



# **Bunch of Wires PHY Specification**

The Open Domain-Specific Architecture BoW  
Workstream

Mark Kuemerle, Ramin Farjad, Ken Poulton, Suresh  
Subramaniam, Bapi Vinnakota

This document to be released under the following OCP copyright release:

#### COPYRIGHT LICENSE AGREEMENT

This Agreement ("Agreement") is entered into on the date set forth below, (the "Effective Date") by and between the Open Compute Project Foundation a Delaware corporation ("OCP") and the entity identified below ("Licensor").

WHEREAS, Licensor is the owner of and/or has certain rights in or to the works of authorship identified in the attached Exhibits (collectively, the "Work").

WHEREAS, OCP desires to copy, distribute, make derivative works of and publish the Work and derivative works thereof, including without limitation in one or more OCP publications and/or on OCP's website, and Licensor will benefit from OCP's use of the Work as described in this Agreement.

NOW THEREFORE, in consideration of the promises in this Agreement, the parties agree as follows:

1. Structure of the Agreement. There may be multiple Exhibits to this Agreement. Each Exhibit will be signed by an authorized representative and will be governed by and subject to the terms set forth in this Agreement, with the licenses applicable to the Work described therein effective as of the date the Exhibit is signed.
2. License. Licensor hereby grants to OCP a non-exclusive, transferable (in accordance with Section 7 below), royalty free, fully-paid, perpetual, irrevocable, worldwide license, under Licensor's copyrights in the Work, with the right to sublicense, to use, reproduce, create derivative works, distribute, and publicly display and perform the Work and derivative works thereof, in whole or in part, as a separate work or as part of a collective work. The foregoing will apply to all mediums now known or hereafter existing.
3. Ownership of the Work / Other Rights Reserved. Except for the foregoing license, as between OCP and Licensor, Licensor retains all right, title and interest in and to the Work and all intellectual property rights therein. Licensor hereby reserves all rights not expressly granted in this Agreement. No additional licenses or rights whatsoever (including without limitation any patent licenses) are granted by implication, exhaustion, estoppel or otherwise.
4. Representations and Indemnification. Licensor represents to OCP that: (i) Licensor is the sole and exclusive owner of the Work and all copyrights therein or Licensor has the right and authority to grant the licenses set forth in this Agreement and (ii) OCP's exercise of the licenses set forth in this Agreement will not result in any infringement of any third party's copyrights or the misappropriation of any third party's trade secrets. Licensor agrees to indemnify and hold OCP harmless from and against any losses, damages, liabilities, settlement amount, costs and expenses (including reasonable attorneys' fees) incurred by OCP in connection with any breach of the foregoing representations. This Section will survive the termination of this Agreement.
5. Term and Termination. This Agreement will commence on the Effective Date and will terminate upon the written agreement of the parties or by written notice by OCP.
6. Governing Law and Forum. This Agreement shall be solely and exclusively governed, construed and enforced in accordance with the laws of the Texas, USA, without reference to conflict of laws principles. Any suit, action or proceeding arising from or relating to this Agreement must be brought, solely and exclusively, in courts located in Travis County, Texas and each party irrevocably consents to the jurisdiction and venue of any such court.
7. Assignment. OCP may assign this Agreement (a) with the consent of Licensor, not to be unreasonably withheld or delayed, or (b) upon notice, but without such consent, in

connection with a merger, acquisition, change of control, or sale of substantially all the assets of OCP. This Agreement shall be binding upon and inure to the benefit of the parties and their successors and permitted assigns.

8. Mutual Limits on Liability. Except as set forth below, in no event shall either party be liable to the other party in any manner, under any theory of liability, whether in contract, tort (including negligence), or other theory, for any indirect, consequential, incidental, exemplary, punitive, statutory or special damages, including lost profits, regardless of whether such party was advised of or was aware of the possibility of such damages. Except as set forth below, in no event shall the total, cumulative liability of either party regarding any and all claims and causes of action, under any theory of liability, whether in contract, tort (including negligence), or otherwise, exceed One Thousand Dollars (\$1,000). The limitations set forth in this Section will not apply to liability arising under Section 4 (Representations and Indemnification) above. This Section will survive termination of this Agreement.
9. Entire Agreement. This Agreement constitutes the entire agreement between the parties with respect to its subject matter and it supersedes all prior or contemporaneous oral or written agreements and representations concerning the subject matter herein. This Agreement may be amended only in a written document signed by both parties. This Agreement shall not be interpreted or construed against the party preparing it.
10. Counterparts and Facsimile Signatures. This Agreement may be executed in counterparts all of which taken together shall constitute one single agreement between the parties. A facsimile transmission of the executed signature page of this Agreement shall constitute due and proper execution of this Agreement by the applicable party.

## Contents

- 1. Introduction**
  - 1.1. Objectives
  - 1.2. Advantages
  - 1.3. Scope
  - 1.4. Language
  - 1.5. Compliance Summary
- 2. BoW Overview**
  - 2.1. BoW Slice
  - 2.2. BoW Modes
  - 2.3. PHY - Link Layer Interface
  - 2.4. BoW Wires
  - 2.5. BoW for Common Transaction Protocols
- 3. Signal Definitions**
  - 3.1. Die-to-die Signals (Wires)
    - 3.1.1. DBI on the AUX wire
  - 3.2. Link Layer Interface - Clock and Data Signals
    - 3.2.1. Link Layer Interface - Clock and Data Signal List
  - 3.3. Link Layer Interface - PHY Control Signals
    - 3.3.1. PHYReset
    - 3.3.2. MDIO Serial Programming Interface
    - 3.3.3. PHYReady
    - 3.3.4. MDIO Register Map
- 4. BoW Modes**
  - 4.1. BoW Modes
- 5. BoW Physical Configuration**
  - 5.1. BoW Components
  - 5.2. Example Link
  - 5.3. Die-to-Die Signals
  - 5.4. Bump Arrangements
  - 5.5. Cross Section
  - 5.6. Signal Ordering
  - 5.7. Checkerboard Slice Pattern
- 6. Timing**
  - 6.1. Clocking
  - 6.2. Clock and Data Specifications
    - 6.2.1. Timing Requirements
- 7. Electrical Specifications**
  - 7.1. Voltages
  - 7.2. ESD
  - 7.3. Termination and Return Loss
- 8. Chip-to-Chip Channel Specifications**
  - 8.1. BoW Channel Specifications on Laminate
  - 8.2. BoW Channel Specifications for Doubly-Terminated Links
    - 8.2.1. Channel Loss
    - 8.2.2. Crosstalk
- 9. Reset and Initialization**
  - 9.1. EVERYTHING AFTER THIS POINT STILL NEEDS REVISION IN THE MAR 2021 REWRITE**
  - 9.2. Control Options
  - 9.3. Data-Transfer Ready
  - 9.4. Standby Mode
  - 9.5. Data-Transfer Ready Signals
  - 9.6. The Effects of De-asserting Data-Transfer Ready
  - 9.7. Initialization
  - 9.8. Power-on Reset Synchronization

- 9.9. Power-on Reset Signals
- 9.10. Power-on Reset Sequence
- 9.11. Unused Interfaces
- 9.12. Test Provision
- 10. Configuration**
  - 10.1. Output State During Configuration
  - 10.2. Chiplet Configuration
  - 10.3. BoW Interface Configuration
  - 10.4. JTAG Configuration
  - 10.5. Configuration Completion Signals
- 11. Calibration**
  - 11.1. Data-Path Calibration
  - 11.2. Adapter Reset
  - 11.3. Calibration Request
  - 11.4. DCC Calibration
  - 11.5. DLL Calibration
  - 11.6. Calibration Completion
  - 11.7. BoW Link Ready
  - 11.8. Link Training
- 12. Control-Signal Shift Register Mapping**
- 13. Testability**
  - 13.1. Test Pattern
  - 13.2. Loopback Test

# Bunch of Wires PHY Specification

## The Open Domain-Specific Architecture BoW Workstream

Mark Kuemerle, Ramin Farjad, Ken Poulton, Suresh Subramaniam, Bapi Vinnakota

### 1. Introduction

The Bunch of Wires (BoW) is a simple, open and interoperable physical interface between two chiplets or chip-scale-packages (CSP) in a common package. This document specifies the BoW interface PHY layer.

#### 1.1. Objectives

The BoW interface is a set of die-to-die parallel interfaces that provides the flexibility to trade off throughput/chipedge for design complexity, cost, and packaging technology.

The use of BoW is expected to be confined to connect die placed close to one another within the same package. In this environment, signal attenuation is small and the interface can be simple.

The definition of the BoW interface aims to meet the following design objectives:

- Inexpensive to implement
- Portable across IC process nodes ranging from 65 nm to 5 nm
- Flexible to support both existing and advancing packaging technologies
- Portable across multiple bump pitches
- Unencumbered by technology license costs
- Very low power (< 1 pJ/bit) as defined by Tx IO Pad, wire and Rx IO Pad.
- Very low latency (<5 ns without FEC, <15 ns with FEC from link layer to link layer)

- High throughput density (100-1000 Gb/s/mm-chip-edge)
- Backwards compatible (across at least two major specification versions)

## 1.2. Advantages

The Bunch of Wires interface provides several key advantages for chiplet-based systems:

- Can operate at higher data rates per pin than existing parallel standards
  - or at lower data rates for compatibility with existing parallel standards
- Can be implemented in legacy technologies (process nodes) with generally available IP
- Can be implemented in low-cost laminates or higher-density silicon-based interconnect
- Can be implemented with much less design effort than a traditional SERDES
- Is not constrained to a specific bump pitch
  - interfaces with somewhat different bump pitches can be connected

Compared to serdes, BoW uses a lower data rate/wire and so it requires more wires. But the lower data rates allow use of single-ended signaling and denser wire packing. In addition, in laminates, BoW can take advantage of multiple wiring layers and in advanced packaging it can take advantage of the much-increased wire density.

## 1.3. Scope

The scope of this document has several levels.

1. The specification of the BoW interface includes these requirements:

- a. Operating modes
- b. Chip-to-chip wire signals
- c. Wire ordering
- d. Timing and electrical specifications on the chip-to-chip interface
- e. Signals at the link layer interface
- f. Configuration, initialization, calibration
- g. Functions that must be supported at the link layer or above

2. The specification includes recommendations for these elements:

- a. Bump patterns
- b. Arrangement of multiple slices in a link
- c. Arrangement of wires in laminate and advanced packaging
- d. signal integrity of the wire channel
- e. Configuration and management programming
- f. Design for test and test methods
- g. Performance estimates
- h. Compliance verification

3. The following activities are outside the scope of this document:

- a. Specific implementations of the interface
- b. Integration of the interface with system-level data flow e.g. interface to a PHY-layer abstraction such as PIPE/PCIe interface to the BoW
- c. The use of this interface outside of a package or entirely inside a chip
- d. Definition of protocols for logical data transfer

4. The following aspects are intended to be addressed in subsequent versions of this specification:

- a. Simultaneous bidirectional data (full duplex on each wire)
- b. Security

## 1.4. Language

- “Shall” or “must” indicates a requirement. Failure to meet the requirement results in non-compliance
- “Should” indicates a strong suggestion, but not a requirement. Failure to implement the suggestion does not result in non-compliance.
- “May” indicates an implementation option.
- The lack of one of the above verbs indicates the material is informative.
- “Reference” indicates a reference design that is provided as example for explanation, but is not a requirement.

## 1.5. Compliance Summary

The specifications must be met over process variation, supply voltage range and temperature range (PVT). Each implementation must document its supported supply voltage range and temperature range.

Table XX will summarize the compliance points that shall be met in order to comply with the BoW specification. Each of the compliance points is discussed in the specification.

Table 1 below summarizes these signals.

Todo: Need to fill in this table:

| Section    | Description                  |
|------------|------------------------------|
| 2.3        | BoW Modes                    |
| 3.1        | Chip to chip signals (wires) |
| <b>5.1</b> | Wire order                   |
|            | Much more....                |

**Table 1.** BoW Compliance Summary

Do we also need a discussion of interoperability?

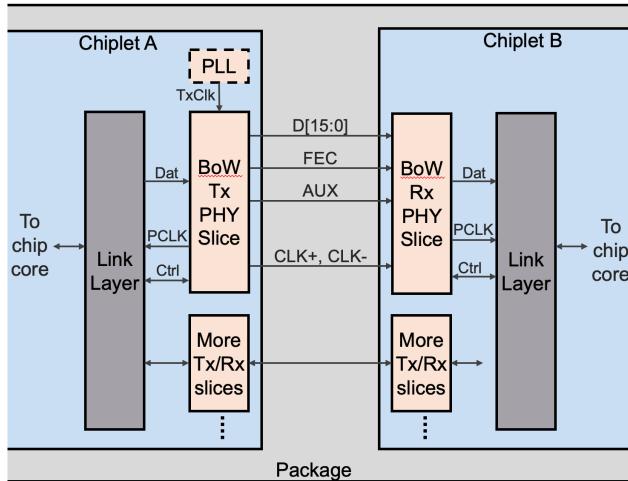
## 2. BoW Overview

Chiplet-based designs require physical and logical connectivity between the die in a single package. This section provides an overview of the BoW physical interface (PHY), its use in a multi-chiplet design, and how with the Open Domain-Specific Architecture stack it can be used as an underlay for popular transaction protocols.

### 2.1. BoW Slice

BoW is an energy-efficient, easy-to-use PHY interface between a pair of die inside a single package as shown in Figure 1. The BoW PHYs between two die are physically connected through wires on a substrate or interposer. A BoW PHY does not have enough drive strength for off-package interfaces, nor is it designed for buses that are entirely on die.

The BoW PHY is defined as a single unidirectional slice. Multiple slices are combined to create links of the desired throughput. A link may be symmetric, asymmetric or unidirectional.

**Figure 1.** BoW Overview

A BoW PHY slice either transmits or receives 16 bits of data between die. Since the BoW is a source-synchronous PHY, each transmitting PHY slice transmits a complementary clock signal CLK+ and CLK- with the data. A BoW PHY optionally has two additional wires designated FEC (for Forward Error Correction) and AUX, for other optional functions such as Data Bus Inversion (DBI).

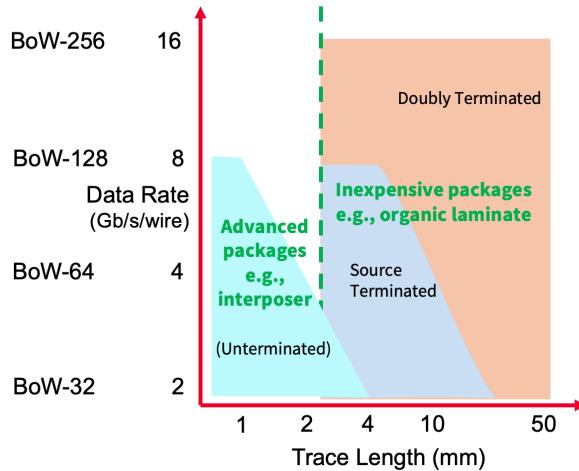
## 2.2. BoW Modes

A BoW interface must conform to one of the Bow Modes seen in Table 2.

| BoW Mode | Slice Bit Rate Gb/s | Wire Bit Rate Gb/s/wire | TxClk GHz |
|----------|---------------------|-------------------------|-----------|
| BoW-32   | 32                  | 2                       | 1         |
| BoW-64   | 64                  | 4                       | 2         |
| BoW-128  | 128                 | 8                       | 4         |
| BoW-256  | 256                 | 16                      | 8         |

**Table 2.** BoW Levels

The BoW Mode defines the speed of clock and data of the PHY on the die-to-die wires. In all modes, the data must be clocked DDR: the data wire bit rate is double the clock wire frequency. All BoW interfaces must be able to interoperate with all the lower modes. Supporting rates other than the defined four modes is an implementation choice. There is more detail on BoW Modes in section 4.1.



**Figure 2.** BoW Data Rate vs. Reach tradeoff

Figure 2 shows the tradeoff between package, data rate, termination, and reach. Source-terminated BoW on laminate allows a longer reach than advanced packaging, but the wider design rules in laminate means that both of these cases are barely able to reach 8 Gb/s/wire. A doubly-terminated link offers longer distances and higher rates, but requires a more complicated receiver design.

### 2.3. PHY - Link Layer Interface

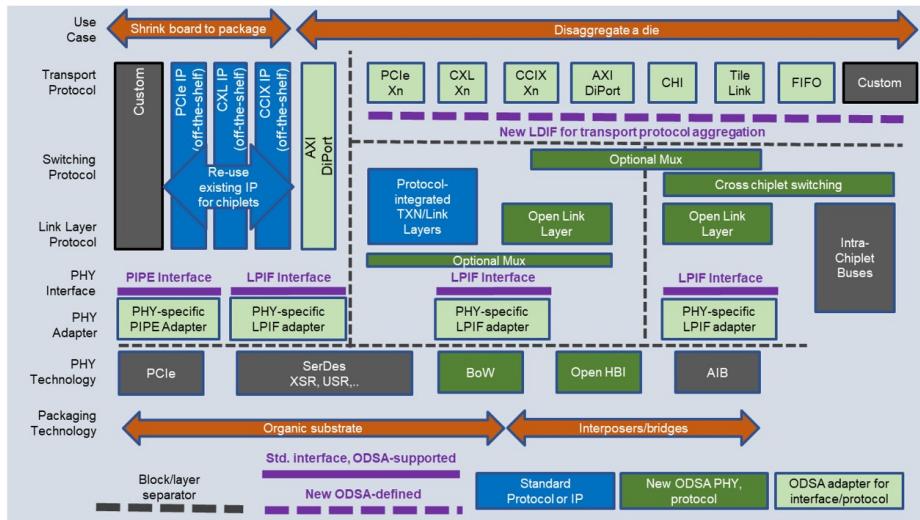
The speed at the link layer interface (Figure 1) is implementation-dependent. Typically, PCLK will be the TxClk frequency divided by a power of 2, so 250, 500 and 1000 MHz are common rates. The data at the link layer interface is SDR (bit rate equal to PCLK frequency).

### 2.4. BoW Wires

Within the package, the BoW datapath is transported on physical passive wires between the pair of connected die. The specifics of the wires, such as their density, maximum length, impedance characteristics and how they are realized vary with the packaging technology. In order to minimize power, unterminated and source-terminated links will have short reaches requiring chips to be adjacent.

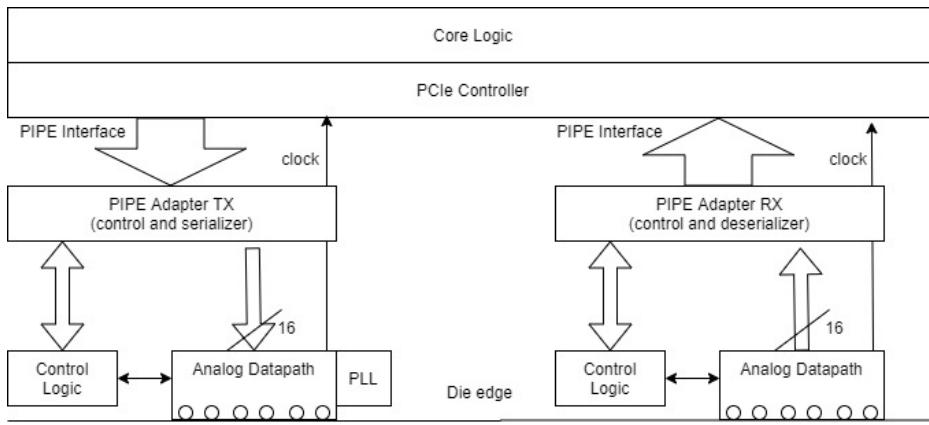
### 2.5. BoW for Common Transaction Protocols

Two connected die in a multi-chiplet device need to exchange logical information. The ODSA aims to define an open physical and logical interface for chiplets, as shown in Figure 3 to enable chiplets from multiple vendors to interoperate and be integrated in a multi-die package. The Bunch of Wires is an open D2D PHY option in the interface. The logical component of the ODSA interface aims to support protocols used for the two most common chiplet use cases, package aggregation and die disaggregation across a wide range of open and proprietary D2D PHYs such as PCIe, CXL, CCIX, AXI and proprietary streaming protocols.

**Figure 3.** The BoW PHY in the ODSA Stack

The ODSA stack abstracts the PHY layer from the logical interface by using the well-defined abstraction interfaces PIPE and LPIF. Any logic transaction controller, such as a PCIe controller, that supports a PIPE or LPIF interface can use any D2D PHY that also supports that interface as its physical layer. As shown in Figure 3, the BoW interface may receive data through either the PIPE or LPIF interfaces to support common transaction protocols. For this use case, some BoW-specific adapter logic will be needed to support the requirements of PIPE or LPIF. The specifications for these adapters are outside the scope of this document. Figure 4 shows how the BoW with an PIPE adapter can be interfaced to a PCIe controller.

Bapi: please remove “serializer” and “deserializer” from the labels in Figure 4 - these are part of the PHY.

**Figure 4.** BoW with a PIPE adapter for PCIe transactions

### 3. Signal Definitions

#### 3.1. Die-to-die Signals (Wires)

As shown in Figure 1, each BoW slice consists of a differential clock pair, 16 single-ended data wires, and optional an optional pair of wires FEC and AUX.

Each BoW slice is unidirectional when in operation. A chiplet may be designed with Rx-only and Tx-only slices, or each slice may have both Tx and Rx capability which is configured at runtime. A bidirectional link is composed of some number of slices configured for Rx and some for Tx.

FEC (Forward Error Correction) is an optional signal that allows using FEC to improve the bit error rate (BER). By using an additional wire when FEC is enabled, the payload data rate is not affected and the wire data rate need not change. This allows  $F(PCLK) = F(TxClk) / 2^n$  with FEC off or on, which simplifies the clock generation and serialization functions. If used, FEC is implemented in the Link layer, and the PHY treats the FEC bit the same as the other data bits.

AUX is an optional signal that can be used for purposes such as Data Bus Inversion (DBI), flow control, redundancy for defect repair, etc. The Link layers of Chiplets A and B will need to agree on the details on FEC and AUX usage. An implementation may choose to support the FEC and AUX wires, or to omit both of them.

Table 3 summarizes these signals.

| Function                 | # Wires | Signal Name | Notes        |
|--------------------------|---------|-------------|--------------|
| Clock                    | 2       | CLK+, CLK-  | Differential |
| Data                     | 16      | D0-15       |              |
| Forward Error Correction | 0/1     | FEC         | Optional     |
| Auxiliary                | 0/1     | AUX         | Optional     |

**Table 3.** BoW Signals at the Die To Die Interface

### 3.1.1. DBI on the AUX wire

Data Bus Inversion (DBI) can be used to mitigate simultaneous switching output (SSO) noise of a BoW PHY by reducing the number of BoW data wires that switch between adjacent data transfer cycles. DBI functionality is optional; it one of several possible uses of the AUX wire. DBI can be implemented in the PHY or in the Link layer.

Within a slice's 16 data signals, the TX DBI logic calculates the DBI bit based on the number of data signals changing from their previous state on the BoW slice wires.

$$\text{DBIcurrent} = ((\text{data}[15]\text{current XOR data}[15]\text{prev}) + (\text{data}[14]\text{current XOR data}[14]\text{prev}) \dots + (\text{data}[1]\text{current XOR data}[1]\text{prev}) + (\text{data}[0]\text{current XOR data}[0]\text{prev})) > 8 ? 1 : 0;$$

If the DBI bit=1 then the Tx DBI logic inverts the Data bits. If DBI = 1 then the Rx DBI logic inverts the Data bits to recover the original data.

## 3.2. Link Layer Interface - Clock and Data Signals

The data at the link layer interface must be SDR (bit rate equal to PCLK frequency). Each Tx or Rx slice shall interface to the link layer with a datapath width of 32, 64, 128, 256 or 512 bits. The corresponding PCLK (Tx, or Rx) shall have a frequency (datawidth\*PCLK) that matches the overall throughput on the output wires of the slice ( $16 * \text{data\_rate}$ ). This allows supporting a range of core logic speeds.

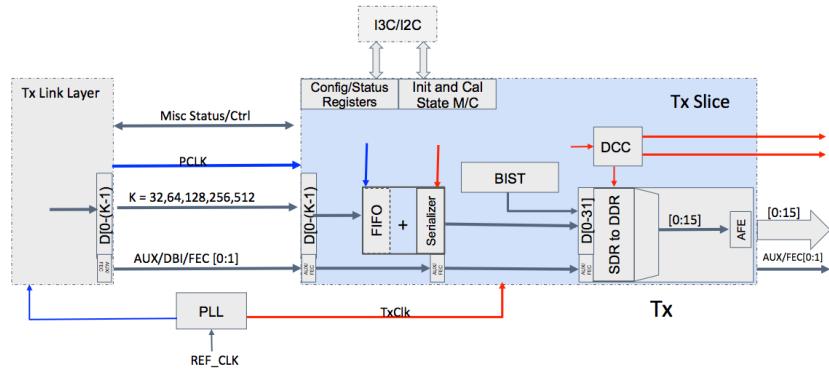
The BoW PHY must contain the serialization or deserialization needed to go between the data rates at the link level interface and the wire interface.

A clock domain crossing (CDC FIFO) is required between the chip core and the PHY's internal clocks; this can be either in the PHY or the link layer. The CDC receives clocks from both the

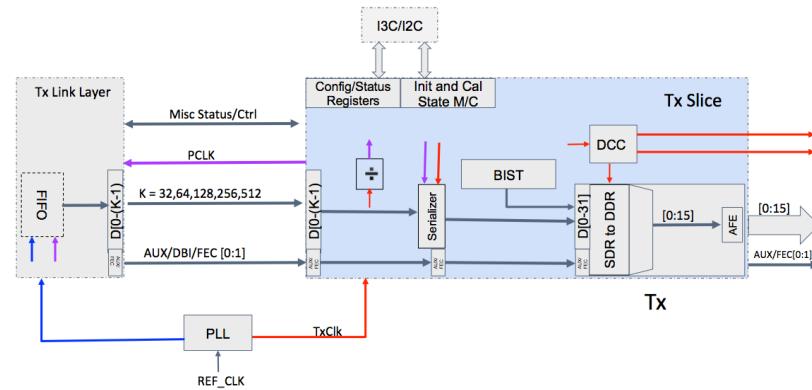
core and the PHY internal clocks, so PCLK flows towards the CDC as seen in the following four figures.

Example datapath implementations of the FIFO inside the PHY are shown in Figs 5, 7. FIFO in the link level is shown in Figs 6, 8. Note that these four figures are intended to illustrate the signals at the perimeter of the PHY slices. The details of the blocks internal to a slice is implementation-dependent.

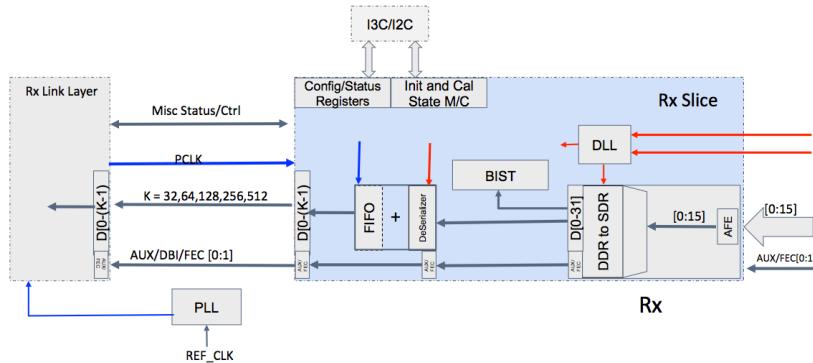
The following four figures need to be revised to show the MDIO interface to the Link layer instead of going direct to I2C.



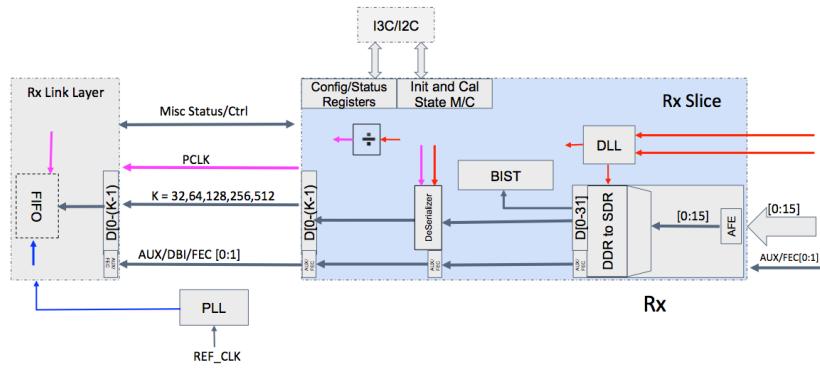
**Figure 5.** BoW Link Layer and Tx PHY Slice with FIFO in the PHY



**Figure 6.** BoW Link Layer and Tx Slice with FIFO in the link layer



**Figure 7.** BoW Link Layer and Rx PHY Slice with FIFO in the PHY



**Figure 8.** BoW Link Layer and Rx Slice with FIFO in the link level

### 3.2.1. Link Layer Interface - Clock and Data Signal List

The signals in Table 4 shall constitute the data interface between the link layer and the PHY.

| Signal | # Bits | Tx Slice | Rx Slice | Description  |
|--------|--------|----------|----------|--|
| Data   | 16*K   | In       | Out      | Data   |
| FEC    | K      | In       | Out      | Forward Error Correction   |
| AUX    | K      | In       | Out      | Auxiliary uses   |
| PCLK   | 1      | *        | *        | * Out if CDC is in Link Level, In if CDC is in the PHY   |
| TxClk  | 1      | In       | NA       | Comes from a PLL or other clock source, not the Link layer. The TxClk source is usually shared among many Tx slices. |

**Table 4.** Link Layer Interface Signals

### 3.3. Link Layer Interface - PHY Control Signals

The signals in Table 5 shall constitute the control interface from the link layer to the PHY.

| Signal   | # Bits | Tx Slice | Rx Slice | Description  |
|----------|--------|----------|----------|--|
| PHYReset | 1      | In       | In       | Resets the BoW slice   |
| PHYReady | 1      | NA       | Out      | 1 indicates that the clocks in the Rx PHY are aligned (DCCs and DLLs locked) |

| Signal | # Bits | Tx Slice | Rx Slice | Description                             |
|--------|--------|----------|----------|---|
| MDC    | 1      | In       | In       | Clock for MDIO serial control interface |
| MDIO   | 1      | BiDi     | BiDi     | Data for MDIO serial control interface  |

**Table 5.** Link Layer Interface Control Signals

Additional implementation-dependent signals may exist.

### 3.3.1. PHYReset

The PHYReset pin is asserted by the Link layer to initialize the PHY. The PHYReset signal shall reset the internal registers to their HW default states, which shall allow MDIO programming and internal self-alignment to take place.

The reset states are otherwise implementation-dependent and shall be documented in the datasheet of a particular implementation.

### 3.3.2. MDIO Serial Programming Interface

The MDC and MDIO pins control the programming and status readout of the PHY by the Link layer. Per the spec at [https://en.wikipedia.org/wiki/Management\\_Data\\_Input/Output](https://en.wikipedia.org/wiki/Management_Data_Input/Output) up to 32 slices may be connected to one bus and each slice may have up to 32 16-bit registers.

The MDIO bus interface shall operate at up to 25 MHz.

[ Need to add a real MDI spec reference. ]

### 3.3.3. PHYReady

After the Rx slice clock self-alignments are complete, each Rx PHY slice shall assert its PHYReady pin. These may be AND'd together by the Link layer or treated separately. How an Rx PHY slice determines completion of the self-alignment is implementation-dependent. For instance, it can be determined by observing the settling of the DLL or by a simple timer.

### 3.3.4. MDIO Register Map

The registers accessed by the MDIO Serial Programming Interface are implementation dependent. The registers shall be fully documented in the datasheet of a particular implementation.

These registers may control:

- Configuration such as Tx vs. Rx, BoW Mode (TxClk rate), PCLK rate
- Test features such as loopback modes
- External control of DCC, DLL
- Readout of status such as DCC and DLL self-alignment settings

## 4. BoW Modes

### 4.1. BoW Modes

A BoW interface must conform to one of the Bow Modes seen in Table 6.

| BoW Mode | Slice Bit Rate Gb/s | Wire Bit Rate Gb/s/wire | TxClk GHz |
|----------|---------------------|-------------------------|-----------|
| BoW-32   | 32                  | 2                       | 1         |
| BoW-64   | 64                  | 4                       | 2         |
| BoW-128  | 128                 | 8                       | 4         |
| BoW-256  | 256                 | 16                      | 8         |

**Table 6.** BoW Modes

The BoW Mode defines the speed of clock and data of the PHY on the die-to-die wires. In all modes, the data must be clocked DDR: the data wire bit rate is double the clock wire frequency. All BoW interfaces must be able to interoperate with all the lower modes. Supporting rates other than the defined four modes is an implementation choice.

The recommended maximum wire reach for different packaging types and terminations is seen in Table 7. Exceeding these reach values degrades the voltage margins at the receiver. "Laminate" is intended to include organic laminate packages (a.k.a. "buildup) and similar technologies with approximately 25 um line and space rules. "Advanced" is intended to include silicon interposer and similar technologies.

These have much finer line and space dimensions, but traces are usually much more resistive than in organic laminate packages and must operate with reduced trace lengths. Termination is not expected to be necessary for implementations targeting Advanced packaging.

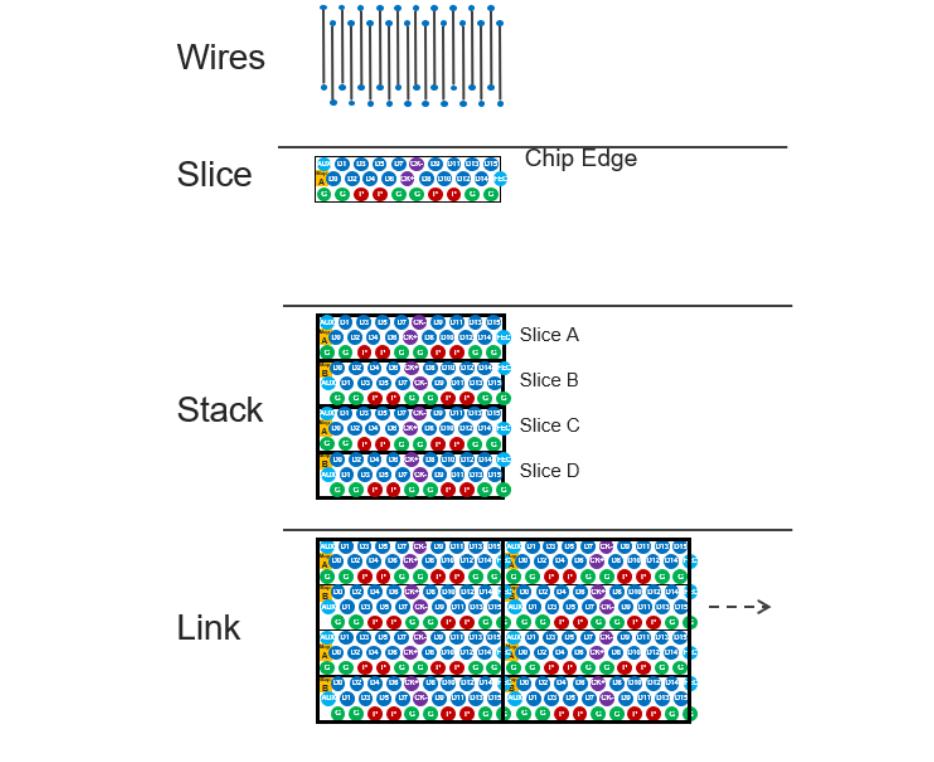
|          |               |       | <b>Laminate</b> | <b>Laminate</b>   | <b>Laminate</b>   | <b>Advanced</b> |
|----------|---------------|-------|-----------------|-------------------|-------------------|-----------------|
|          |               |       | Unterminated    | Source Terminated | Doubly Terminated | Unterminate     |
| Bow Mode | Wire Bit Rate | TxClk | Reach           | Reach             | Reach             | Reach           |
|          | (Gb/s/wire)   | (GHz) | (mm)            | (mm)              | (mm)              | (mm)            |
| BoW-32   | 2             | 1     | 10              | 20                | 50                | 4               |
| BoW-64   | 4             | 2     | NA              | 10                | 50                | 2               |
| BoW-128  | 8             | 4     | NA              | 5                 | 50                | 1               |
| BoW-256  | 16            | 8     | NA              | NA                | 50                | NA              |

**Table 7.** BoW Wire Reaches

Adding termination increases the speed and/or reach, at the expense of greater design complexity.

## 5. BoW Physical Configuration

### 5.1. BoW Components



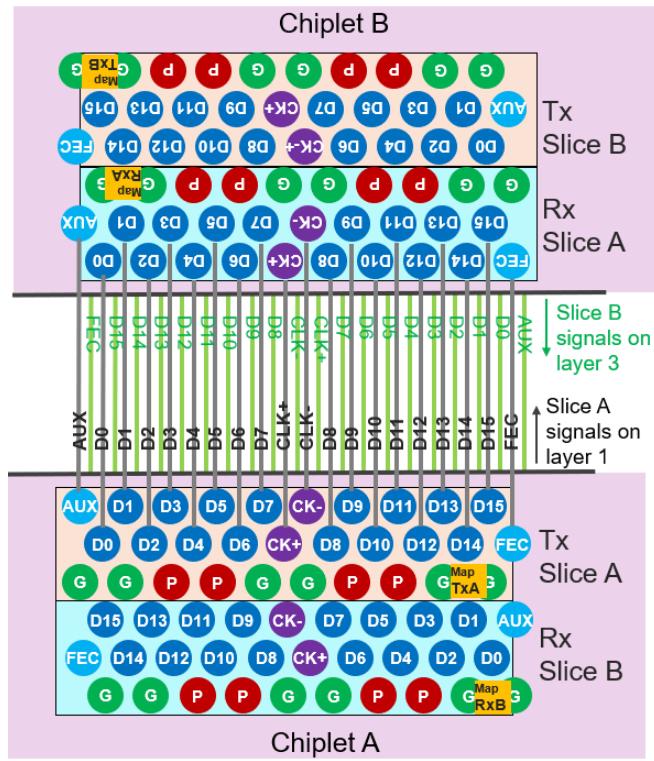
**Figure 9.** BoW Link Components

A BoW link between two chiplets is made up of wires, slices, and stacks as seen in Figure 9.

- The signal traces in the package between chiplets are called **wires**.
- The basic unit is a **slice** with 18 or 20 signal bumps. It must have 2 bumps for the differential clock and 16 single-ended data bumps. It may also have the optional single-ended signals AUX and FEC. The long edge of a slice is parallel to the chip edge.
- Multiple slices may be placed in a **stack**. The slice positions are designated A, B, C, etc, starting with the slice closest to the edge of the chip.
- A **link** from one chiplet to another must be composed of one or more stacks placed along the chip edge. A link may be configured with equal numbers of Rx and Tx slices, or it may be asymmetric or one-way.

## 5.2. Example Link

The minimal bidirectional reference link is shown in Figure 10.



**Figure 10.** BoW Minimal Bidirectional Reference Link

In this example, each chiplet has one Tx slice and one Rx slice, arranged in a single stack on each chiplet. In a laminate package, the position-A slices (at the chips' edges) are connected together on the topmost routing layer used for signals and the position-B slices are connected together on the next layer used for signals.

### 5.3. Die-to-Die Signals

| Function                 | # Signals | Signal Name | Notes        |
|--------------------------|-----------|-------------|--------------|
| Clock                    | 2         | CLK+, CLK-  | Differential |
| Data                     | 16        | D[15:0]     |              |
| Forward Error Correction | 0/1       | FEC         | Optional     |
| Auxiliary                | 0/1       | AUX         | Optional     |

**Table 8.** BoW Die-to-Die Signals

Each BoW slice consists of a differential clock pair, 16 single-ended data wires, and optional wires FEC and AUX. Each BoW slice is unidirectional when in operation. A PHY may be designed as Rx-only and Tx-only slices, or each slice may have both Tx and Rx capability, one of which is selected at configuration time. A bidirectional link is composed of some number of slices configured for Rx and some for Tx.

FEC (Forward Error Correction) is an optional signal that allows using FEC to improve the bit error rate (BER). AUX is an optional signal that can be used for purposes such as DBI, flow control, redundancy, etc. Chiplets A and B will need to agree on the details on FEC and AUX usage, which is defined in the Link layer.

## 5.4. Bump Arrangements

The reference example in Figure 10 uses hexagonal closest packing for the bumps: two rows for signal bumps and one row for power and ground bumps. In this pattern, the wire pitch is half the bump pitch. In order to maintain the closest bump packing, slices in rows B and D have a different bump pattern than slices A and C. But bump patterns are not specified by BoW; only the signal *ordering* at the chip edge is specified for interoperability.

Alternate bump arrangements may include:

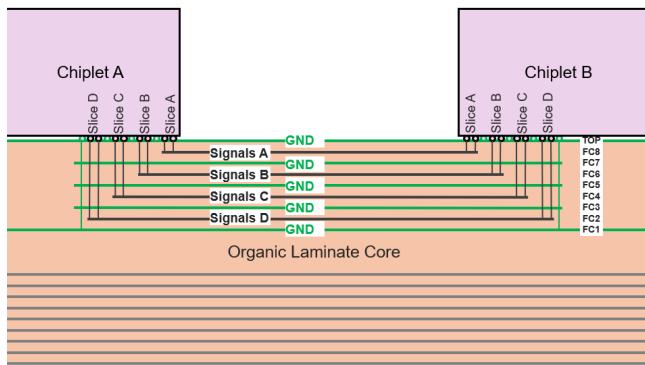
- 90-degree rotation of the hexagonal packing direction to decrease the wire pitch 14%;
- square bump arrays instead of hexagonal for regularity of layout;
- more than two rows of signal bumps to decrease the wire pitch;
- different ordering of power and ground bumps;
- multiple power and ground rows.

An alternate slice arrangement may be to place the Tx and Rx slices side by side at the chip edges. This would take up more chip edge, but allow all the signals to run on the same package layer.

Somewhat different wire and bump pitches between two chiplets can be accommodated with fan-out in the chip-to-chip wires. This is limited by the max wire length.

## 5.5. Cross Section

A cross section for an organic laminate (a.k.a. “buildup”) package is shown in Figure 11.



**Figure 11.** Cross section of a BoW Link in an Organic Laminate Package

In an organic laminate package, signal layers should be alternated with ground layers in order to maintain a controlled impedance of 50 ohms. Each slice position (A, B, C, D) should be associated with one signal layer and there is no mixing of signals from multiple slices.

In any technology, the position-A slice on chiplet A must be connected to the position-A slice on chiplet B (one must be configured for Tx and one for Rx). The position-B slices are connected together, and so on.

There is no specified limit to the number of slices in a stack. In organic laminate, the practical limit in 2020 is an 8-2-8 laminate which supports 4 slices. Layers on the bottom side of the package typically cannot be used for BoW signals due to low via density passing through the thick central core layer.

In advanced packaging technologies, the shorter wire lengths allow the use of non-controlled-impedance wires and unterminated transmitters and receivers. The smaller wire and space dimensions may allow the wires for multiple slices to be interleaved on a single wiring layer.

## 5.6. Signal Ordering

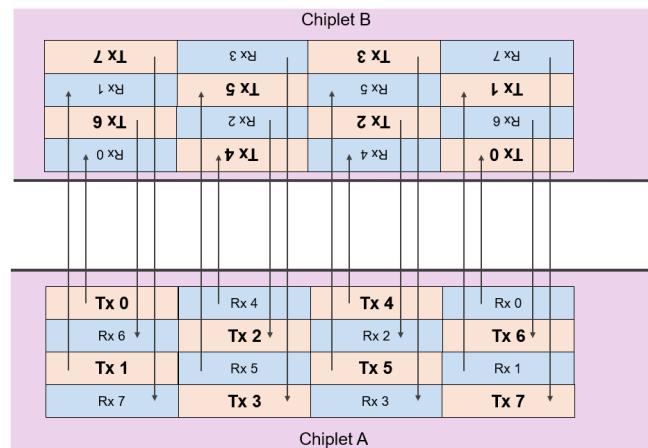
A BoW interface must conform to these wire and slice order rules at the edge of the chip:

- The signals for a Tx slice are in the following order at the chip edge, going clockwise around the chiplet: AUX, D0, D1, D2, D3, D4, D5, D6, D7, CLK+, CLK-, D8, D9, D10, D11, D12, D13, D14, D15, FEC
- The signals for an Rx slice are in the reversed order (ascending goes counter-clockwise)
- The Tx slices in a link are numbered from 0 at the upper left edge of the link (facing from the chip center to the edge) and ascending through the Tx slices in a stack, then from stack to stack clockwise.
- The Rx slices in a link are numbered from 0 at the upper right, through the Rx slices in a stack, then stack to stack counterclockwise.

These rules allow BoW chiplets to be connected without signal reordering regardless of chiplet rotations.

## 5.7. Checkerboard Slice Pattern

For bidirectional links with more than one stack on each side, a checkerboard pattern of Tx and Rx slices should be used (Figure 12). This allows efficient connection of chiplets with differing stack depths and numbers of stacks. Figure 12 shows a bidirectional link with 4 stacks of 4 slices each, for 8 Tx and 8 Rx slices on each chiplet.



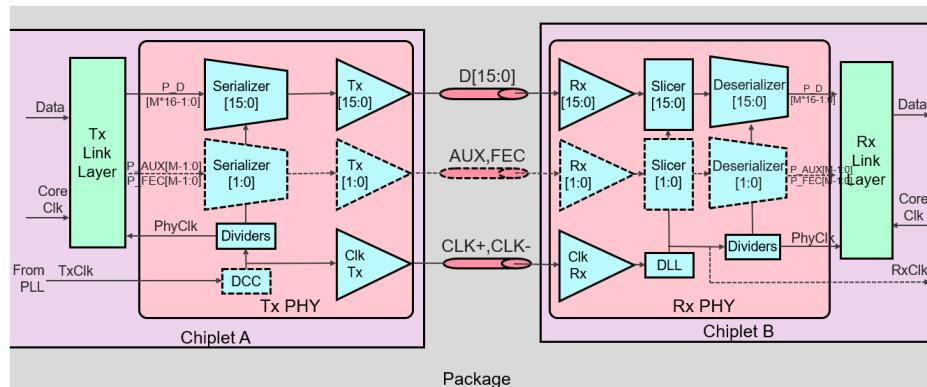
**Figure 12.** Checkerboard Pattern of Tx and Rx Slices in a 1.25 Tb/s Link

An alternate approach may be used: design every slice to operate as either Rx or Tx to be configured after assembly or upon powerup. This allows complete flexibility in interoperability and also provides an opportunity for wafer-test loopback testing.

In BoW-64 at 4 Gb/s/wire, the link in Figure 12 provides a total of 1.0 Tb/s. In an organic substrate using the hexagonal bump pattern of Figure 10 with a bump pitch is 130 um, the total edge width is 4.16 mm without AUX and FEC, or 5.2 mm with; the depth from the edge is 1.35 mm. In an interposer, if the bump pitch is 40 um, the edge width is 1.28 or 1.60 mm and the depth is 0.42 mm.

## 6. Timing

### 6.1. Clocking



**Figure 13.** BoW Clock and Data Block Diagram - One Tx Slice, One Rx Slice

Figure 13 shows the clock and data flow for a single Tx slice and a single Rx slice. On the Tx side, data bits (and optional FEC and AUX bits) come in a wide word from the link layer, and are serialized to the line rate. At the Rx side, they are sampled with a common slicer clock in most BoW implementations. BoW-256 may optionally implement per-bit delay adjust or per-bit slicer clock adjust.

All BoW interfaces shall be DDR (Double Data Rate) at the chip-to-chip interface: the data bit rate is twice the clock frequency, so data is clocked in on both edges of the clock in the Rx slice.

The following table provides clock and data rates for an example with 4 Gbps wire data rate and M=4 to support a 1 Gbps data rate at the Link-PHY interface.

| Signal                          | Rate            | SDR/DDR |
|---------------------------------|-----------------|---------|
| TxClk                           | 2 GHz           |         |
| CLK+,CLK-<br>D[15:0],AUX,FEC    | 2 GHz<br>4 Gbps | DDR     |
| PhyClk                          | 1 GHz           | SDR     |
| P_D[63:0],P_AUX[3:0],P_FEC[3:0] | 1 Gbps          | SDR     |

**Table 9.** Example Clock and Data Rates for Figure 13 with 4 Gbps, M=4

The ratio M should be limited to integers, preferably powers of two, and other ratios implemented in a gearbox in the Link layer.

The DDR clock TxClk is provided to the Tx PHY from elsewhere on Chiplet-A. This may come from an on-chip PLL (typically shared across multiple slices) or it can be routed from the RxClk of an Rx slice on Chiplet-A. In order to meet duty cycle requirements, a Duty Cycle Corrector (DCC) may be needed in the Tx slice. TxClk is used to drive the serializers and provide the output CLK+,CLK- to Chiplet-B.

On the Rx side, the PHY must align the slicer clock to sample the data correctly. This may be done with a DLL or adjustable delays or other methods. The PHY shall include control logic to self-align the slicer clock for correct sampling of the data. Alignment is started by signal AlignDll from the Rx Link Layer; the PHY provides a signal DllAligned to the Link Layer when it is complete.

All BoW interfaces shall be source synchronous within a slice. BoW-32 to BoW-128 interfaces do not require per-wire alignment - the signals within a slice are aligned sufficiently well by matching their paths. BoW-256 interfaces may need per-wire delay adjustment or per-slicer clock adjustment.

Clock skew between the slices in each direction of a link depends on the implementation of the TxClk distribution to all the Tx slices. That is, for the data flow from Chiplet A to Chiplet B, the TxClk distribution on Chiplet A sets the the clock skew of the Tx slices on Chiplet A and the clock skew of the Rx slices on Chiplet B, and vice versa for flow from B to A. This skew must be no more than 100 ps/stack along the chip edge. There is no specification of the skew between TxClk on Chiplet A vs TxClk on Chiplet B.

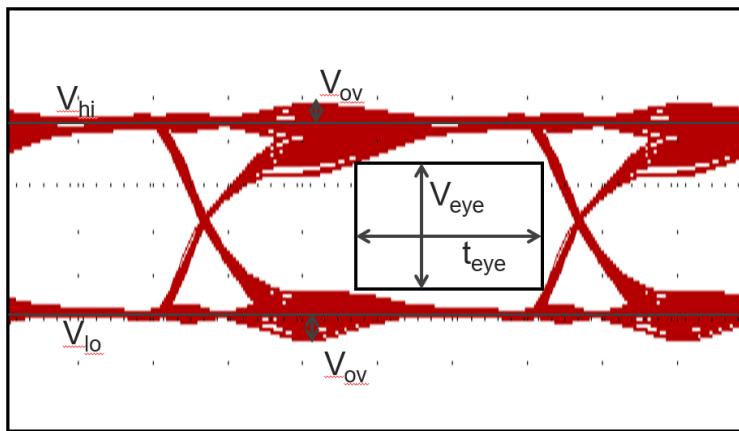
On both the Tx and Rx sides, the Link layer must include a Clock Domain Crossing (CDC) to align the data between CoreClk and PhyClk. These CDCs must also be able to absorb the slice-to-slice clock skew and core clock distribution skew across the whole link.

If DCCs are included in the PHY and they need an alignment cycle, they shall include control logic to perform self-alignment.

## 6.2. Clock and Data Specifications

### 6.2.1. Timing Requirements

Figure 14 shows the definition of the eye diagram parameters.



**Figure 14.** Eye Diagram Definitions

The CLK and data signals at the receiving slice bumps must meet the conditions in Table 10

| Symbol            | Spec  | Unterminated or Source Terminated               | Doubly Terminated                               |
|-------------------|---|---|---|
| V <sub>hi</sub>   | High signal voltage                                 | 750 mV  | 562 mV  |
| V <sub>lo</sub>   | Low signal voltage                                  | 0 mV  | 188 mV  |
| V <sub>tol</sub>  | Tolerance of V <sub>hi</sub> , V <sub>lo</sub> (5%) | +/- 37 mV                                       | +/- 19 mV                                       |
| t <sub>eye</sub>  | Data eye width                                      | 50% UI  | 50% UI  |
| V <sub>eye</sub>  | Data eye height                                     | 40%(V <sub>hi</sub> -V <sub>lo</sub> ) (300 mV) | 20%(V <sub>hi</sub> -V <sub>lo</sub> ) (75 mV)  |
| V <sub>ov</sub>   | Data and CLK overshoot                              | 25%(V <sub>hi</sub> -V <sub>lo</sub> ) (188 mV) | 50%(V <sub>hi</sub> -V <sub>lo</sub> ) (188 mV) |
| t <sub>skew</sub> | Slice to slice CLK skew                             | 100 ps/stack                                    | 100 ps/stack                                    |

**Table 10.** BoW Signal Integrity Requirements

V<sub>hi</sub> of 0.75 V must be supported by all BoW implementations, but other values may be supported.

$t_{\text{eye}}$  must be evaluated for each of the bits in a slice relative to the differential CLK+ - CLK- signal for that slice.  $t_{\text{eye}}$  must be evaluated for CLK edges up to 3 UI earlier than the eye center. This is because even though jitter on the data edges is correlated with the CLK jitter at the Tx side, the slicer in the Rx side is likely to use a different CLK edge due to delays in the Rx-side clock alignment circuit (usually a DLL). The evaluation of jitter must include all possible jitter contributors, including reference clock, clock distribution networks, any DCC, PLL and DLL jitter, power-supply noise and switching noise.

The slice to slice clock skew  $t_{\text{skew}}$  across the width of a BoW link (along the chip edge) must be less than 100 ps/stack. This is dominated by the TxClk distribution network.

Since these signals do not leave the package, these values must be verified with simulation.

If the slice implementation allows programmatic control of the DLL alignment values, varying those values after locking the DLL may provide timing margin information. If the slice implementation allows programmatic control of the receiver voltage thresholds, varying those values may provide vertical margin information.

| Spec                          | Value  | Normalized Value | Condition                                 |
|-------------------------------|--------|------------------|---|
| Bit to bit skew: fanout       | 0 ps   | 0.00 UI          | fanout for mismatched bump pitches        |
| Bit to bit skew: mismatch     | 3.6 ps | 0.014 UI         | 0.6 ps rms device mismatch (6 sigma)      |
| Bit to bit skew: clock distn  | 3.9 ps | 0.016 UI         | 0.65 mm from CLK to bits D0, D15          |
| CLK to mean data skew         | 12 ps  | 0.05 UI          | error of DLL alignment                    |
| CLK duty cycle                | 50%    | 0.50 UI          | target                                    |
| CLK duty cycle error          | 2%     | 0.02 UI          | after DCC settled                         |
| CLK+ vs. CLK- skew            | 3.6 ps | 0.014 UI         |   |
| CLK cyc-to-cyc random jitter  | TBD    |                  | thermal jitter of the whole Tx clock path |
| CLK cyc-to-cyc bounded jitter | TBD    |                  | bounded jitter of the whole Tx clock path |
| CLK rise/fall time            | 43 ps  | 0.17 UI          |   |
| Data rise/fall time           | 43 ps  | 0.17 UI          |   |
| Data Eye width                | 125 ps | 0.50 UI          | not including jitter                      |
| Data Eye height               | 450 mV | 60% Vswing       | Vswing = nominal signal voltage swing     |
| Data overshoot/undershoot     | 75 mV  | 10% Vswing       |   |

The jitter in a link should meet:

~center DataEyeWidth - BitToBitSkewFanout - BitToBitSkewMismatch  
 - BitToBitClockDistrn - CLKtoMeanDataSkew - CLKdutyCycleError  
 - CLKrandomJitter - DataToDataSkew < 0.10 UI

XX>

## 7. Electrical Specifications

### 7.1. Voltages

BoW implementations must support electrical specifications corresponding to rail to rail signaling based on a 0.75 V +/- 5% power supply as in Table 10. BoW interfaces may also support higher or lower signaling voltages but must support 0.75 V based signaling to ensure interoperability. BoW does not specify the VDD power rail voltage.

### 7.2. ESD

BOW I/O shall be designed to withstand 50 V CDM (Charged Device Model) and 250 V HBM (Human Body Model). This requirement is deemed sufficient for intra-package signalling, similar to other die-to-die interface standards.

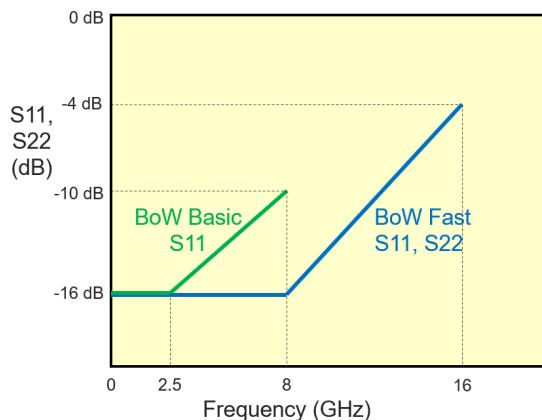
### 7.3. Termination and Return Loss

See Table 7 for the recommended termination vs. reach and mode. In most cases, on laminate or similar packages, BoW transmitters should be source-terminated to 50 +/- 8 ohms and BoW receivers unterminated.

For reach over 10 mm, BoW receivers should also be terminated in 50 +/- 8 ohms and designed for the smaller signal swings this delivers.

A terminated BoW transmitter or receiver (both data and clock) shall have 16 dB return loss (-16 dB S11 or S22) from 0 to F(CLK) and -4 dB loss at 2\*F(CLK) when outputting a logic 0, 1 or at the midscale voltage. See the trace “BoW Fast” in Figure 15 for the BoW-128 case.

Need to update Figure 15.



**Figure 15.** BoW Termination Return Loss S11 and S22

A Source-Series-Terminated (SST) transmitter can generally meet the the transmitter specification.

## 8. Chip-to-Chip Channel Specifications

The channel (wires) between chips should meet the following specs to ensure signal integrity.

### 8.1. BoW Channel Specifications on Laminate

BoW channel length on laminate is limited by the round trip reflection delay to 10 mm for 4 Gb/s/wire.

BoW channels longer than 2 mm on laminate should meet these specs:

| Parameter                                    | Value        | Comment            |
|--|--------------|--------------------|
| Length mismatch within a slice               | 1 mm         | = ~6 ps <= 0.05 UI |
| Impedance                                    | 50+/- 5 ohms |                    |
| C <sub>cross</sub> /C <sub>total</sub> ratio | < 40%        |                    |
| R <sub>series</sub>                          | < 4 ohms     |                    |

**Table 11.** BoW Clock Specifications

$C_{\text{cross}}$  is the total capacitance of a wire to all its neighboring wires.  $C_{\text{total}}$  is the total capacitance of a wire including to grounds.

Channels on interposer will have different requirements, not yet specified.

## 8.2. BoW Channel Specifications for Doubly-Terminated Links

Doubly-terminated links should meet the following characteristics.

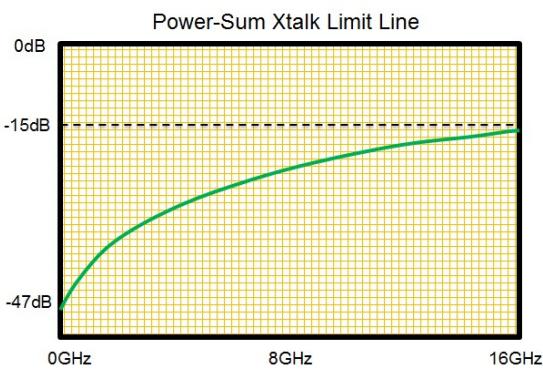
### 8.2.1. Channel Loss

To avoid the need for equalization, the channel should meet the limit in Figure 16.

**Figure 16.** BoW Doubly-Terminated Wire Channel Loss Limit

### 8.2.2. Crosstalk

The crosstalk in the channel should meet the limit in Figure 17.

**Figure 17.** BoW Doubly-Terminated Wire Crosstalk Limit

Power-sum crosstalk is the sum of crosstalk power of all aggressors on a target trace. The limit is

$$\text{XtalkLimit} = -37 * e^{-f/8\text{GHz}} - 10 \text{ dB}$$

## 9. Reset and Initialization

### 9.1. EVERYTHING AFTER THIS POINT STILL NEEDS REVISION IN THE MAR 2021 REWRITE

### 9.2. Control Options

In a system with one or more BoW interfaces, each interface pair (defined as one Tx slice on a first chiplet and one Rx slice on a second chiplet) in the system shall achieve interface ready status in each of its component slices. Once done, the interface shall signal readiness to the rest of the system. If any BoW link or slice is down (either at the link layer or the PHY level), it shall communicate this information to the appropriate interface partner as well as to the rest of the system.

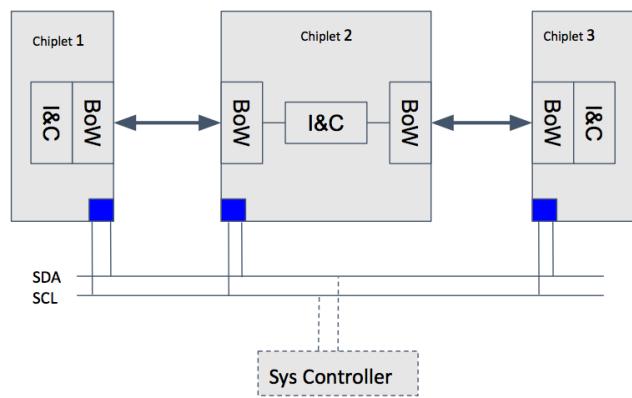
Calibration and training will require the two endpoints of an interface to exchange status and control information. There is no dedicated sideband control interface defined. Instead, this exchange shall be facilitated using an independent I2C (I3C) interface, assumed to exist outside of the BoW interface, on each chiplet. I2C(I3C) was chosen as the preferred interface for the following reasons:

- It is a two wire interface
- It is multi-point
- It supports multi-leader and multi-follower topology
- The protocol for data reads and writes is standardized
- It is widely used as a configuration/status monitoring interface
- It does not require additional control signals such as chip select (SPI, one per follower) or BSDL files (JTAG) to access register information

However, the system designer is free to choose any suitable method for their application.

An example BoW system configuration is shown in Figure 3. Any of the chiplets shown can act as an I2C (I3C) leader or follower. Alternatively, a central system controller (shown in dotted lines in the figure) shall behave as the leader and the BoW chiplets shall be followers.

To facilitate device identification and target communications at the proper device, link, and slice, each BoW Interface shall have a unique Device\_ID, Link\_ID, and Slice\_ID. A BoW interface mapping table [connection topology] should also be provided by the system designer to facilitate proper assignment of link and slice states on each of the interface partners during initialization, calibration, or other sideband activity. The specifics of how this topology information is propagated to each chiplet is left to the system designer.



**Figure 18.** Example BoW System Configuration

### 9.3. Data-Transfer Ready

A data-transfer ready signal shall be made available for control by the link layer. The data-transfer ready signal may be de-asserted due to application-driven changes, including but not limited to:

- An intentional change in clock frequency
- Receipt of bad data

De-asserting the data-transfer ready signal may also be necessary due to conditions within the BoW interface, which may include but are not limited to:

- Completion of configuration during power-up
- Initiation of reset by the link partner
- Loss of DLL lock

Internal BoW conditions indicating the need for de-assertion of the data-transfer ready signal shall be sent to the link layers so that the link layer can de-assert the data-transfer ready signal. Once data-transfer ready has been re-asserted after having been de-asserted, the BoW Adapter shall be re-calibrated (Section 11). The reverse is not true: calibration may be initiated without de-asserting data-transfer ready first.

### 9.4. Standby Mode

A signal shall be placed into standby mode by one of the following means:

- Driving the signal LO
- Putting the signal into tristate and enabling the weak pull-down. During initialization, data outputs shall be placed into standby mode.

### 9.5. Data-Transfer Ready Signals

Each slice shall have a tx\_mac\_rdy signal that is controlled by the Transmit MAC. When the tx\_mac\_rdy signal is asserted HI by the MAC, it shall indicate that the transmit slice is ready for calibration and data transfer. De-assertion of tx\_mac\_rdy shall affect only its own slice; other slices may continue transmitting data.

The tx\_mac\_rdy signal shall be forwarded to the link partner, and appropriate status and control register shall be updated, in order to inform the Receive link layer that the Transmit link layer is or is not ready for calibration.

### 9.6. The Effects of De-asserting Data-Transfer Ready

While the tx\_mac\_rdy signal is de-asserted:

[Note: because this is not a hardwired signal, there is a latency in response depending on the polling frequency of the I2C sideband interface. Need to investigate whether there is a mechanism where each chiplet can become a Leader dynamically and broadcast changes in status to its link partner]

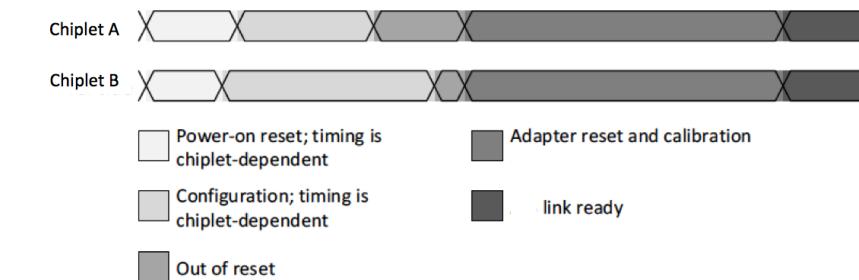
- Data transmission shall halt
- Data outputs shall be placed into standby mode (Section 9.3)
- The clock output tx\_fwd\_clk shall go into standby mode.
- The reset signal tx\_mac\_rdy shall be sent to the interface partner in order to communicate that data transmission has halted and to allow for the interface partner to be reset.

The contents of any retiming registers in the data path shall be undefined following de-assertion of data-transfer ready. De-assertion of the data-transfer ready signal shall not affect the free-running clock signals or the -control signals.

## 9.7. Initialization

Initialization will consist of three steps in sequence:

- Power-on reset synchronization
- Configuration
- Calibration



**Figure 19.** BoW Initialization

If there are multiple BoW interfaces on a single chiplet, they shall all come out of configuration at the same time, but they may complete adapter reset and calibration at different times depending on implementation.

## 9.8. Power-on Reset Synchronization

Power-on reset, being the first step in initialization, shall not require any features enabled by configuration, since configuration will not occur until after power-on reset.

## 9.9. Power-on Reset Signals

One signal (register bit) shall participate in power-on reset: `power_on_reset`.

## 9.10. Power-on Reset Sequence

During power-on reset, all input and output signals shall be placed into standby mode (Section 9.3). The power-on reset sequence shall proceed as follows:

1. If no paired device\_ID signal is detected by the Receive, then the Receive may act both to ensure that it and its chiplet are in a safe state and to alert the MAC.
2. If no paired device\_ID signal is detected by the Transmit, then the Transmit may act both to ensure that it and its chiplet are in a safe state and to alert the MAC
3. Each chiplet shall implement its own power-on reset routine. At the beginning of the routine, Receive interfaces shall assert their `power_on_reset` signals HI.
4. When a chiplet completes its power-on reset sequence:
  - a. Transmit interfaces shall begin the configuration stage.
  - b. Receive interfaces shall de-assert their `power_on_reset` signals LO and begin the configuration stage.

## 9.11. Unused Interfaces

In order to ensure correct operation for chiplets with unused Transmit interfaces, the `power_on_reset` register bits for those unused interfaces shall be set HI. In order to ensure correct operation for chiplets with unused Receive interfaces, the Device\_ID register bits for those unused interfaces shall be set to 0x0000.

## 9.12. Test Provision

In order to test the power-on reset sequence at the wafer level, two signals shall be provided for use by automated test equipment to override the power\_on\_reset and device\_detect signals when there is no Transmit/Receive pair available. por\_ovrd overrides the power\_on\_reset signal, and device\_detect\_ovrd overrides the device\_detect signal.

# 10. Configuration

Configuration may include:

- Host chiplet configuration (in the case of an FPGA or similar chiplet)
- BoW interface configuration
- BoW redundancy activation

The tx\_mac\_rdy signal shall be de-asserted LO during configuration and shall be asserted HI when configuration completes and the chiplet is ready for calibration and data transfer. The clock input from the link layershall be stable prior to assertion of tx\_mac\_rdy.

### 10.1. Output State During Configuration

All outputs, including data outputs, shall be in standby mode (Section 9.3) during configuration.

### 10.2. Chiplet Configuration

Configuration of any non-BoW aspects of the chiplet is outside the scope of this specification.

### 10.3. BoW Interface Configuration

All intended BoW features shall be configured at power-up.

### 10.4. JTAG Configuration

The chiplet data sheet should document the configuration requirements that allow for successfully implementation of JTAG EXTEST and INTEST operations.

Control Shift Register Readiness

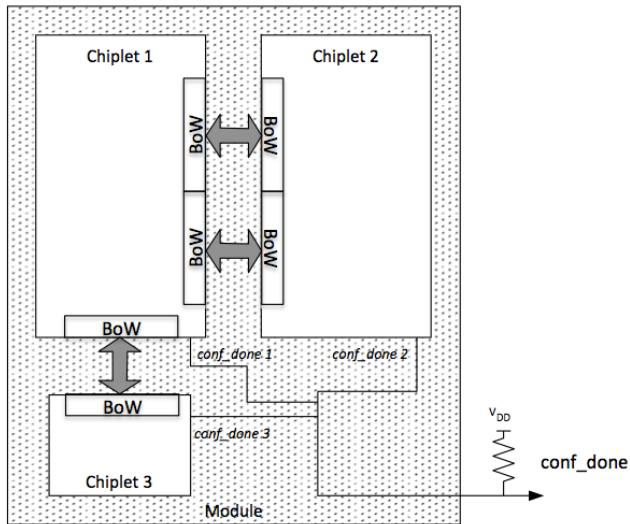
The control shift register shall be operational once configuration is complete.

### 10.5. Configuration Completion Signals

[Instead of conf\_done, can we use this mechanism as a global interrupt to allow the side-band to respond with more immediacy or retain its conf\_done status prior to calibration. Post link initialization, reuse it as a global interrupt signal. I think this can be done cleanly based on the status of a number of other signals participating in the initialization process]

Each chiplet shall have a conf\_done signal. conf\_done shall be an open-drain output. It shall be asserted LO when configuring, and it shall be released when configuration of all interfaces on the chiplet is complete, the analog circuits are stable, and the free-running clock is stable. conf\_done shall indicate only that BoW configuration is complete. No other configuration completion (MAC, FPGA, etc.) shall be included in the generation of the conf\_done signal.

All conf\_done signals from all chiplets of a module should be connected in a wired-AND configuration to generate a module-level CONF\_DONE signal that shall be HI when all chiplets on the module have completed BoW configuration. The pull-up resistor used to implement the wired-AND function may reside on the module containing the chiplets with BoW interfaces, or it may reside off the module. The CONF\_DONE signal should be provided as an output of the module regardless of the resistor placement.




---

**Figure 20.** Configuration Complete Signals

Data outputs shall remain in standby mode (Section 9.3) until CONF\_DONE is asserted, including in the case where CONF\_DONE is pulled low some time after being asserted high.

The resistance and VDD values should comply with

| Parameter          | Value |
|--------------------|-------|
| Pull-up resistance | 1kOhm |
| Pull-up Vdd        | 0.9V  |

---

**Table 12.** Wired-AND Pull-up Specifications

## 11. Calibration

The calibration sequence shall proceed as follows:

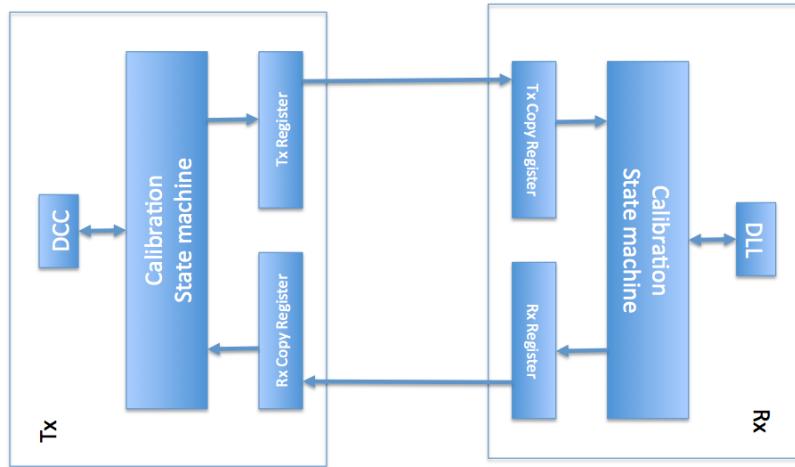
- Adapter reset
- Data path calibration

A BoW interface shall have an tx\_adapter\_rstn signal that is asserted by the MAC. It shall be forwarded to the link partner of the interface through the register interface.

When either the Transmit (tx\_adapter\_rstn) or Receive (rx\_adapter\_rstn) adapter reset signal is asserted LO, the adapter shall reset the calibration state machines. If adapter reset follows deassertion of data-transfer ready, tx(rx)\_mac\_rdy must be asserted HI before tx(rx)\_adapter\_rstn is asserted HI.

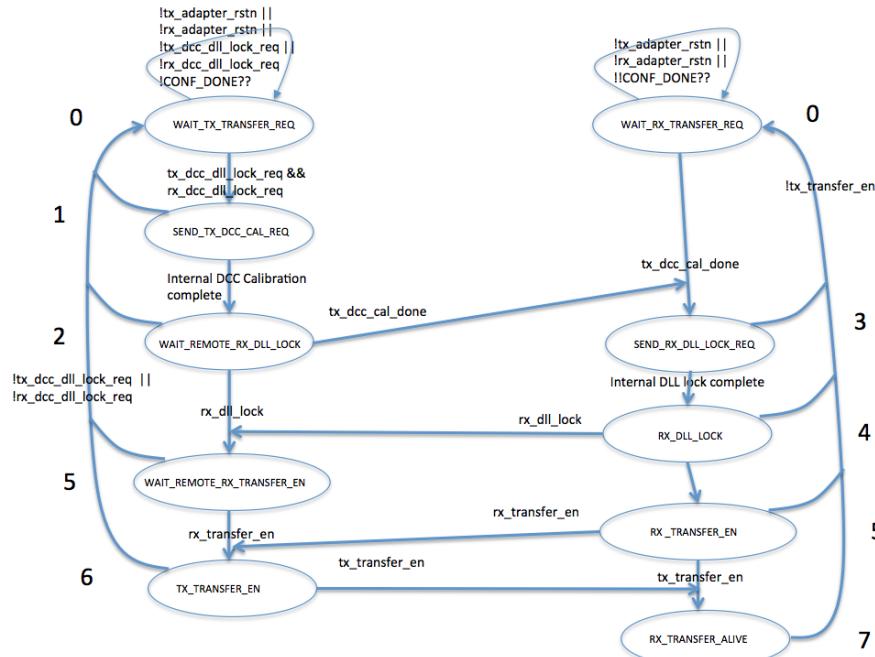
### 11.1. Data-Path Calibration

Data-path calibration shall be implemented via state machines on the Transmit and Receive sides of the interface that intercommunicate via the control signals.

**Figure 21.** Data-Path Calibration Architecture

Following the de-assertion of the adapter-reset signal(s), a calibration request shall be made by asserting a calibration request signal. Either the transmit or receive slice can initiate data path calibration.

Datapath calibration shall comply with Figure 22. The numbers in black indicate the sequence of steps

**Figure 22.** Datapath Calibration State Machine

Signals used in the datapath calibration sequence are listed in Table 13.

| Signals             | Description  |
|---------------------|--|
| rx_dcc_dll_lock_req | Request from Receive to start calibration. Once asserted, shall remain asserted until a new calibration is requested.  |
| tx_dcc_dll_lock_req | Request from Transmit to start calibration. Once asserted, shall remain asserted until a new calibration is requested.   |
| tx_dcc_cal_done     | Indicates that Transmit has completed its DCC calibration. Once asserted, shall remain asserted until a new calibration is requested.                          |
| rx_dll_lock         | Indicates that Receive has completed its DLL lock procedure. Once asserted, shall remain asserted until a new calibration is requested.                        |
| rx_transfer_en      | Indicates that Receive has completed its RX path calibration and is ready to receive data. Once asserted, shall remain asserted until calibration is complete. |
| tx_transfer_en      | Indicates that Transmit has completed its TX path calibration and is ready to receive data.  |

**Table 13.** Datapath Calibration Signals

## 11.2. Adapter Reset

Data-path calibration shall be initiated when the link layer asserts the tx\_adapter\_rstn signal LO. If the data-transfer ready signal was de-asserted prior to the start of calibration, then the tx\_mac\_rdy signal must be asserted HI prior to asserting the adapter-reset signals HI. The link layer must de-assert the adapter-reset signal prior to requesting calibration start.

## 11.3. Calibration Request

Calibration can be requested by either the Transmit or the Receive slice using the tx\_dcc\_dll\_lock\_req signal or the rx\_dcc\_dll\_lock\_req signal, respectively.

| Calibration Initiator | Dataflow direction  | Initiation signal   |
|-----------------------|---------------------|---------------------|
| Transmit              | Transmit to Receive | tx_dcc_dll_lock_req |
| Receive               | Receive to Transmit | rx_dcc_dll_lock_req |

**Table 14.** Calibration Initiation Signals

Calibration for a dataflow direction shall commence when either Transmit and Receive side has asserted its calibration request signal for that dataflow direction. Calibration request signals shall remain asserted until a new calibration is requested.

## 11.4. DCC Calibration

Upon receipt of an xx\_dcc\_dll\_lock\_req signal, the DCC shall be calibrated. The means of calibration is not specified and is left to the designer. If the optional DCC is not present, then the state machine in Section 11.1 shall remain the same, with the DCC calibration state serving only to provide a signal indicating DCC calibration completion.

## 11.5. DLL Calibration

Following DCC calibration, the receiving DLL shall be calibrated. The means of calibrating the DLL is not specified and is left to the designer. If the optional DLL is not present, then the state machine in Section 11.1 shall remain the same, with the DLL lock state serving only to provide a signal indicating DLL lock completion.

## 11.6. Calibration Completion

Calibration completion shall be indicated by the following signals. Full completion shall be indicated when all four signals are asserted HI. All four signals, once asserted, shall remain asserted until a new calibration sequence is requested.

| Calibration Completion Signal | Meaning  |
|-------------------------------|--|
| tx_transfer_en                | Transmit transmit block has completed calibration. |
| rx_transfer_en                | Receive receive block has completed calibration    |

**Table 15.** Calibration Completion Signals

## 11.7. BoW Link Ready

When both tx\_transfer\_en and rx\_transfer\_en are true, then the link shall be ready to transmit data.

## 11.8. Link Training

Link training will be addressed in a future revision of the spec

## 12. Control-Signal Shift Register Mapping

The MDIO interface registers shall be fully documented in the PHY datasheet.

## 13. Testability

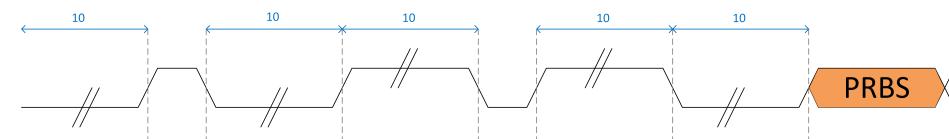
### 13.1. Test Pattern

Suggested test patterns are:

- PRBS-9 Pattern, defined by polynomial of  $X^9 + X^5 + 1$
- PRBS-31 Pattern, defined by polynomial of  $X^{31} + X^{28} + 1$

Furthermore, to cover the DC wandering and single bit response, the following suggested pattern should be added to the beginning of the preferred PRBS pattern.

[‘0’] X 10 + ‘1’ + [‘0’] X 10 + [‘1’] X 10 + ‘0’ + [‘1’] X 10 + [‘0’] X 10



**Figure 23.** Stress Test Pattern

### 13.2. Loopback Test

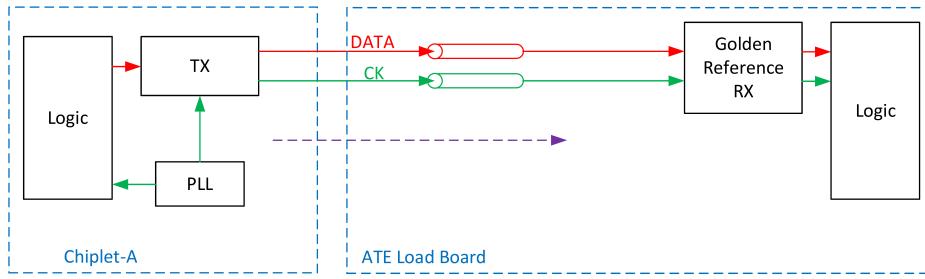
A BoW interface will be used for loopback testing in two use cases: at wafer-sort time for chiplet test for full-system bring-up, and debug validation.

Wafer sort tests are currently only practical for the BoW interface with regular bump pitches (~130um), where ATE (automatic testing equipment) probe boards with matching pin pitches are

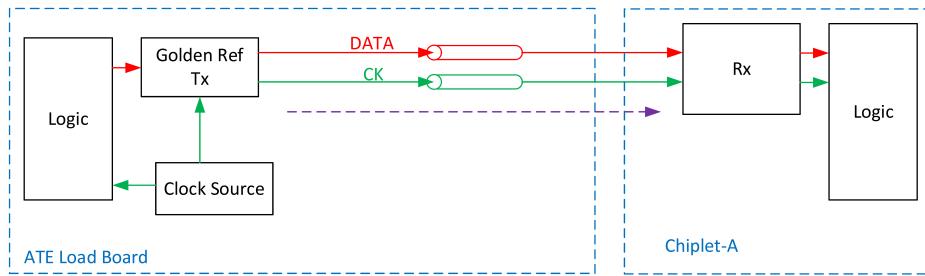
available. Microbump probes will require additional effort.

Unidirectional links will need open-loop testing. In Tx-Open-Loop testing, shown in Figure 24, Chiplet-A transmits a known test pattern (PRBS9 or PRBS31) to a golden reference receiver through the ATE load board. The received pattern is verified in the ATE load board.

Rx-Open-Loop testing, shown in Figure 25, is used for a link where the DUT is only a receiver. A golden reference Tx transmits a known pattern (PRBS9 or PRBS31) through the channel to the chiplet. The received pattern will be analyzed for quality and functional tests.



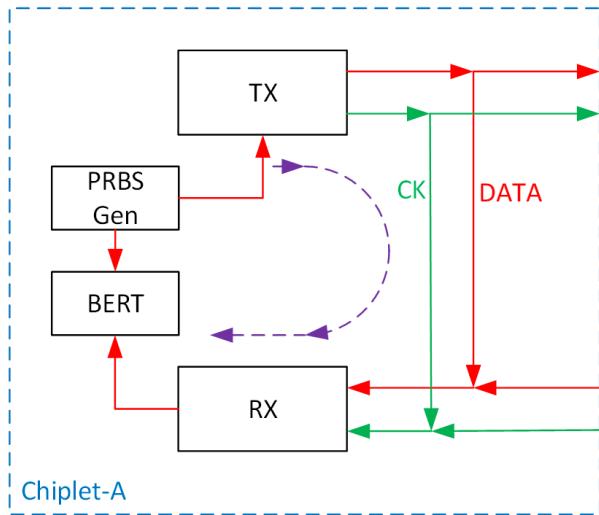
**Figure 24.** Open loop Tx chiplet testing



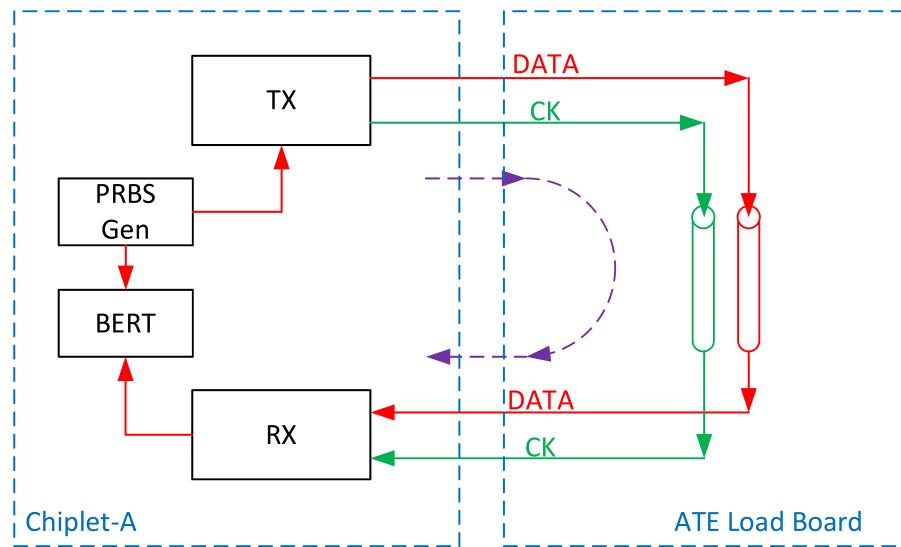
**Figure 25.** Open loop Rx chiplet testing

In bidirectional links, loopback tests can be implemented in two modes:

- short loopback mode: Data is looped back within the chip (shown in Figure 26). The short loopback can be triggered by the ATE.
- long loopback mode , the PRBS pattern is generated by chiplet-A, sent over the replica channel on the ATE load board which loops it back (shown in Figure 27). The received pattern will be passed to a bit error rate tester (BERT) to analyze the performance of the link with off-chip data and clock wires.



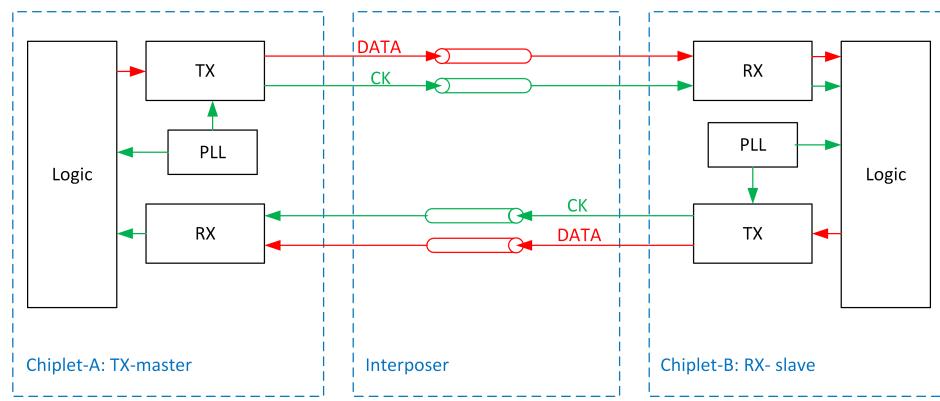
**Figure 26.** Short loopback testing



**Figure 27.** Long loopback testing

Both loopback modes can potentially be used for in-field validation bring-up and test. Cooperation across chiplets will be required to execute these tests in the field. Open-loop testing requires the use of a fixed test pattern recognized by both ends and is the only option for unidirectional links. Long loopback mode can be implemented on interposer or organic laminate for validation/verification purposes.

Figure 28 shows how a long loopback mode is executed across two chiplets for in-field validation and test where Tx and Rx are in different chiplets. Furthermore, this configuration can be expanded to loop back the data from the transmitter of chiplet-A to the receiver of chiplet-A.



**Figure 28.** Chiplet-to-chiplet long loopback

Created with [Madoko.net](#).