Worker Parameter Macros Stricter

If an RCC Worker had been using the undocumented / internal “macro form” of a parameter, *e.g.* “PARAM_biasValue()”, that form will no longer work. Instead, RCC Workers *must* use the documented form, *e.g.* “BIAS_PARAM_BIASVALUE”¹.

G-II.6 ocpidev Changes

- The **show projects** command no longer shows tests and primitives with low verbosity (“-v”); use high verbosity instead (“-vv”)².

G-II.7 “myClock” HDL Code Deprecated / Removed

While revamping and examining the framework’s ability to use multiple clocks within HDL Workers, it was found that the “myClock” capability was documented but likely never fully worked. This feature is *planned* to be restored *and tested* for the 1.6 release using a different XML attribute name.

G-II.8 Infrastructure OWDs Moved

Various OWDs³ were incorrectly located within the “core” Project’s “specs” directory, even though they are infrastructure Workers that an end user shouldn’t need to be concerned with. They have been moved out to reduce confusion and clutter.

¹These examples are from the `bias_param.c` worker, comparing 1.4 to 1.5.

²Any automated parsing of data should be using JSON interface, which also needs this additional verbosity.

³The OWDs explicitly were `metadata.xml`, `ocscp.xml`, `time_client.xml`, and `unoc_node.xml`.

G-II.9 RPM Upgrade Notes

Some previous RPM-based installations may not have proper dependencies documented for some sub-packages, *e.g.* version 1.4 of `opencpi-hw-platform-e3xx-xilinx13_4` requires “`opencpi`,” not *explicitly* “`opencpi version 1.4`.” This may cause version mismatches when upgrading; **ensure all packages currently installed on a system are *explicitly* added to the yum command line when upgrading.**

G-II.10 HDL Wrapper Changes

The internal “wrapper” that encapsulates an HDL Worker has changed considerably (see Section G-I.4). This has changed the pipeline depths of the inputs and outputs of the Workers. **This *should not affect* any users; an Application should *never* make assumptions concerning, or require specific relationships between, the timing of data between Workers⁴.** However, if developers previously made implementation-specific assumptions, their Applications will now fail and will need to be fixed to properly use the Framework to transport *all* required data and metadata.

G-III Major Known Deficiencies

G-III.1 “udev” Radio Rules Sometimes Does Not Work

Starting with 1.4, some RPMs pre-install symlinks for rules in `/etc/udev/rules.d/` for certain radio platforms, *e.g.* a rule to create a symlink from `/dev/matchstiq_z1_0` to `/dev/ttyUSB0`. If the OpenCPI installation location is on a different partition than the host machine’s root partition, the symlinks for these rules may be broken at boot-time because the target partition is not yet mounted when `udev` is reading the rules. To fix this, the command “`sudo udevadm control --reload`” should be executed upon a full boot. With CentOS 7, a custom `systemd` service can be implemented to perform this action when the partition has been mounted.

G-III.2 Emulators Not Constrained To Specific Containers In Unit Tests

When an emulator-based test harness is built for multiple simulators (*e.g.* “`isim`” and “`xsim`”), running the unit test does not guarantee that all workers are running on the same container. For example, when “`lime_dac_ts.test`” is built for both `isim` and `xsim`, test execution could pass for `isim`, but might later fail with `xsim`. Review of the log file shows that the emulator worker is erroneously running on an `isim` container and not the expected `xsim` container. The recommended workaround is to only build and test for a *single* simulator at a given time, ensuring there is only one artifact that satisfies the test’s requirements.

G-III.3 Provided LiquidDSP Build May Link Incorrectly

Note: This bug *only* affects source-based building; RPM-based users will not encounter this issue.

The “LiquidDSP” library is provided as a prerequisite of the “assets” Project. It is currently used by one worker, “`complex_mixer.rcc`”, which uses it for its Numerically Controlled Oscillator (NCO). If the LiquidDSP library is built on a system that has the “FFTW” library installed, it will attempt to use it. However, any user-developed RCC Worker attempting to use LiquidDSP functions that use the FFTW back-end will fail at runtime with an undefined symbol (*e.g.* `fftwf_execute`). The recommended workaround is to use the compiled `libliquid` from the RPM installation (*e.g.* `/opt/opencpi/cdk/centos7/lib/libliquid.a` or `/opt/opencpi/cdk/xilinx13_3/lib/libliquid.a`) or to rebuild the prerequisites without the FFTW libraries installed.

This problem was found late within the 1.5 release cycle and will be addressed with the next major release.

G-III.4 Previously Documented Deficiencies

The following deficiencies are still present in 1.5:

- 32-bit Architecture File Size Limit (cf. M-II)
- OpenCPI HDL Assembly Filename Constraint (cf. M-I)
- Maximum Message Length in PCI-Based Platforms (cf. L-III.1)

⁴Other ways timing may be affected would include backpressure from upstream Workers, network latency between remote Containers, etc.

- Properties Before Raw Properties Cannot Have Parameterized Lengths (cf. K-III.7)
- MAC Address on ZedBoard when Using the xilinx13_3 Software Platform (use xilinx13_4 to avoid the issue) (cf. L-III.2)
- PCI-Based Platforms Not Always Detected in Remote Tests (cf. K-III.5)
- Lime RX/TX Proxy Tests on CentOS6 (cf. K-III.10)
- PCI Platforms Can Only Transmit Using One Channel (cf. I-III.1)
- Connection Direction In GUI Assumes XML order (cf. H-III.2)
- Some HDL Workers Will Need To Be Updated When `props_in.raw.address` Width Fixed (cf. H-III.4)

The following previously-documented deficiencies have been fixed in this release:

- ZedBoard IP Required Even In Standalone Mode (cf. H-III.1)
- OnlyPlatforms/ExcludePlatforms Does Not Work For Individual Unit Test Cases (cf. H-III.3)
- Some `ocpi.assets` Workers Have Flow Control Issues (cf. H-III.6)
- Default `python` Must Be Python2 (cf. H-III.5)

H Release 1.4.0

H-I Enhancements

H-I.1 Support for Ettus E310 Platform

The Ettus Research USRP E310 is now supported within the OpenCPI framework. This includes the changes to the AD9361 device workers for using a CMOS interface, increasing the number of modes that the framework supports⁵. There is currently a limitation of a single TX and RX channel being supported; see the *E3xx Getting Started Guide* for more information.

H-I.2 Component Unit Test Enhancements

The provided Component Unit Test infrastructure has been enhanced to automatically incorporate elements that address common issues when developing HDL workers:

1. Input message metadata manipulation applied to the Unit Under Test's (UUT) input port(s) helps ensure an HDL Worker can properly handle various message-protocol edge conditions, *e.g.* start-of-message without data until a later clock. **This manipulation is *not* enabled by default.**
2. Output message throttling applied to the UUT's output port(s). Not respecting “backpressure” is often the reason an HDL Worker fails on a “real” hardware platform after passing all tests within a simulation. **This backpressure is now enabled in *all* test benches by default.**

Exercising Workers in this manner during the simulation design phase has been shown to greatly increase the likelihood of success on physical hardware. More information can be found in the *Component Development Guide*.

H-I.3 Multi-Slave Proxy Support

Previously, a Proxy Worker could only represent a single Worker but not a collection of Workers that had related functions, *e.g.* the transmit and receive paths of a single RF link. This resulted in various workarounds that are not officially supported by the Framework. For example, one user had a Proxy “peeking” into the Application to find peer Workers: this does not work with Remote Containers with the Application on another physical machine. Now a Proxy can (optionally) define more than one slave Worker with full interface compatibility with previous implementations⁶. More information can be found in the *Component Development Guide*.

H-I.4 Transmit Blanking / Powerdown for DACs

Both the AD9361 and LIME DACs are now blanked / powered down⁷ when not in use. To perform this on-demand, they use a new “tx.event” Port/Protocol, as well as properties exposed in their proxies. See the respective Worker datasheets for more details. **This added capability resulted in a change to the qdac-spec that may affect current user applications.** If an application used `connect=qadc` (connect attribute), that application will require an additional `to` attribute specified directing it to explicitly use the IN input port.

H-I.5 ocpidev Improvements

The `ocpidev` tool has been significantly improved, including:

- run some Applications
- run Unit Tests
- show Unit Tests within a Project
- show additional Project information

See H-II.7 for `ocpidev` command changes.

⁵AD9361 LVDS signaling has been supported since 1.3; cf. J-I.3.

⁶The interface of using an object named “slave” remains. The type of the object is considered an implementation detail and *has* changed.

⁷Implemented by TX RF carrier suppression

H-I.6 VHDL-Native Pattern and Capture HDL Workers

The `pattern.v2` and `capture.v2` Workers have been written from the ground up in VHDL to replace the machine-generated Verilog-sourced `pattern` and `capture` Workers. **These latter Workers are now considered deprecated.** The new Workers include fully-functional test benches and data sheets. Platform and BSP Developers will find these Workers to be very useful when debugging various issues related to low-level OpenCPI development.

H-I.7 New Documentation RPM

In addition to the PDF-based documentation available at github.io, a new RPM is now included “`opencpi-doc`,” which includes most of that documentation. Most of the PDFs are installed into `/usr/share/doc/`⁸, but a convenience symlink can be found at `/opt/opencpi/documentation.html`. This path should not change across multiple OpenCPI versions and can be passed directly to most browsers, *e.g.* “`xdg-open /opt/opencpi/documentation.html`”. If you receive the RPM directly from the AV team, it may include BSP documentation that is not available on GitHub.

H-I.8 Zynq Parts Defined

All Zynq-based Xilinx parts that are supported by the current Framework, but not necessarily used in an OpenCPI-supported platform, have been defined within the Framework build system. This means that an end user creating a BSP for a “larger fabric” Zynq-7000, *e.g.* `xc7z100`, will not need to change any Framework files and have all required changes isolated to a single Project. This does *not* include additional platforms that have different internal configurations, *e.g.* Zynq-UltraScale+ devices.

H-II Changes and Notes

H-II.1 I/Q Data Ordering

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.*:

$$\begin{aligned} I_0 &= \text{InPort}_0[15 : 0]; Q_0 = \text{InPort}_0[31 : 16] \\ I_1 &= \text{InPort}_1[15 : 0]; Q_1 = \text{InPort}_1[31 : 16] \end{aligned}$$

The following Application and Workers were identified as improperly handling `iqstream_protocol` complex data (I/Q) presented in the manner noted above and have been corrected:

- `assets/components/dsp_comps`:

<code>cic_dec.hdl</code>	<code>dc_offset_filter.hdl</code>	<code>pr_cordic.hdl</code>
<code>cic_int.hdl</code>	<code>fir_complex_sse.hdl</code>	<code>rp_cordic.hdl</code>
<code>complex_mixer.hdl</code> ⁹	<code>iq_imbalance_fixer.hdl</code>	
<code>complex_mixer.rcc</code> ⁹	<code>phase_to_amp_cordic.hdl</code>	

- `assets/hdl/devices`:

<code>ad9361_dac.hdl</code>	<code>lime_dac.hdl</code>	<code>lime_dac_ts.hdl</code>
-----------------------------	---------------------------	------------------------------

- `assets/applications`

⁸The Fedora/Red Hat standard documentation location.

⁹See last paragraph as well concerning Property changes.

FSK

Rx_App

It is important to note that this list includes various DACs as well as the `complex_mixer`, meaning that **user's RF Applications from previous versions of the Framework *may* need to be updated**, *e.g.* to handle this change in the DAC's interpretation of data presented for transmission. Also related to this, the `phs_inc` Property of `complex_mixer` may require inversion when compared to previous versions.

H-II.2 Upgrade Notes

If you have previous versions of the RPM-based OpenCPI RPMs installed, they must all be manually removed, including the prerequisites. **There is no automatic upgrade path.** Manually performing “`sudo rm -rf /opt/opencpi/prerequisites`” after uninstalling **all** previous RPMs *may* also be needed.

H-II.3 Significant SD Card Layout Changes

There have been significant changes to the layout of the Framework and its support files. These are mostly implementation details that the end user does not need to be concerned with. However, scripts and paths that are used for embedded development (both locally and using NFS mounts) have changed enough that users should wipe any SD cards they have used and follow the directions given in the respective platform's *Getting Started Guide* to create a new one.

H-II.4 Intel Quartus and Quartus Pro Version Change

This release has been tested with Quartus Standard 17.1 which is a newly supported version of the tool. Version 15.1 should still work, but future versions will be tested with 17.1. Additionally, preliminary support for Quartus Pro 17.0.2 is also available within the Framework, but is not used by any platforms at this time.

H-II.5 New Input Port ZLM-Handling Primitive

For an HDL Worker to handle the *de facto* end-of-stream signaling used by the Component Unit Test infrastructure (a ZLM with OpCode 0), a convenience primitive is now available and found at `projects/core/hdl/primitives/util/zlm_detector.vhd` with an example implementation shown in the `capture_v2.hdl` Worker.

H-II.6 Change in QDAC/QDAC_TS Component Spec

The QDACs now have two ports. The “IN” ports have not changed, but some applications may need to be updated if they did not explicitly connect to them. See H-I.4 above for more information.

H-II.7 Change in ocpidev Default Output and Scope

Some of the output from `ocpidev's show` has changed to default to being more “human friendly” and reporting in a tabular fashion (`--table`). Many of these commands retain a `--simple` parameter for quick parsing. Most also support `--json`, which is the recommended and most stable method of parsing `ocpidev` output.

Some of `ocpidev's show` commands now have the concept of “scope” where the user is requesting a list of *somethings* provided by a single project, or *all somethings* available within all registered Projects. At this time, no commands support both of the scoping arguments `--global-scope` and `--local-scope`. However, to ensure future `ocpidev` enhancements remain compatible with user scripts, some commands will force the user to explicitly specify a scope if the “reasonable” not-yet-implemented default is not yet ready. For example, in a future release, “`ocpidev show tests`” will report *all* Unit Tests that are available. That report is not yet available, so the user *must* currently explicitly scope the request with “`ocpidev show tests --local-scope`”. However, something like “`ocpidev show registry --local-scope`” would make no sense, so will never be supported.

H-II.8 Rounding and Truncation in qdac-spec Workers

Various DACs were found to be improperly casting *signed* data to *unsigned* during certain operations, resulting in an overflow when the maximum value was transmitted.

H-II.9 HDL Workers Failing Unit Tests That Previously Passed

If a user experiences Unit Test failures that were not present in previous releases, the most likely cause is that “backpressure” is now automatically asserted by default; see H-I.2 for details.

H-II.10 New Framework Build System

The OSS OpenCPI Project has migrated the Framework build system to a new custom-made system that produces RPMs directly¹⁰ and removes the need for “prerequisite” RPMs. If upgrading from a previous RPM-based installation, the prerequisite RPMs might require manual deletion. Any mention of prerequisite RPMs in the Appendices (*e.g.* L-II.6) below can likely be ignored.

H-II.11 New RCC Platform Configuration

Prior to 1.4, RCC Platforms were defined in `/projects/core/rcc/platforms`. With 1.4, they can be in the `/rcc/platforms` directory of any Project. *The Platform Development Guide* defines how RCC Platforms are *now* defined and a condensed migration guide can be found at `projects/core/rcc/platforms/README.migrating`.

H-III Major Known Deficiencies

H-III.1 ZedBoard IP Required Even In Standalone Mode

Applications (including XML-only ones) fail if there is not an IP address assigned to the ZedBoard, even when in “standalone mode.” If the network connection is not used, forcing a MAC (cf. Section L-III.2) and IP addresses is required. To set a temporary IP address, the command “`ifconfig eth0 192.168.244.244`” can be used. This problem was found late within the 1.4 release cycle and should be addressed with the next major release.

H-III.2 Connection Direction In GUI Assumes XML order

When importing hand-written XML into the GUI, the `to` and `from` attributes are ignored on connections. This does not affect any XML generated by the GUI.

H-III.3 OnlyPlatforms/ExcludePlatforms Does Not Work For Individual Unit Test Cases

You can include and exclude platforms at a full-test level, but individual cases currently ignore the directives.

H-III.4 Some HDL Workers Will Need To Be Updated When `props.in.raw.address` Width Fixed

The “raw” address scheme for HDL workers is documented as a 32-bit bus. In testing, it was found that this is currently implemented as only 16 bits. While this has *not* yet been fixed, users should review their own libraries for any Workers that make assumptions concerning the size of the bus. Any `raw.address` access should be written in more generic VHDL, *e.g.*:

```
-      event_rawprops.raw.address <= x"0040"; --TXRF modules enable register
+      event_rawprops.raw.address <= (6 => '1', others => '0'); --TXRF modules enable register (0x40)
```

H-III.5 Default python Must Be Python2

According to the PEP-394 specification, system-level `python` *should* point to a 2.x version of the Python interpreter. Some scripts within the Framework and Unit Tests *require* this configuration.

H-III.6 Some `ocpi.assets` Workers Have Flow Control Issues

The Component Unit Test infrastructure has been enhanced (cf. Section H-I.2) and deficiencies within some Workers have been revealed during regression testing. The following `ocpi.assets` Workers are currently known to have defects which will be fixed in a future release:

¹⁰If a user was building custom RPMs (from the `releng` directory), the process has changed considerably.

`bias_ver.hdl``cic_int.hdl``fir_real_sse.hdl``cic_dec.hdl``fir_complex_sse.hdl`

H-III.7 Previously Documented Deficiencies

The following deficiencies are still present in 1.4:

- 32-bit Architecture File Size Limit (cf. M-II)
- OpenCPI HDL Assembly Filename Constraint (cf. M-I)
- Maximum Message Length in PCI-Based Platforms (cf. L-III.1)
- Properties Before Raw Properties Cannot Have Parameterized Lengths (cf. K-III.7)
- MAC Address on ZedBoard when Using the xilinx13.3 Software Platform (use xilinx13.4 to avoid the issue) (cf. L-III.2)
- PCI-Based Platforms Not Always Detected in Remote Tests (cf. K-III.5)
- Lime RX/TX Proxy Tests on CentOS6 (cf. K-III.10)
- PCI Platforms Can Only Transmit Using One Channel (cf. I-III.1)

The following previously-documented deficiencies have been fixed in this release:

- XML Parser Mixed-Case Issue (cf. K-III.8)
- Platform Development Guide (cf. L-III.3)

I Release 1.3.1

I-I Enhancements

- All documentation has now been updated to cover 1.3 capabilities and features
- Bugfixes and enhancements for AD9361
- Workarounds for bug noted in Section I-III.1
- USB Blaster `udev` rules fixed
- `ocpizynq` tool fixed

I-II Changes and Notes

None to report.

I-III Major Known Deficiencies

- *Note:* no other deficiencies from Section J-III have been mitigated and all remain.

I-III.1 PCI Platforms Can Only Transmit Using One Channel

The current implementation of the PCI-e Specification (ALST4 and ML605 platforms) only allows DMA transaction from the host to the platform on the *first External Port defined in the Assembly or Container XML*. This means the *input* to any Assembly/Container should be listed first. See `projects/assets/hdl/assemblies/empty/cnt_1rx_1tx_bypassasm_fmcomms_2_3_hpc_LVDS_ml605.xml` as an example. *This does not affect any Zynq-based platforms.* This problem was found late within the 1.3 release cycle and should be addressed with the next major release.

J Release 1.3

J-I Enhancements

J-I.1 OSS Release

OpenCPI is now available on GitHub under the OpenCPI Organization. The first release is in a git branch, but full integration is planned in the future. This includes all of the previously-proprietary “ocpiassets / Assets” Project, including the Board Support Package (BSP) for Epiq Solutions’ Matchstiq Z1.

J-I.2 Python and GNU Radio Integration (Technology Preview)

There are now SWIG bindings available to allow a top-level Python script to use the ACI to manipulate an Application in the same manner as from C++¹¹. One use case is a GNU Radio block that allows GNU Radio Companion to handle the entire lifecycle of an OpenCPI Application. This block can handle data flows of complex I/Q streams or any primitive data type (*e.g.* floats or bytes) in and out of the Application. A reference implementation of this block is provided in OpenCPI’s fork of GNU Radio Companion.

This fork of GNU Radio Companion also allows for Application and HDL Assembly building, as well as deployment independent of the AV GUI. Some of this capability was presented at the GNU Radio Conference GRCon17; the video is available on YouTube.

J-I.3 AD9361 Support

The popular Analog Devices, Inc’s AD9361 RF transceiver IC is now supported via a series of Device Workers. The modularity of these workers eases AD9361 transceiver integration with future FPGA systems. Most of the major AD9361 features are supported, including RX/TX RF digitization/data streaming and RF tuning/gain control. The AD9361 Device Workers currently only fully support some operating modes, but contain infrastructure in anticipation of future expansion. These device workers utilize the vendor’s No-OS software library for high-level control.

There is also “card” support for the FMCOMMS2 and FMCOMMS3 evaluation/prototyping RF transceiver FMC cards centered on the AD9361. The newly released `ocpi.assets` Project includes a single-channel RX narrowband I/Q stream to file (`applications/rx_app`) and single-channel RX/TX FSK modem (`applications/FSK`) reference applications, which each support the FMCOMMS2 and FMCOMMS3 card on the zed platform.

J-I.4 GUI Improvements

Since 1.2, the GUI has multiple improvements, some of which are:

- **The ability to build OpenCPI Assets** - The Component and Application developer can now build without the command line. The build tool allows RCC and HDL compilation from numerous levels in a Project from individual assets to the entire Project. The tool also allows the developer to build and run Component unit tests while allowing multiple concurrent builds in separated tabbed consoles.
- **New Perspective** - This Eclipse Perspective is the new presentation of the build tool. It consists of a number of views where three are specific to OpenCPI.
- **Project Explorer view** - This view allows simple navigation to select items to build and provides a way forward to provide OpenCPI features without having to explore the file system.
- **OpenCPI Operations Panel** - Selections can be placed in this panel and built for all available RCC and HDL Targets and/or Platforms.
- **Build Status View** - Provides a simple graphical presentation of the launched build or test run and its status (*e.g.* running). The user may also re-run build, clean, or run executions from this view and bring the build console window forward.
- **New HDL Platform Editor** - The HDL developer can now create and view HDL platform XML in a graphical presentation.

¹¹This integration is only available on CentOS 7 platforms.

J-I.5 Project Registry

Release 1.3 introduces a new concept of “Project Registry” to organize and interrelate a set of Projects which allows Projects to be moved (cf. J-I.6) as well as present an explicit relationship between them, *e.g.* “*this* copy of `ocpi.assets` was built based upon *that* copy of `ocpi.core`.” Registry manipulation is performed using `ocpidev` (cf. J-I.7).

The default configuration is that every Project on a machine shares a single Registry (located at `/opt/opencpi/project-registry`). A user must be in the Unix group “`opencpi`” to manage this global Registry.

This new capability deprecates the previously-used `OCPI_PROJECT_PATH` build-time variable as well as the scripts `freeze_project.sh` and `thaw_project.sh` noted in K-III.6.

J-I.6 Project Migration

With the new Project Registry (cf. J-I.5), Projects can now be moved to new locations or archived and then restored on another machine with the use of the `ocpidev unregister` and `ocpidev register` commands. This mitigates the previously-documented deficiency found in K-III.6.

J-I.7 ocpidev Improvements

The “`ocpidev`” command has been improved in various ways, including:

- `bash` completion for many commands (try `ocpidev <tab><tab>`)
- Ability to show possible HDL platforms: `ocpidev show hdl platforms --table`
- Ability to show possible RCC platforms: `ocpidev show rcc platforms`
- Ability to build unit tests
- Project Registry (J-I.5) manipulations

J-I.8 RCC Platforms

With the new Project Registry (cf. J-I.5), RCC Platforms can now be defined exclusively within a Project outside of the CDK / Core Project. An example of this can be seen in the `inactive` project under `rcc/platforms/macos10_13`. This improvement allows BSPs to be self-contained within a Project.

J-I.9 Remote Containers (Technology Preview)

The full-OSS OpenCPI code base has imported previously-written code that allows Remote Containers.

J-I.10 OpenCL Containers (Technology Preview)

The full-OSS OpenCPI code base has imported previously-written code that allows OpenCL (“OCL model”) Containers on CPUs and GPUs.

J-II Changes and Notes

This section describes items that may cause users of previous versions to modify their code, build system, etc.

J-II.1 Major Project Restructuring

Starting with Release 1.3, the CDK is no longer a top-level Project. All Components that were previously found in the CDK, *e.g.* `file_read`, have migrated to `ocpi.core`. The previously-documented “Base Project” has been removed; it was the RPM-specific manner of separating the CDK from the core components needed to run OpenCPI. The various items (cards, devices, specs, etc) that were previously found in the Base Project have been distributed into one of three new Projects. The previously non-OSS proprietary items have been redistributed within the new Projects as well. The Projects are:

- `ocpi.core` - Core items for the Framework
- `ocpi.assets` - Items fully-supported by the OpenCPI Team

- `ocpi.inactive` - “Everything else”

These items being moved will cause Application XML to possibly fail, *e.g.* anything calling for the `ocpi.file_read` spec now needs to look for `ocpi.core.file_read`. There is a similar rename of `ocpiassets` $\Rightarrow$ `ocpi.assets`.

There is a provided utility script (`/opt/opencpi/cdk/scripts/upgradeApp_v1_3.py`) to assist in this migration. This script will modify the package information for the specs required. There are three ways it can be called:

- If given a single argument (the Application XML file name), it will write the same name with `updated_` prepended.
- If given a second argument, the modified file will be written to that file name.
- If the second argument is `same`, the original file will be overwritten.

This change also removes the `/opt/opencpi/base_project_source/new_base_project.sh` script found in earlier releases; its equivalent is now `/opt/opencpi/projects/new_project_source`, which uses the same arguments as noted in the *Getting Started Guide*.

J-II.2 Upgrade Notes

If you have previous versions of the RPM-based OpenCPI RPMs installed, they must all be manually removed, including the prerequisites. **There is no automatic upgrade path.** Manually performing “`sudo rm -rf /opt/opencpi/prerequisites`” after uninstalling **all** previous RPMs *may* also be needed.

J-II.3 Prerequisite RPMs Built With Newer Language Standards

All prerequisites are now built with the C++11 or the C98 standard implemented. This will increase interoperability and avoid ambiguous situations, *e.g.* the size of a `bool` variable.

J-II.4 Legacy Platform Names Continue to be Deprecated

The legacy suffix “`_pf`” for simulator platforms, *e.g.* `isim_pf`, is not supported when running unit tests. Unit tests were introduced after the suffix was deprecated, so the effect on end users should be minimal.

J-II.5 OCPI_XILINX_VIVADO_DIR Variable Changed

A bug in Release 1.2 required the variable to be set “deep”, *e.g.* `/data/Xilinx/Vivado/2017.1`. You *must* now set it to the top-level value, *e.g.* `/data/Xilinx`. If a specific version of Vivado is required, `OCPI_XILINX_VIVADO_VERSION` should be set, *e.g.* `2017.01`. As a reminder, these variables only apply if you have both ISE and Vivado installed in different base directories, *e.g.* ISE in `/opt/Xilinx` vs. Vivado in `/data/Xilinx`.

J-II.6 `ocpidev` and Requesting RCC Platform To Match HDL Platform

When building Projects with `ocpidev`, the user can tell the build system to automatically select the proper RCC Platform when given an HDL Platform. This option is `--rcc-hdl-platform` with a mnemonic of “build *this* RCC Platform for *that* HDL Platform.” Previous Releases had the flags backwards (“`--build-hdl-rcc-platform`”). This is incorrect and has been deprecated.

J-II.7 `zero_padding` Component in `ocpi.assets` deprecated

In the FSK reference application, `zero_padding.hdl` was replaced by two new Workers, `zero_pad.hdl` and `mfsk_mapper.hdl`. The change was made because `zero_padding.hdl` performed two functions: binary FSK mapping and inserting zeros between symbols. Whenever possible, functions should be separated into their own Components for increased flexibility and reuse. The `zero_padding` Component will be removed in Release 2.0.

J-III Major Known Deficiencies

J-III.1 Previously Documented Deficiencies

The following deficiencies are still present in 1.3:

- 32-bit Architecture File Size Limit (cf. M-II)
- OpenCPI HDL Assembly Filename Constraint (cf. M-I)
- XML Parser Mixed-Case Issue (cf. K-III.8)
- Maximum Message Length in PCI-Based Platforms (cf. L-III.1)
- Platform Development Guide (cf. L-III.3)
- Properties Before Raw Properties Cannot Have Parameterized Lengths (cf. K-III.7)
- MAC Address on ZedBoard when Using the xilinx13_3 Software Platform (use xilinx13_4 to avoid the issue) (cf. L-III.2)
- PCI-Based Platforms Not Always Detected in Remote Tests (cf. K-III.5)
- Lime RX/TX Proxy Tests on CentOS6 (cf. K-III.10)

The following previously-documented deficiencies have been fixed in this release:

- Driver Failing to Allocate Memory on Zynq-Based Platforms (cf. K-III.1)
- Built Projects Cannot Be Moved (cf. K-III.6)
- HDL Workers Cannot Send SOM Without Valid Data (cf. K-III.2)
- File_Read and File_Write Cannot Handle ZLM Mid-file (cf. K-III.4)
- SDP Cannot Handle More Than One ZLM Per Application Run (cf. K-III.3)

The following previously-documented deficiencies will *not* be fixed:

- Simulation of Primitive Cores Needs **Makefile** Modified (cf. K-III.9): It was determined that this is not a bug. For primitive cores, if a user sets the **SourceFiles** variable, they must explicitly list all files, including ***_pkg.vhd** files. This is true for *all* platforms.

K Release 1.2

K-I Enhancements

K-I.1 Vivado Support for Applicable Xilinx-Based Platforms

By far the #1 requested feature by OpenCPI users, the Framework now supports the use of Xilinx’s Vivado on the appropriate Platforms. **This toolchain is now the default** where applicable (mostly Zynq-based), and it is highly recommended that users of Xilinx-based platforms review the provided *Vivado Usage Guide*.

End users should be aware (see *Vivado Usage Guide* for more details) that:

- If you simulate with ModelSim, you may need to upgrade. Vivado requires a minimum version of 10.5c. Testing has been done using 10.6a.
- If your design requires the use of Xilinx CORE Generator IP (**coregen**) items, they must be regenerated with Vivado to be *simulated* in XSIM. Your existing **ngc** files from ISE can continue to be used for *synthesis*, but porting to new *Vivado IP* is recommended.
- There is a new platform, **zed_ise**, that is intended for testing purposes, but released with the distribution. This platform is the ZedBoard using the Xilinx ISE tool chain.

K-I.2 New command option “ocpidev build”

To help improve the end-user experience, OpenCPI 1.2 requires less command-line usage of “**make**”. The previously-available tool **ocpidev** now has a “**build**” verb available. More information can be found in the (OSS) *Component* and *Application Development Guides*.

K-I.3 GUI Now Delivered as RPM

There is now an **opencpi-ide** RPM included in the distribution. A separate installation of Eclipse for each end user is no longer required. Once installed, any user can launch with the **ocpigui** command.

K-I.4 Artifact Version Checking

While backwards compatibility is a major goal of OpenCPI, binary compatibility across versions is *not* guaranteed. Every artifact should always be rebuilt with the latest framework. To enforce this requirement, the framework will check all artifact metadata at deployment time to verify the version used to build. Version 1.1 inserted the proper metadata, but the comparison is new in 1.2, allowing 1.2 to recognize and reject 1.1 artifacts. Artifacts rejected for no other reason will issue a warning message at all debugging levels. This behavior can be overridden by setting the environment variable **OCPI_ALLOW_VERSION_MISMATCH** to “1”; *this is not recommended*.

K-I.5 Miscellaneous Improvements

A quick list of improvements not documented above include:

- Cross-platform ACI builds using **ocpidev**-generated applications previously failed
- Picoflexor T6A Software updated to vendor’s Drop 5.5.2
- **ocpiassets** project can now be imported into GUI
- Proxies can now use string properties¹²

K-II Changes and Notes

This section describes items that may cause users of previous versions to modify their code, build system, etc.

K-II.1 rx-spec and tx-spec Support Property Interdependence

Certain front-ends have different ranges based other properties, *e.g.* the current sample rate, requiring some properties that were previously “Initial” (immutable) to have their restrictions relaxed. *This compatibility break should only affect platform developers*, unless applications were incorrectly caching these values. All interfaces above that level should be unaffected.

¹²This is noted here because it is one of the first user reported bugs.

K-II.2 Vivado Support

Please see K-I.1 for modifications that *may* need to be performed.

K-III Major Known Deficiencies

K-III.1 Driver Failing to Allocate Memory on Zynq-Based Platforms

In some situations after repeated multi-model (RCC and HDL) application runs on Zynq-based platforms (e.g. ZedBoard), the kernel driver will fail to allocate contiguous memory for DMA data transfers. On x86-based platforms, memory for this use is allocated at boot time before memory fragmentation. Currently, no memory is being allocated at boot time for the Zynq-based platforms. Single ocpirun or ACI application runs from the command line do not present any errors. When running an app in a long loop, such as in the unit test framework, or an application that writes a large amount of data (approx. 1GB) to file, this error *may* occur.

K-III.2 HDL Workers Cannot Send SOM Without Valid Data

Although the HDL Component documentation claims the Framework's internal protocols support a separate Start-of-Message independent of valid data designations, this sequence causes the Component's output port to ignore any remaining data. This was found late within the 1.2 release cycle and should be addressed with the next release.

K-III.3 SDP Cannot Handle More Than One ZLM Per Application Run

The SDP transport will not propagate a second ZLM, so at this time the *only* ZLM should be an implied end-of-application transmission. This problem does not exist when communicating between RCC-only Workers. Related to, but independent of K-III.4. (See also M-VI.)

K-III.4 File_Read and File_Write Cannot Handle ZLM Mid-file

`file_read` (HDL implementation) and `file_write` (RCC and HDL implementations) will stop if a Zero-Length-Message (ZLM) is found when operating in "message mode" regardless of the location within the file. Related to, but independent of K-III.3. (See also M-VI.)

K-III.5 PCI-Based Platforms Not Always Detected in Remote Tests

Using `OCPI_REMOTE_TEST_SYSTEMS`, JTAG connection problems have been observed, possibly related to system reboots. They seem to be resolved after disconnecting and reconnecting the USB cable.

K-III.6 Built Projects Cannot Be Moved

When a Project is built, the dependencies of each artifact and other support files are captured. Some of these contain absolute paths pointing to where the Project is currently located. The distribution includes a pair of *unsupported* scripts that *may* allow movement of Projects. They can be found in `/opt/opencpi/cdk/scripts/` and are named `freeze_project.sh` and `thaw_project.sh`.

K-III.7 Properties Before Raw Properties Cannot Have Parameterized Lengths

"Raw properties" start at a certain property with the properties before that being "non-raw" properties. Currently, none of these "before" properties can have their length be set by an expression dependent on another parameter property.

K-III.8 XML Parser Mixed-Case Issue

The XML parser does not always recognize tags if differently-cased versions of the same tag are in the same XML file, e.g. an OAS having both `<Instance .../>` and `<instance .../>` XML tags will fail. Users of the GUI should not be affected.

K-III.9 Simulation of Primitive Cores Needs Makefile Modified

When designing a Primitive vendor-specific core that is intended for simulation, the required `*_pkg` file containing the core's stub must be explicitly listed first in the Makefile's `SourceFiles` variable.

K-III.10 Lime RX/TX Proxy Tests on CentOS6

The proxy tests for the Lime transceiver use a feature of C++11 that is not supported by the compiler in CentOS 6. The workaround is to use similar code to the following in the various places `to_string()` was used:

```
-      std::string ClkInStr;  
-      ClkInStr = to_string(2 * sample_rate * 1e6).c_str();  
+      std::ostringstream ostr_clkkin;  
+      ostr_clkkin << (2 * sample_rate * 1e6);  
+      std::string ClkInStr(ostr_clkkin.str());
```

K-III.11 Previously Documented Deficiencies

The following deficiencies are still present in 1.2:

- 32-bit Architecture File Size Limit (cf. M-II)
- OpenCPI HDL Assembly Filename Constraint (cf. M-I)
- MAC Address on ZedBoard when Using the xilinx13_3 Software Platform (use xilinx13_4 to avoid the issue) (cf. L-III.2)
- Maximum Message Length in PCI-Based Platforms (cf. L-III.1)
- Platform Development Guide (cf. L-III.3)

K-IV Addenda to Previous Release Notes

- The default `CWD_MAX_LENGTH` noted in L-I.3 is now set to 256.
- The Liquid RPMs referenced in L-II.6 are now **required** to build the `ocpiassets` project (`complex_mixer.rcc`). They are still considered optional when installing.

L Release 1.1

L-I Enhancements

A quick list of improvements include:

- Simulatable Data Plane: Allows HW and SW co-simulation to test integration between heterogeneous Workers
- Scalable Data Plane: Allows Platform Workers to more efficiently use wider data paths on future platforms, e.g. a 128-bit bus instead of 32-bit¹³
- New automatic test harness creation, including limited remote platform support¹⁴
- Automatic bus width adaption between HDL Workers
- Additional Assets components to help design and debug Applications
- HDL Simulator on-the-fly launching¹⁵
- Extended GUI capabilities
- Official CentOS 7 driver support
- Split platform names more explicitly between hardware and software platform types
- Many fields in XML that require numerical values can now have mathematical calculations
- Added platform support (Picoflexor RCC-Only)

L-I.1 Debug Symbols for Framework Now Available

An RPM containing most debug symbols for the Framework (`opencpi-debuginfo-*.x86_64.rpm`) is now provided. This greatly enhances the debugging of RCC Workers using industry-standard debuggers such as `gdb` and `ddd`.

L-I.2 Kernel Driver Memory Arguments

In previous versions, the “`memmap`” argument was given to the Linux kernel command line. This can conflict with other software that may also require reserved memory. If the user requires the reservation of more than one memory area using the `memmap` option, the OpenCPI driver will:

- Assume the first `memmap` is for its use
- Explicitly use the range/subrange marked with an additional `opencpi_memmap`. *The original `memmap` must still be provided* for the kernel to make the reservation.

See `/opt/opencpi/driver/README` for more information, including usage examples and configuration.

L-I.3 HDL File_Read and File_Write CWD

The HDL Workers implementing the `File_Read` and `File_Write` specs distributed in 1.0 failed if used in a directory that exceeded 100 characters deep. There is an additional parameter `CWD_MAX_LENGTH` on these Workers that can be increased to allow deeper directories when needed.

L-II Changes and Notes

This section describes items that may cause users of previous versions to modify their code, build system, etc.

¹³Currently limited to some Zynq-based platforms.

¹⁴Most `ocpiassets` have been moved to the new test system to provide examples.

¹⁵See Section L-II.1 for more information.

L-II.1 Simulators Renamed and Work Out-of-the-Box

The simulator platform names have been renamed to remove the “_pf” suffix, *e.g.* `isim`. The `isim`¹⁶ and `modelsim` simulation platforms can now be targeted once the appropriate environment variables are set, *e.g.* `OCPI.MODELSIM_DIR`. With `ocpirun` able to launch simulations, the usage of “`ocpihdl simulate`” is now deprecated and does nothing.

An example:

```
$ ocpirun -C
Available containers:
#  Model Platform      OS      OS-Version  Arch      Name
0  hdl   isim              linux    c7          x86_64    lsim:isim
1  hdl   modelsim          linux    c7          x86_64    lsim:modelsim
2  rcc   centos7           linux    c7          x86_64    rcc0
```

L-II.2 HDL Width Adapters Not Required

One of the advantages of the new *Scalable Data Plane* is the Framework’s automatic insertion of “width adapters” whenever needed between Workers in an Assembly. This means the user can remove any `wsi_16_to_32`, `wsi_32_to_16`, or `wsi_width_adapter` Components that are instantiated in their Assemblies or Applications. It is harmless to leave them in existing designs, but future Applications should not instantiate them.

L-II.3 Platform Name Changes

To allow for more hardware and software separation in the future, there is now more distinction used in platform naming. Previously, the keyword “zed” was used to describe both software (the Xilinx-based Linux kernel) and hardware (the ZedBoard). This ambiguity led to confusion for some users and implied a nonexistent reliance on the ZedBoard. Starting with OpenCPI 1.1, the software platform for most Zynq-based hardware is labeled “`xilinx13_3`”, analogous to “`centos7`”. The “13.3” is shorthand for Xilinx’s Linux release from the Third Quarter (Q3) of 2013. This means that previous code that used an `OCPI.TARGET_PLATFORM` of `zed` must now be changed to `xilinx13.3`.

L-II.4 RCC Cross-compilation No Longer Uses OCPI.CROSS_HOST

Previous OpenCPI releases used `OCPI.CROSS_HOST` to define RCC cross-compilation. This required the usage of less-than-memorable values such as `arm-xilinx-linux-gnueabi`. The new manner is to use `OCPI.TARGET_PLATFORM`, `RccPlatforms`, or `RccHdlPlatforms`, which allow more intuitive and consistent values (See Section L-II.3). For example, the example above could be replaced with “`OCPI.TARGET_PLATFORM=xilinx13.3`”, “`RccPlatforms=xilinx13.3`”, or “`RccHdlPlatforms=zed`”.

The `RccPlatforms` and `RccHdlPlatforms` variables allow the targeting of multiple platforms with a single `make` call. While either format can be used for Workers, ACI Applications can only be built for a single target platform using the `OCPI.TARGET_PLATFORM` variable.

L-II.5 Updating a 1.0.x Unit Test for the 1.1 Framework

Some unit tests for Components were released with the 1.0.x version of `ocpiassets` that relied upon simulator availability or certain platform names reported by `ocpirun` (see Sections L-II.1 and L-II.3). If any users have developed their own unit tests based on these, some updates may be required for them to work using the new release.

All of the provided unit tests have been updated to work with the new release and can be used for reference. If the previous unit test is using `ocpirun`, the `Makefile` needs to be updated to add the following to any lines that are forming the execution string:

```
-P $@=$P
```

This is forcing the platform of your unit test to be set explicitly with `ocpirun`’s `-P` option.

If the unit test is using an ACI Application to run the unit test, the platform needs to be set similarly within the C++ source. The Platform will be passed from the `Makefile` to the ACI Application as a command-line argument

¹⁶Only version 14.7 without rebuilding the Base Project.

by adding `$P` to the end of any execution strings. The source code needs to be updated to use this new argument by adding the following to the C++ application, while changing “XXX” to your Worker’s name and “2” to the correct argument number based on your current inputs:

```
std::string pf("XXX=");
pf += argv[2];
OA::PValue params[] = {
    ...
    OA::PVString("platform", pf.c_str()),
    ...
    OA::PVEnd};
...
OA::Application app(appNameXml, params);
```

This patch is adding the platform that is passed in (via `argv[2]`) to the `params` array that is used to constrain the application’s search space.

The user could do a full conversion to the new testing framework released with 1.1, but this is expected to be more work.

L-II.6 Prerequisite RPM Changes

The Open Source XZ Utils has replaced the LZMA Utils. This has caused the provided RPM to change names. Additionally, new RPMs are now provided for other prerequisite packages; consult the *Installation Guide* for a list. An upgrade to the new RPMs *should* cause previous versions to be removed, but the recommended procedure is to uninstall all 1.0.x RPMs before installing 1.1.

For the first time, OpenCPI is providing a pre-packaged third-party library that is not directly required by the Framework: `liquid-dsp`. This is provided to assist end users in the creation of DSP-heavy RCC Workers that execute on all currently-supported platforms without requiring any additional configuration for cross-compilation. It is treated like any other prerequisite library; usage information can be found in the *OpenCPI RCC Development Guide*.

L-II.7 Deprecated RCC controlOperations

The RCC `controlOperations` `afterConfigure` and `beforeQuery` have been deprecated and removed from the IDE. They were never called by the Framework and the C++ interface provides the finer-granularity `XXX_read()` and `XXX_written()` notification methods.

L-II.8 64-Bit Build Host Issues (“Value too large for defined data type”)

Some 32-bit cross-compilers exhibit somewhat-rare problems on 64-bit host machines using file systems that allow 64-bit inodes, *e.g.* XFS. A new RPM, `ocpi-rereq-build-support-inode64`, provides a workaround. Additional information can be found in the Picoflexor T6A documentation.

L-III Major Known Deficiencies

This is a list of major known deficiencies at the release of 1.1 that were not previously documented in Section M. This is *not* an exhaustive list.

L-III.1 Maximum Message Length in PCI-Based Platforms

PCI-based platforms, *e.g.* ML605 and ALST4, are limited to a 32KB buffer. By default, that buffer is split into two, resulting in two 16KB buffers. The maximum message length, in bytes, is then 16KB-16¹⁷. RCC Workers will see any data beyond the limit as zero-filled.

¹⁷16 bytes required for metadata

L-III.2 MAC Address on ZedBoard when Using the xilinx13_3 Software Platform (use xilinx13_4 to avoid the issue)

The currently used Linux release (“xilinx13_3”) cannot properly set the MAC address on the ZedBoard. This should only affect users who have *more than one ZedBoard* on a single network. Consult the *ZedBoard Getting Started Guide* for a workaround.

L-III.3 Platform Development Guide

While all documentation is constantly under improvement, the *Platform Development Guide* in particular is currently undergoing major changes, in part due to the SDP improvements. If an end user intends to create a Board Support Package (BSP) for a new target platform, they should contact the OpenCPI Team to ensure the latest Guide is used.

M Release 1.0.1

These are “Release Notes” from Version 1.0.1 that are still relevant as of Version 1.1.

M-I OpenCPI HDL Assembly Filename Constraint

Due to a limitation with the ISE Simulator (isim), the filename of any HDL Assembly must be all lowercase letters. A capital letter anywhere in the filename will result in an error.

M-II 32-bit Architecture File Size Limit

On some software platforms, there will be an error when writing files greater than 2^{31} bytes (about 2.2 GB), especially if using NFS. This includes the ARM 32-bit architecture on Zynq-based platforms.

M-III Eclipse Project Description File

A necessary file known as the project description file (`.project`) is generated when creating a Project in the AV GUI. This file is required by Eclipse and is used to describe the project in its workspace. Removal of this file will cause an error.

M-IV Refreshing the AV GUI

Whenever files are being created or removed via command-line for an Project, the Project within the IDE needs to be refreshed. Examples include executing `make clean`, `make tests`, `make rcc`, etc. Many of the `make` commands will create or remove files in the Project, so it is important to refresh the Project within the IDE to reflect the changes.

M-V Non-default Xilinx Installation Locations

OpenCPI assumes Xilinx’s installation location is `/opt/`. If the Xilinx tools were installed in another location, the cross compiling environment variable can not be inferred correctly. To resolve this issue, refer to the `/opt/opencpi/cdk/env/xilinx.sh.example` file and the *Installation Guide*.

M-VI Zero-Length Message Finite State Machines

Some workers make use of a finite state machine to monitor a zero-length message (*ZLM*) from an upstream worker. This implementation was designed specifically with unit tests in mind so that simulations could be ended with an end-of-file (EOF) condition from the File Read Worker rather than waiting for a fixed amount of time. Simulation times vary across machines and Workers, so predicting how much time a Worker might need on various machines is not deterministic. In the future, the handling of zero-length message may change, but the current finite state machines for handling them are provided to complete the unit tests of the workers.