



OpenCAPI 3.0 Certified Test Resources Engineering Note

Version 1.0

20 May 2020—OpenCAPI Confidential

Approved

Approved for Distribution to OpenCAPI Members Only

Approved

OpenCAPI 3.0 Certified Test Resources Engineering Note

OpenCAPI Compliance Work Group
OpenCAPI Consortium

Version 1.0 (20 May 2020)

Copyright © OpenCAPI Consortium 2020.

All capitalized terms in the following text have the meanings assigned to them in the OpenCAPI Intellectual Property Rights Policy (the “OpenCAPI IPR Policy”). The full Policy may be found at the OpenCAPI Consortium website.

Use of this document is controlled by the OpenCAPI IPR Policy.

This document and the information contained herein is provided on an "AS IS" basis AND TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, THE OPENCAPI CONSORTIUM AS WELL AS THE AUTHORS AND DEVELOPERS OF THIS DRAFT STANDARD OR OTHER DOCUMENT HEREBY DISCLAIM ALL OTHER WARRANTIES AND CONDITIONS, EITHER EXPRESS, IMPLIED OR STATUTORY, INCLUDING BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES, DUTIES OR CONDITIONS OF MERCHANTABILITY, OF FITNESS FOR A PARTICULAR PURPOSE, OF ACCURACY OR COMPLETENESS OF RESPONSES, OF RESULTS, OF WORKMANLIKE EFFORT, OF LACK OF VIRUSES, OF LACK OF NEGLIGENCE OR NON-INFRINGEMENT.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published, and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this section are included on all such copies and derivative works. However, this document itself may not be modified in any way, including by removing the copyright notice or references to OpenCAPI, except as needed for the purpose of developing any document or deliverable produced by an OpenCAPI Work Group (in which case the rules applicable to copyrights, as set forth in the OpenCAPI IPR Policy, must be followed) or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by OpenCAPI or its successors or assigns while the specification in question is the current version. Upon release by the OpenCAPI Consortium of a new version, all the above rights shall automatically cease.

OpenCAPI and the OpenCAPI logo design are trademarks of the OpenCAPI Consortium.

Other company, product, and service names may be trademarks or service marks of others.

Abstract

This document defines the requirements that need to be met to be asserted as an OpenCAPI 3.0 Ready device or OpenCAPI 3.0 Ready host. It is the work product of the OpenCAPI Consortium Compliance Work Group

This document is handled in compliance with the requirements outlined in the OpenCAPI Consortium Work Group (WG) process document. Comments, questions, etc. can be submitted to membership@opencapi.org.

Contents

Revision log	4
About this document	5
Conventions.....	5
Notes	5
Engineering notes	5
Developer notes	5
Terms	6
References	7
1. Introduction	8
2. OpenCAPI Ready testing	9
3. PHY validation tests	10
3.1 System and adapter test configuration	10
3.2 System and adapter data collection	10
3.3 System PHY qualification testing.....	11
3.4 Adapter PHY qualification testing	12
3.5 Clock characterization testing.....	13

Approved

Revision log

Each release of this document supersedes all previously released versions. The revision log lists all significant changes made to the document since its initial release.

Revision date	Summary of changes
20 May 2020	Version 1.0. Initial release.

About this document

This document defines the requirements that must be met to be asserted as an OpenCAPI 3.0 Ready device or OpenCAPI 3.0 Ready host. It is the work product of the OpenCAPI Consortium OpenCAPI Compliance Work Group.

Conventions

The OpenCAPI Consortium documentation uses several typesetting conventions.

Notes

This section describes Engineering and Developer notes.

Engineering notes

Engineering notes provide additional implementation details and recommendations not found elsewhere. The notes might include architectural compliance requirements. That is, the text might include Architecture compliance terminology. These notes should be read by all implementation and verification teams to ensure architectural compliance.

Engineering note:

This is an example of an Engineering note. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Proin cursus hendrerit enim, vel tempus nibh ornare ut. Quisque ac augue eu augue convallis hendrerit. Mauris iaculis viverra ipsum nec dapibus. Nunc at porta libero. Curabitur luctus ultrices augue non pulvinar. Vestibulum mattis non ipsum at venenatis. Suspendisse euismod, neque et suscipit luctus, odio metus semper lectus, quis volutpat est libero quis nunc. Vivamus rutrum mauris sed tristique malesuada.

Developer notes

Developer notes are used to document the reasoning and discussions that led to the current version of the architecture. These notes might also include recommended changes for future versions of the architecture, or warnings of approaches that have failed in the past. These notes should be read by verification teams and contributors to the architecture.

Developer note:

This is an example of a Developer note. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Proin cursus hendrerit enim, vel tempus nibh ornare ut. Quisque ac augue eu augue convallis hendrerit. Mauris iaculis viverra ipsum nec dapibus. Nunc at porta libero. Curabitur luctus ultrices augue non pulvinar. Vestibulum mattis non ipsum at venenatis. Suspendisse euismod, neque et suscipit luctus, odio metus semper lectus, quis volutpat est libero quis nunc. Vivamus rutrum mauris sed tristique malesuada.

Approved

Terms

The following terms are used in this document.

AFP	Advanced function presentation.
AFU	Attached functional unit. Architecturally, AFU refers to an end-point unit or resource. Communication from the processor to the AFU goes through a protocol stack, transaction layer (TL), data link layer (DL), and physical medium layer (PHY). Command and data packets at the AFU interface are specified by the AFU command/data interface, which is the interface between the AFU protocol stack and the AFU.
BER	Bit error rate.
BW	Bandwidth.
CPU	Central processing unit.
CRC	Cyclic redundancy check.
DFE	Decision feedback equalization.
DL	OpenCAPI data link layer found on the host processor.
DLx	OpenCAPI data link layer found on the external OpenCAPI device.
FPGA	Field-programmable gate array.
IBERT	Integrated bit-error-rate test.
MCP	Management control point.
OPAL	Open Power Abstraction Layer.
OpenCAPI Ready™	The term defined in this document that asserts a minimum set of characteristics has been met to show a product should be interoperable with other OpenCAPI products.
PCIe	Peripheral Component Interconnect Express.
PHY	Physical medium layer. The PHY layer interfaces to the DL and the network.
SERDES	Serializer/deserializer.
TL	OpenCAPI transaction layer found on the host processor.
TLx	OpenCAPI transaction layer found on the external OpenCAPI device.
TMLA	Trademark license agreement.

Approved

References

The following documents can be helpful when reading this specification.

[OpenCAPI 3.0 Transaction Layer Specification](#)

25 Gbps Physical Signaling Specification

25 Gbps Interface Mechanical Specification

OpenCAPI 3.0 Certified Definition

OpenCAPI 3.0 Ready Definition

The following information is located on the [OpenCAPI Consortium website](#):

- OpenCAPI Ready trademark
- OpenCAPI Ready mark (Logo)
- OpenCAPI Ready list
- OpenCAPI Ready request form
- TMLA links/references/options

1. Introduction

The OpenCAPI Certified™ program is used by the OpenCAPI Consortium to enable OpenCAPI ecosystem product developers to indicate that a product has been shown/demonstrated to meet a minimum set of characteristics and should be interoperable with other OpenCAPI Certified products. This document serves as a reference for independent test labs performing OpenCAPI Certified testing.

Approved

2. OpenCAPI Ready testing

Functional testing is defined in the OpenCAPI Ready Definition document. While product manufacturers can perform the OpenCAPI Ready Functional Testing as a self-assessment, they may also choose to use an Independent Test Lab. It is recommended that products going through OpenCAPI 3.0 Certified testing have also gone through OpenCAPI 3.0 Ready testing; either through self-assessment or assessment by an Independent Test Lab.

3. PHY validation tests

The following tests should be performed by an Independent Test Lab as part of OpenCAPI Certified testing.

3.1 System and adapter test configuration

- Install on the Host system the latest GA1 level firmware, including any required patches that have not been released yet, including the necessary I2C reset signal patch. Set system fans to maximum.
- Install one or four BW250SoC adapters. Adapters must have presence detect rework and must not have been used in a system without an OPAL I2C reset patch.
- Connect the Host system and adapters using an Amphenol Oculink - Slimsas Cable (200 mm) or similar.

3.2 System and adapter data collection

For each test, collect the following data:

- System name
- Adapter serial numbers
- Image name and release date
 - IBERT, AFP, MCP
- For IBERT
 - Bit error count for each lane
 - Eye opening for each lane
 - Test parameters (BER)
 - Both FPGA and POWER9 results
 - Any observations
- For AFP / MCP
 - CRC count for each loop
 - Number of loops with CRCs
 - Functional test results
 - BW for AFP
 - Pass/fail for MCP
 - Save all created Log files and make notes of any observations

3.3 System PHY qualification testing

If the product being tested is a Host system, perform the following with a golden adapter card. The primary variables between tests are the SERDES configurations settings of Boost, Pre-emphasis and Post-emphasis.

1. Connect Host system and adapter in system. Configure adapter with DFE on, Pre-emphasis Nominal, and Post-emphasis Nominal.
2. Configure Host with Boost = 0, Pre-emphasis Nominal, Post-emphasis Nominal.
 - a. Run iBERT to verify BER of 10^{-12} . Repeat for all lane, connector, and CPU combinations.
3. Configure Host with Boost = 0, Pre-emphasis Nominal + 1, Post-emphasis Nominal + 1.
 - a. Run iBERT to verify BER of 10^{-12} . Repeat for all lane, connector, and CPU combinations.
4. Configure Host with Boost = 0, Pre-emphasis Nominal - 1, Post-emphasis Nominal - 1.
 - a. Run iBERT to verify BER of 10^{-12} . Repeat for all lane, connector, and CPU combinations.
5. Configure Host with Boost = 0, Pre-emphasis Nominal, Post-emphasis Nominal.
 - a. Reboot five times; check the OPAL log for training completion and no CRC errors.
 - b. Repeat for all lane, connector, and CPU combinations.
6. Configure Host with Boost = 0, Pre-emphasis Nominal + 1, Post-emphasis Nominal + 1.
 - a. Reboot five times; check the OPAL log for training completion and no CRC errors.
 - b. Repeat for all lane, connector, and CPU combinations.
7. Configure Host with Boost = 0, Pre-emphasis Nominal - 1, Post-emphasis Nominal - 1.
 - a. Reboot five times; check the OPAL log for training completion and no CRC errors.
 - b. Repeat for all lane, connector, and CPU combinations.
8. Configure Host with Boost = 0, Pre-emphasis Nominal, Post-emphasis Nominal.
 - a. With the system in an idle state, check the CRC error detect registers 100 times. OCutil can be used to do this.
 - b. Repeat for all lane, connector, and CPU combinations.

3.4 Adapter PHY qualification testing

If the product being tested is an adapter card, perform the following tests with the card installed in a golden host system. The primary variables between tests are the SERDES configurations settings of DFE, Pre-emphasis, and Post-emphasis.

1. Connect Host system and adapter in the system. Configure the Host with nominal SERDES settings.
2. Configure the adapter with DFE on, Pre-emphasis Nominal, Post-emphasis Nominal.
 - a. Run iBERT to verify BER of 10^{-12} . Repeat for all lane, connector, and CPU combinations.
3. Configure the adapter with DFE on, Pre-emphasis Nominal + 1, Post-emphasis Nominal + 1.
 - a. Run iBERT to verify BER of 10^{-12} . Repeat for all lane, connector, and CPU combinations.
4. Configure the adapter with DFE on, Pre-emphasis Nominal - 1, Post-emphasis Nominal - 1.
 - a. Run iBERT to verify BER of 10^{-12} . Repeat for all lane, connector, and CPU combinations.
5. Configure the adapter with DFE on, Pre-emphasis Nominal, Post-emphasis Nominal.
 - a. Reboot five times; check the OPAL log for training completion and no CRC errors.
 - b. Repeat for all lane, connector, and CPU combinations.
6. Configure the adapter with DFE on, Pre-emphasis Nominal + 1, Post-emphasis Nominal + 1.
 - a. Reboot five times; check the OPAL log for training completion and no CRC errors.
 - b. Repeat for all lane, connector, and CPU combinations.
7. Configure the adapter with DFE on, Pre-emphasis Nominal - 1, Post-emphasis Nominal - 1.
 - a. Reboot five times; check the OPAL log for training completion and no CRC errors.
 - b. Repeat for all lane, connector, and CPU combinations.
8. Configure the adapter with DFE on, Pre-emphasis Nominal, Post-emphasis Nominal.
 - a. With the system in an idle state, check the CRC error detect registers 100 times. OCutil can be used to do this.
 - b. Repeat for all lane, connector, and CPU combinations.

3.5 Clock characterization testing

The following procedures should be performed on the system RefCLK as part of OpenCAPI Certified testing.

1. Configure SSclk OFF.
2. Refclk measurements to be done at the output of the Clock generator chip. Use a solder-on probe tip at the board via near the output pin of the Clock chip. If a transition via is not offered in the design, then the probe pad can be provided on the top side of the board.
3. Perform a measurement using a sufficient bandwidth oscilloscope and a third-party compliance software package such as SiLabs or IDT. For configuration of the oscilloscope (BW, sample rate, memory size), refer to the corresponding PCIe compliance requirements. Use the oscilloscope to capture the clock waveform and then post process using the third-party compliance software.
4. Repeat with SSclk ON.