OPEN

Compute Project

Bunch of Wires PHY Specification

The Open Domain-Specific Architecture BoW Workstream

This document is a draft and not an approved specification. It is to be released under the following OCP copyright release when completed and approved:

{ page-break-before: always }

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

Bunch of Wires PHY Specification

Bunch of Wires PHY Specification

The Open Domain-Specific Architecture BoW Workstream**Bunch of Wires PHY Specification**

The Open Domain-Specific Architecture BoW Workstream

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

Section	Description
2.3	BoW Modes
3.1	Chip to chip signals (wires)
5.1	Wire order
	Much more....

Table 1. BoW Compliance Summary

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:

Do we also need a discussion of interoperability?

2. BoW Overview

This section provides an overview of the BoW physical interface (PHY) and its use in a multi-chiplet design.

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 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. The BoW PHYs between two die are physically connected through wires on a

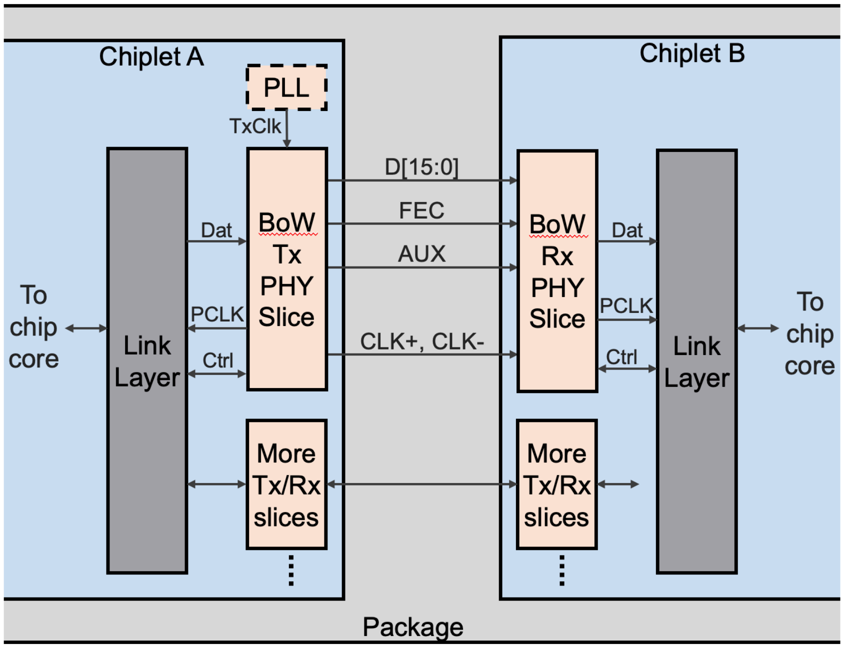

Figure 1. BoW Overview

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.

This document specifies the protocol for a BoW PHY slice. The aggregation of multiple PHYs into a link is beyond the scope of this document.
A BoW PHY slice either transmits or receives 16 bits of data between die. The BoW is a source-synchronous PHY and 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 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, impedence 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.

2.3. BoW Modes

A BoW PHY must operate in one of the Bow Modes listed in ascending order in Table 2. A 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.

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

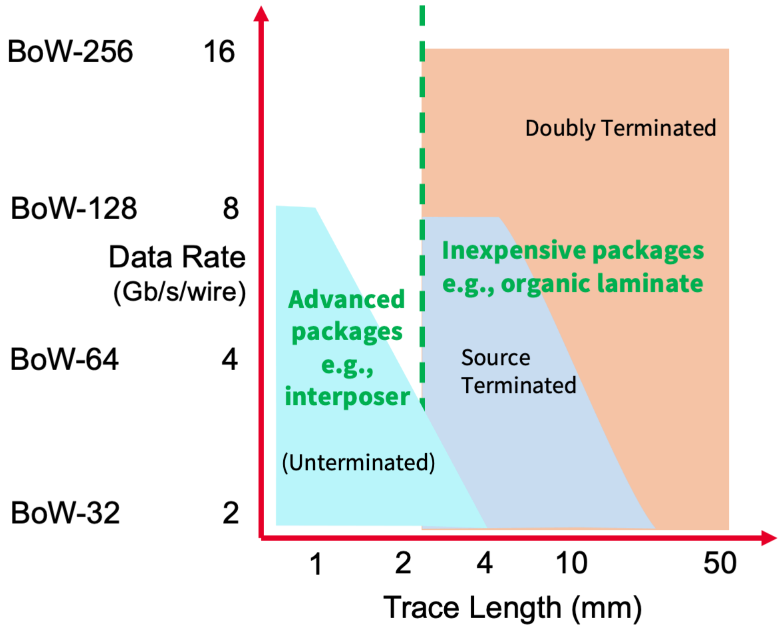

Figure 2. BoW Data Rate vs. Reach tradeoff

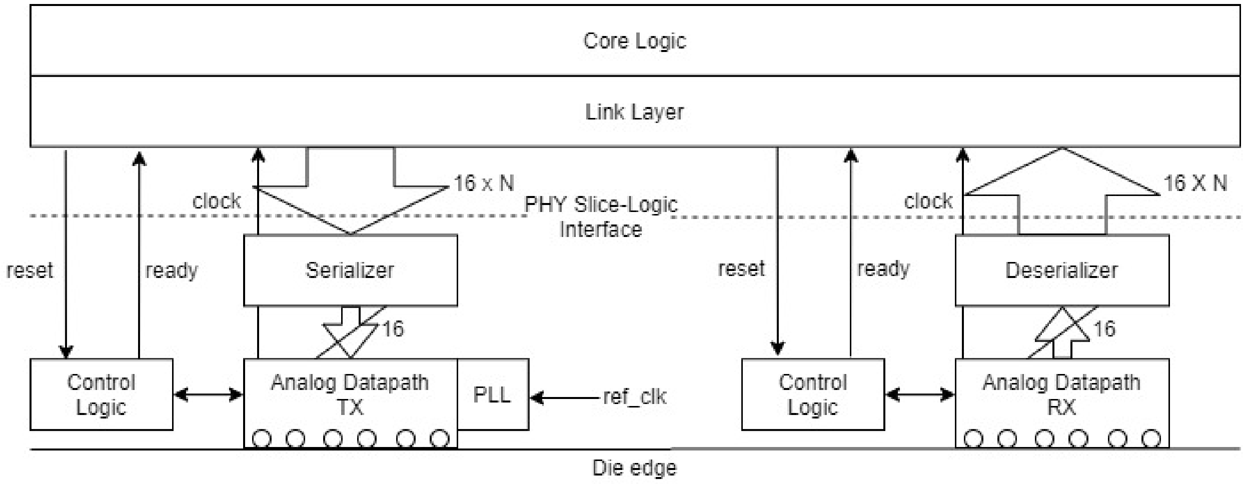

Figure 3. BoW slice logic interface

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.

2.4. PHY - Link Layer Interface

Figure 3 shows the interface between a BoW slice and the digital link layer logic in a chip. 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).

3. Signal Definitions

This section specifies the control data signals into and out of device logic and package for BoW Rx and Tx slices.

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 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

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

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.

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.

DBIcurrent = ((data[15]current XOR data[15]prev) + (data[14]current XOR data[14]prev) ... + (data[1]current XOR data[1]prev) + (data[0]current XOR data[0]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. Slice Logic Interface

Figure 3 shows the data and control signals in the interfaces to the logic in the die in each BoW transmit and receive slice. The data at the slice logic interface must be DDR (bit rate equal to twice the PCLK frequency). The BoW PHY is expected to be used with a link controller. The transmission of logical information across a slice and the control and management of a slice are the responsibility of the link controller.

3.2.1. Slice Logic Interface: Data Signals

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

3.2.2. Slice Logic Interface: Control Signals

A BoW interface must provide the following control and status signals:

- PHY Ready

Signal	# Bits	Tx Slice	Rx Slice	Description
Data	16	In	Out	Data
FEC	1	In	Out	Forward Error Correction (optional)
AUX	1	In	Out	Auxiliary uses (optional)
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

Signal	# Bits	Tx Slice	Rx Slice	Description
PHYReset	1	In	In	Resets the BoW slice
PHYReady	1	Out	Out	1 indicates that the PHY is ready to transmit/receive mission r

Table 5. Link Layer Interface Control Signals

- Selective reset

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

3.2.2.1. PHY Reset The PHYReset pin is asserted by the link controller to initialize the PHY. The PHYReset signal shall allow internal self-alignment to take place. The reset states are otherwise implementation-dependent and shall be documented in the datasheet of a particular implementation.

3.2.2.2. PHY Ready PHY Ready is asserted by the PHY to indicate to the controller that it is ready to transmit/receive data.

3.2.2.3. PHY Ready Tx On Tx, PHY ready asserted indicates PHY is transmitting good clock and ready to transmit data

3.2.2.4. PHY Ready Rx On Rx PHY ready assumes clock received is good when PHY reset is deasserted. After the Rx slice clock self-alignments are complete, each Rx PHY slice shall assert its PHYReady pin. 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. PHY Ready asserted indicates that any data received will be captured correctly.

3.2.3. Programming

There shall be an AMBA APB programming interface to control internal registers for control and status readout of the PHY.

The internal registers are implementation-dependent. The internal registers shall be fully documented in the PHY datasheet.

			Laminate	**Laminate**	**Laminate**	**Advanced**
			Unterminated	Source Terminated	Doubly Terminated	Unterminated
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 6. Recommended BoW Wire Reaches

3.2.4. Link Controller

There shall be a Link Controller (LC) outside the PHY. This will manage initialization of the Link. It may reside on one of the chiplets of the Link, in a third chiplet in the package or outside the package.

Communication from the Link Controller to the PHY slices shall be by a transport mechanism outside the BoW link. This could be a serial link like SPI or I2C, but this is not specified at this time.

Link initialization is described in Section 10.

4. BoW Modes

A BoW interface must conform to one of the Bow Modes seen in Table 2. The recommended maximum wire reach for different packaging types and terminations is seen in Table 6. 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.

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

5. BoW Physical Configuration

5.1. BoW Components

A BoW link between two chiplets is made up of wires, slices, and stacks as seen in Figure 4.

- 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.

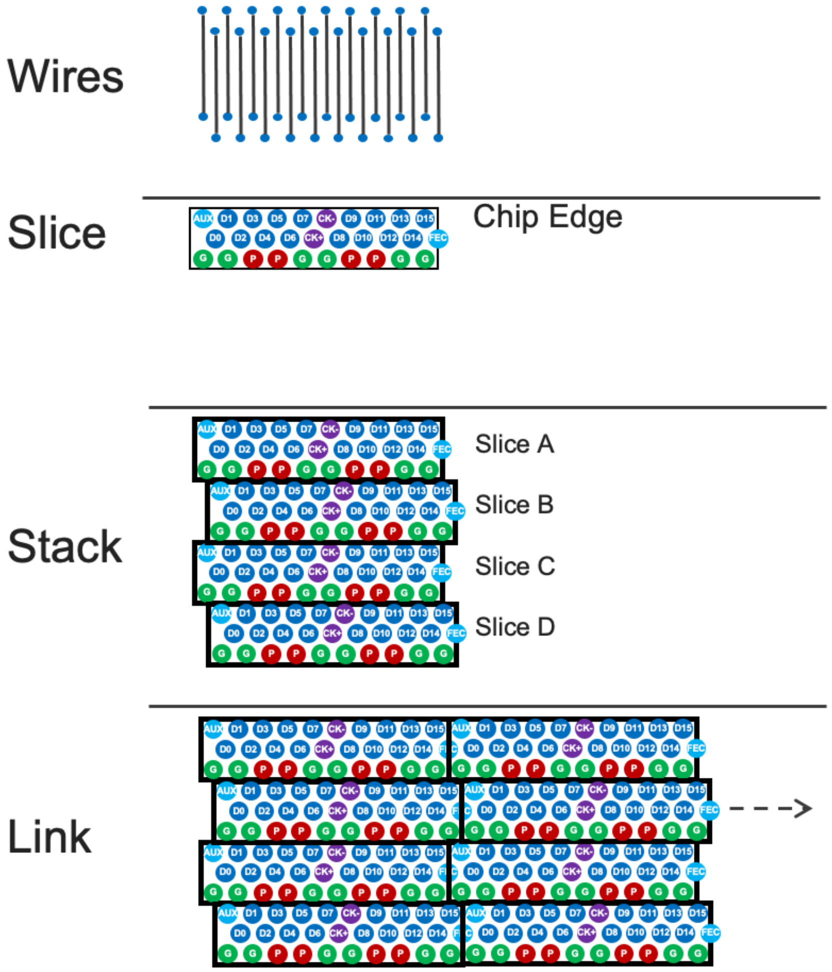

Figure 4. BoW Link Components

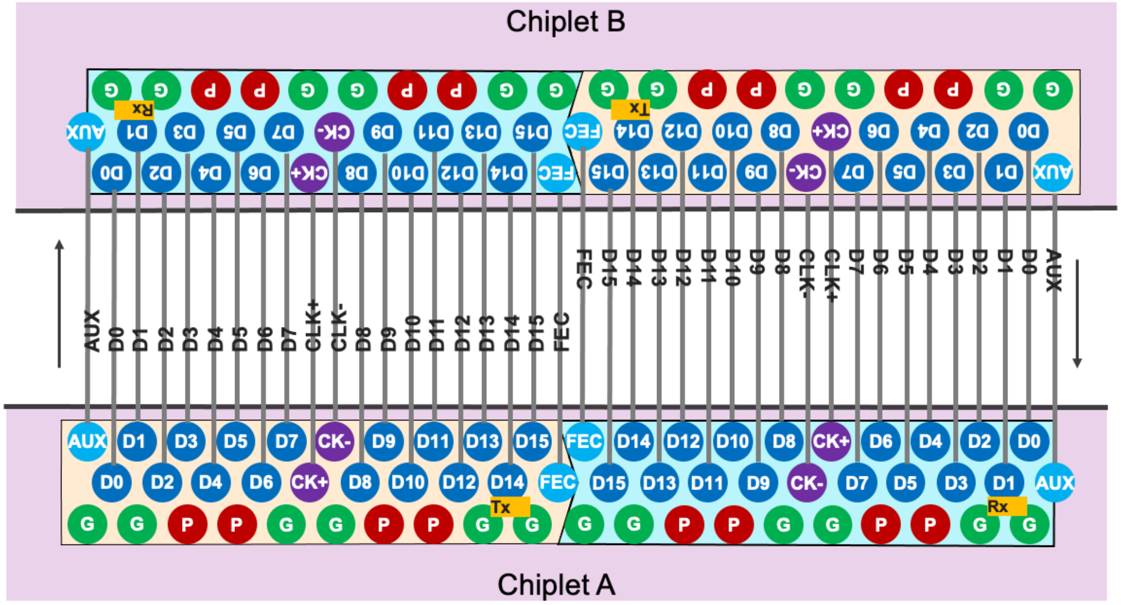

Figure 5. BoW Minimal Bidirectional Reference Link

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 7. BoW Die-to-Die Signals

- 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 5.
In this example, each chiplet has one Tx slice and one Rx slice, arranged in two one-slice stacks on each chiplet.

5.3. 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. Signal Ordering

A BoW interface must conform to these wire 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 same clockwise/counter-clockwise ordering is used on all four sides of a chiplet
- The AUX and FEC signals may be omitted

5.5. Bump Arrangements

Note that bump patterns are not specified by BoW; only the signal *ordering* at the chip edge is specified for interoperability.

The reference example in Figure 5 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.

5.5.1. Alternate Bump Arrangements

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 without changing the bump pitch)
- different ordering of power and ground bumps
- multiple power and ground rows

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

5.6. Cross Section

A cross section for an organic laminate (a.k.a. “buildup”) package is shown in Figure 6.

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 (a 7-2-7 laminate can support 4 slices but

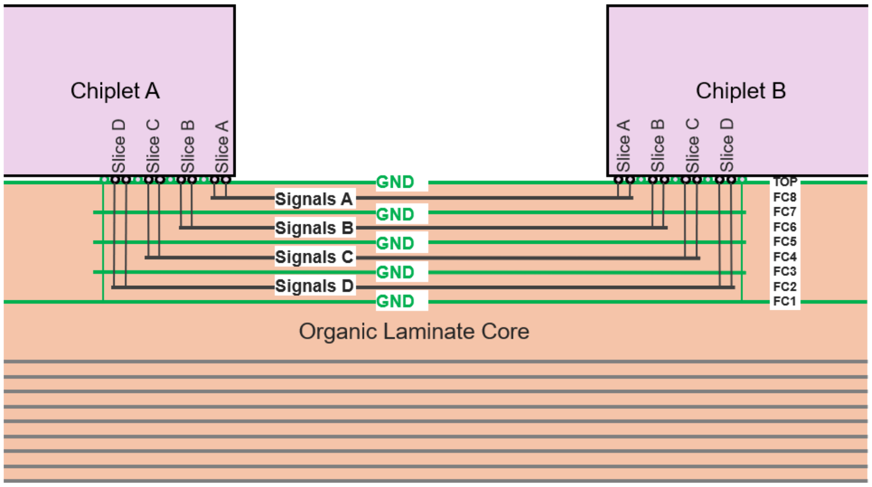

Figure 6. Cross section of a BoW Link in an Organic Laminate Package

with reduced signal integrity). 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. The wire order within each slice must be maintained, even if interleaving with other slices is used.

5.7. Staggered Slices

To optimize the density of hexagonal bump arrays, slices in positions B and D may be offset horizontally by one half the bump pitch as seen in Figure 7. This necessitates a one-bump-pitch horizontal jog in the wires for slices B and D. The practical effect of this 130-um jog across a 2.5+ mm wire between chiplets is very small.

An alternative arrangement is to keep the slices aligned vertically. This requires adding a small extra vertical space between the slices, for an overall increase of 4% of the slice area.

5.8. Slice Numbering

A BoW interface must conform to these slice numbering rules:

- 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.

An example of this numbering is shown in Figure 8.

The signal ordering and slice numbering rules allow BoW chiplets to be connected without signal reordering regardless of chiplet rotations.

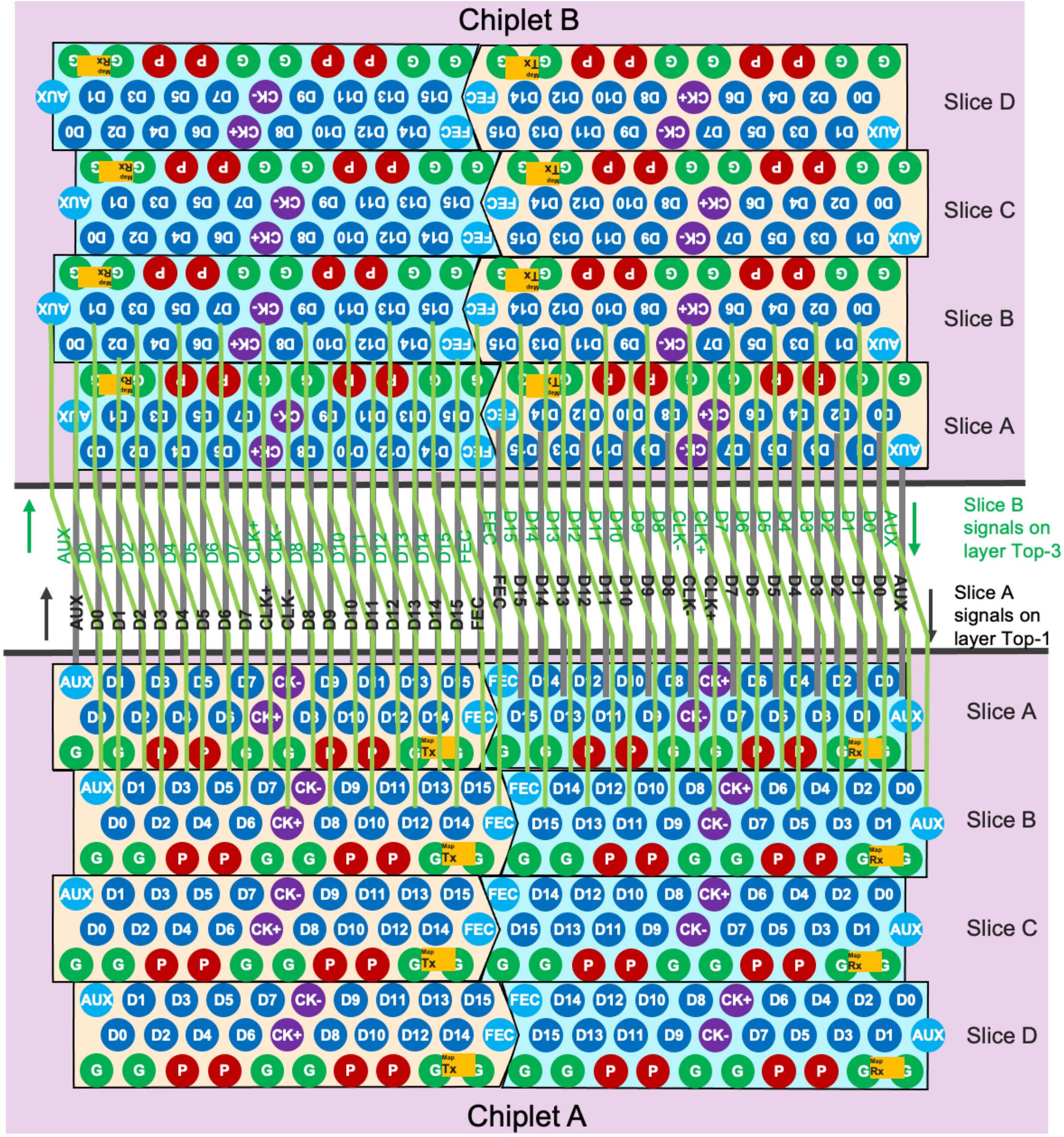

Figure 7. Staggered slices for the densest bump packing

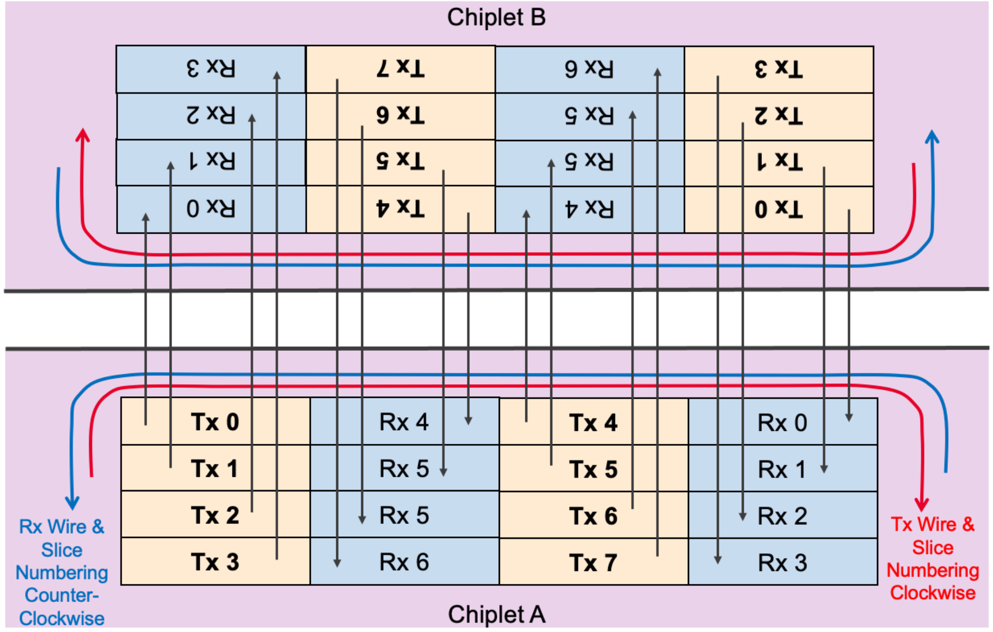

Figure 8. Alternating-Stacks Pattern of Tx and Rx Slices in a Link

5.9. Slice Stacking Pattern for Symmetric Links

For bidirectional links, a pattern of alternating Tx and Rx stacks should be used. Figure 8 shows a bidirectional link with 4 stacks of 4 slices each, for 8 Tx and 8 Rx slices on each chiplet.

Asymmetric and unidirectional links may use any slice pattern, but the slice numbering rules must be observed.

An alternate approach with more flexibility is to design every slice to operate as either Rx or Tx, to be configured after assembly or upon powerup. This allows complete flexibility in link configuration and interoperability and also provides an opportunity for wafer-test loopback testing. In this case, number the slices as if they are all Tx slices.

In BoW-256 at 16 Gb/s/wire, the link in Figure 8 provides a total of 2.0 Tb/s in each direction. In an organic substrate using the hexagonal bump pattern of Figure 5 with a bump pitch is 130 um, the total edge width is 5.2 mm (4.16 mm without AUX and FEC); the depth from the edge is 1.35 mm. In an interposer, if the bump pitch is 40 um, the edge width is 1.60 mm (or 1.28 mm) and the depth is 0.42 mm.

6. Timing

6.1. Clocking

Figure 9 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.

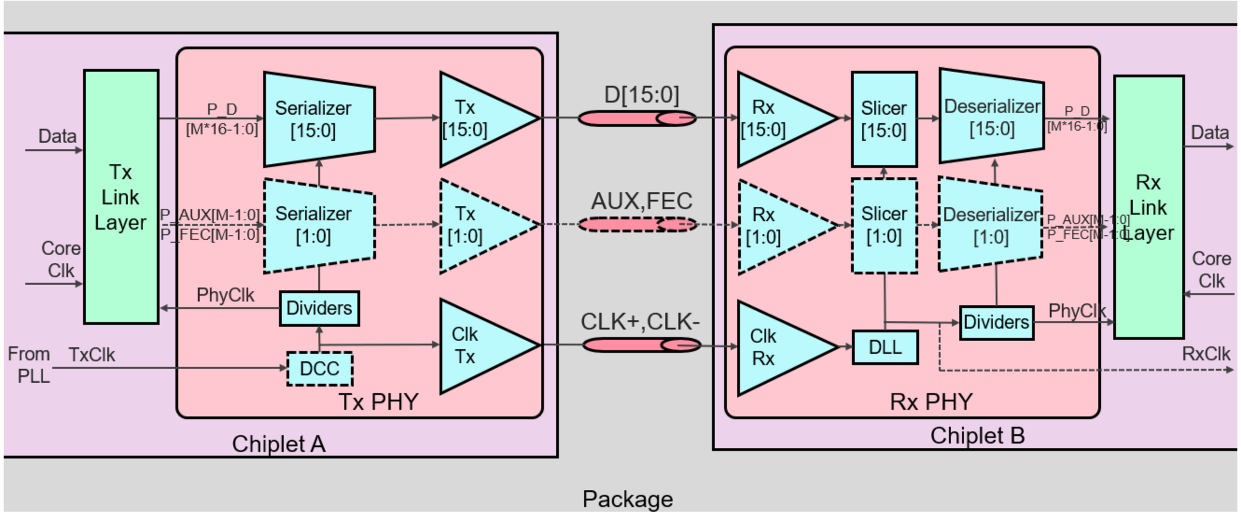

Figure 9. BoW Clock and Data Block Diagram - One Tx Slice, One Rx Slice

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

Table 8. Example Clock and Data Rates for Figure 9 with 4 Gbps, M=4

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.

The ratio M should be limited to integers, preferrably 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.

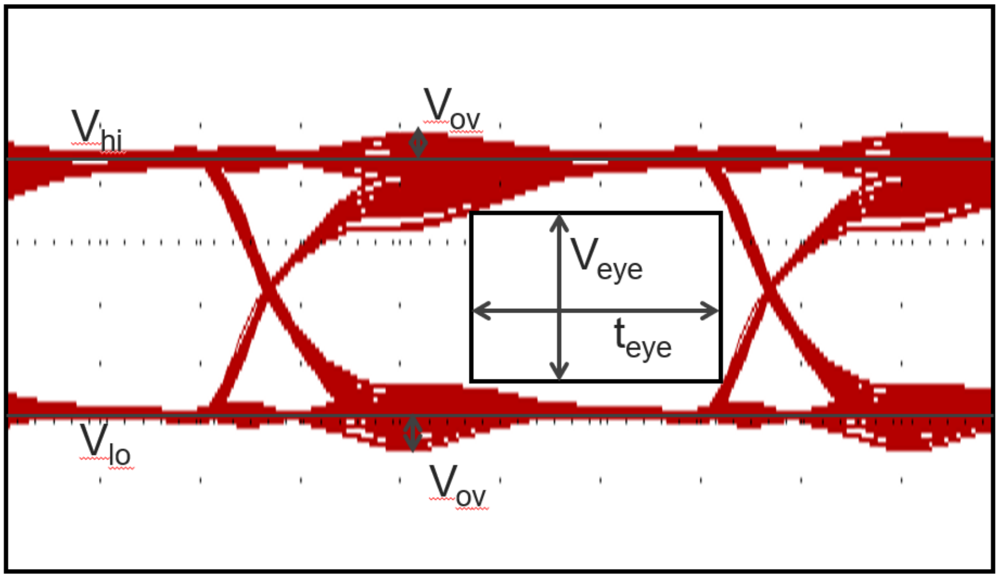

Figure 10. Eye Diagram Definitions

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 10 shows the definition of the eye diagram parameters.

The CLK and data signals at the receiving slice bumps must meet the conditions in Table 9

V_{swing} of 750 mV must be supported by all unterminated or source-terminated BoW implementations, but other values may be supported. V_{swing} of 375 mV must be supported by doubly-terminated implementations. BoW transmitters which support double termination must support termination to 750, 375 or 0 mV.

t_{eye} must be evaluated for each of the bits in a slice relative to the differential CLK+ - CLK- signal for that slice. t_{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

Symbol	Spec	Unterminated or Source Terminated	Doubly Terminated
V_{hi}	High signal voltage	750 mV	
V_{lo}	Low signal voltage	0 mV	
V_{swing}	Signal Swing	750 mV	375 mV
V_{tol}	Tolerance of Vhi, Vlo (5%)	+- 37 mV	+- 19 mV
t_{eye}	Data eye width	50% UI	50% UI
V_{eye}	Data eye height	40%(V_{hi}-V_{lo}) (300 mV)	20%(V_{hi}-V_{lo}) (75 mV)
V_{ov}	Data and CLK overshoot	25%(V_{hi}-V_{lo}) (188 mV)	50%(V_{hi}-V_{lo}) (188 mV)
t_{skew}	Slice to slice CLK skew	100 ps/stack	100 ps/stack

Table 9. BoW Signal Integrity Requirements

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_{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.

7. Electrical Specifications

7.1. Voltages

BoW implementations must support signal swings as specified in Table 9. BoW interfaces may also upport 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 6 for the recommended termination vs. reach and mode. For lower rates on laminate or similar packages, BoW transmitters should be source-terminated to 50 +/-8 ohms and BoW receivers unterminated.

For rates of 16 Gb/s or reach over 10 mm, BoW receivers should also be terminated in 50 +/-8 ohms and designed for the smaller signal swings this delivers.

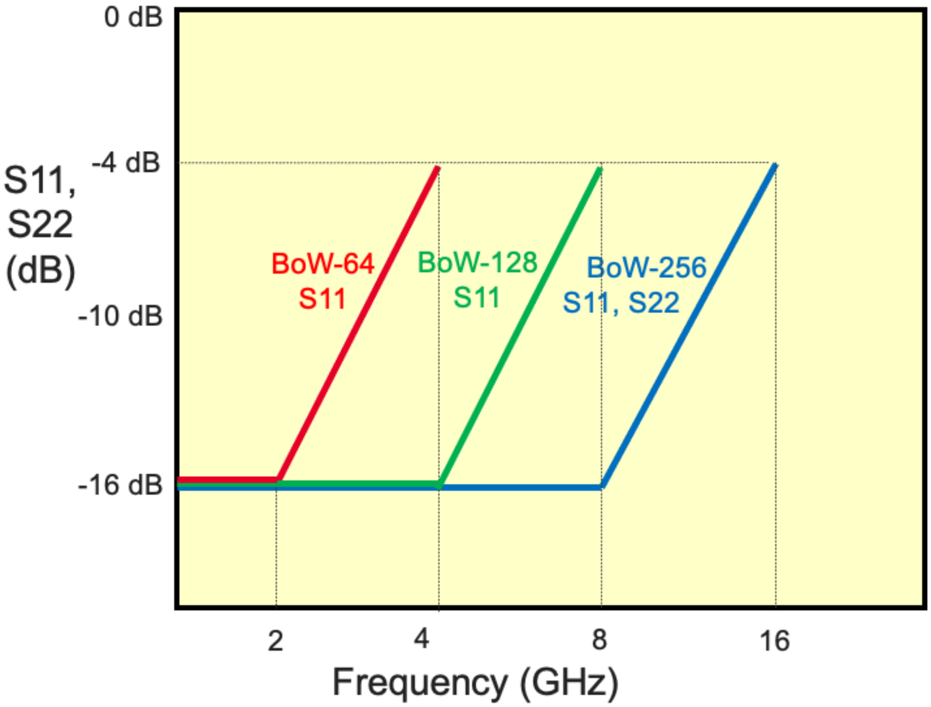

Figure 11. BoW Termination Return Loss S11 and S22

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 Figure 11.
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:

C_{cross} is the total capacitance of a wire to all its neighboring wires. C_{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.

Parameter	Value	Comment
Length mismatch within a slice	1 mm	= ~6 ps <= 0.05 UI
Impedance	50+-5 ohms	
C_{cross}/C_{total} ratio	< 40%	
R_{series}	< 4 ohms	

Table 10. BoW Clock Specifications

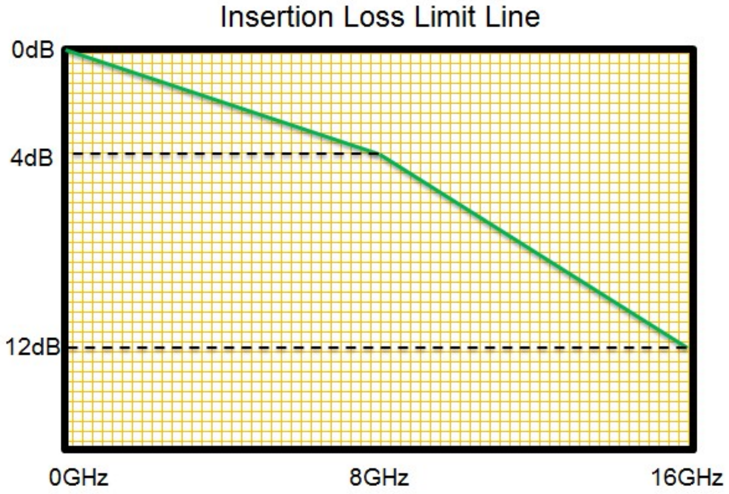

Figure 12. BoW Doubly-Terminated Wire Channel Loss Limit

8.2.1. Channel Loss

To avoid the need for equalization, the channel should meet the limit in Figure 12.

8.2.2. Crosstalk

The crosstalk in the channel should meet the limit in Figure 13.
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. BoW in an ODSA Design

Chiplet-based designs require logical connectivity between the die in a single package, in addition to physical connectivity. This section provides an overview of how with the Open Domain-Specific Architecture stack it can be used as an underlay for popular transaction protocols.

9.1. 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 14 to enable chiplets from multiple vendors to interoperate and be integrated in a multi-die package. The Bunch of Wires

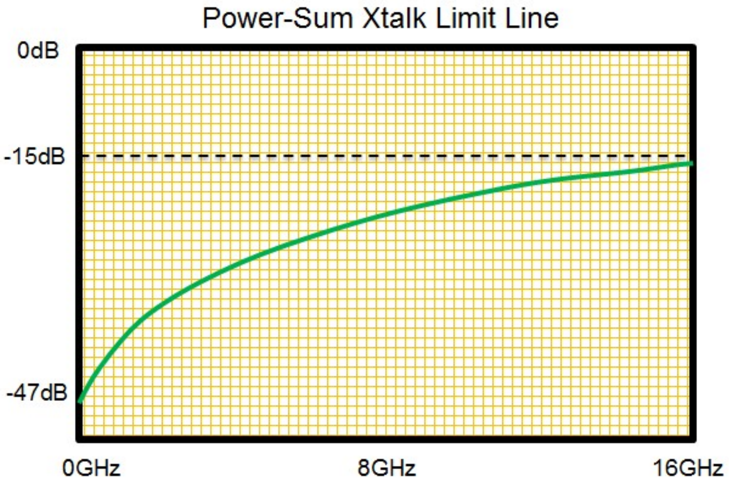

Figure 13. BoW Doubly-Terminated Wire Crosstalk Limit

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.

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 14, 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 15 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 15 - these are part of the PHY.

10. Reset and Initialization

10.1. External Facilities

These facilities must be provided outside the PHY:

- a Link Controller (LC) which will manage initialization of the Link. It may reside on one of the chiplets of the Link, in a third chiplet in the package or outside the package.
- A communication path from the Link Controller to the PHY slices outside the BoW link. This could be a serial link like SPI or I2C, but this is not specified at this time.
- A source of training pattern data outside the PHY, assumed to be the Link Layer here. This must be able to repetitively transmit an arbitrary 288-bit pattern (16 bits per wire) required by the Rx slice for clock alignment as specified in the datasheet for the Rx PHY.
- The Reset input to each PHY shall be asserted upon powerup. It may also be asserted by

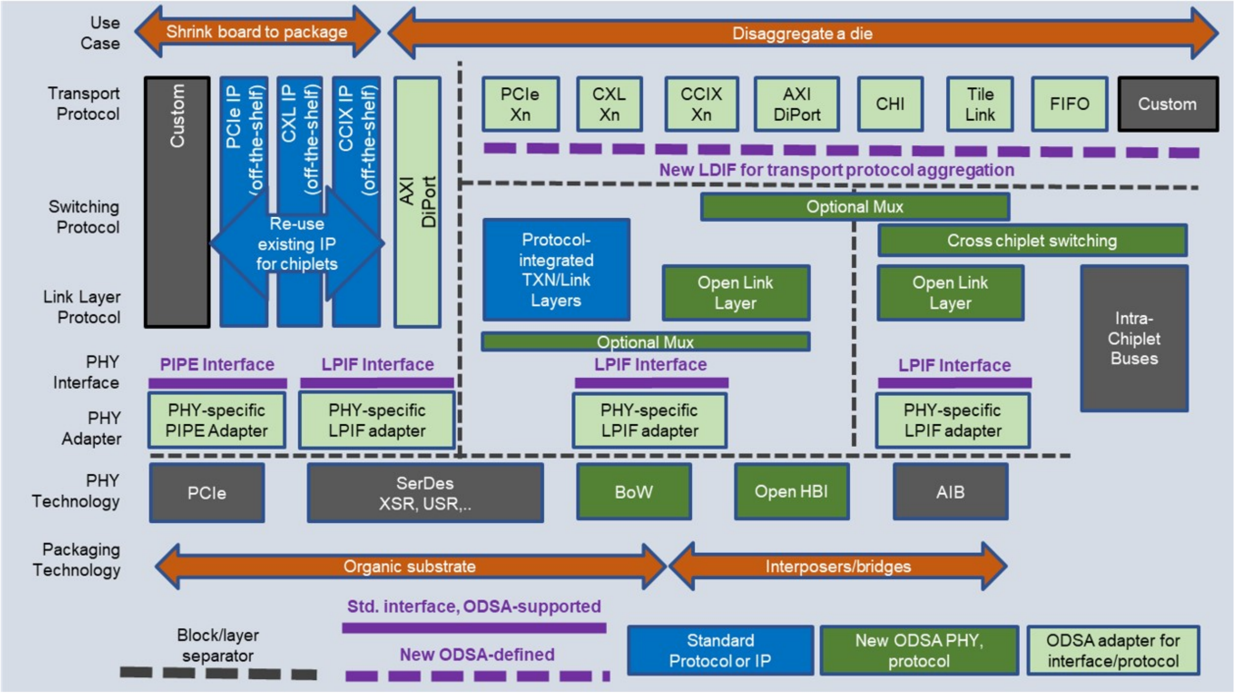

Figure 14. The BoW PHY in the ODSA Stack

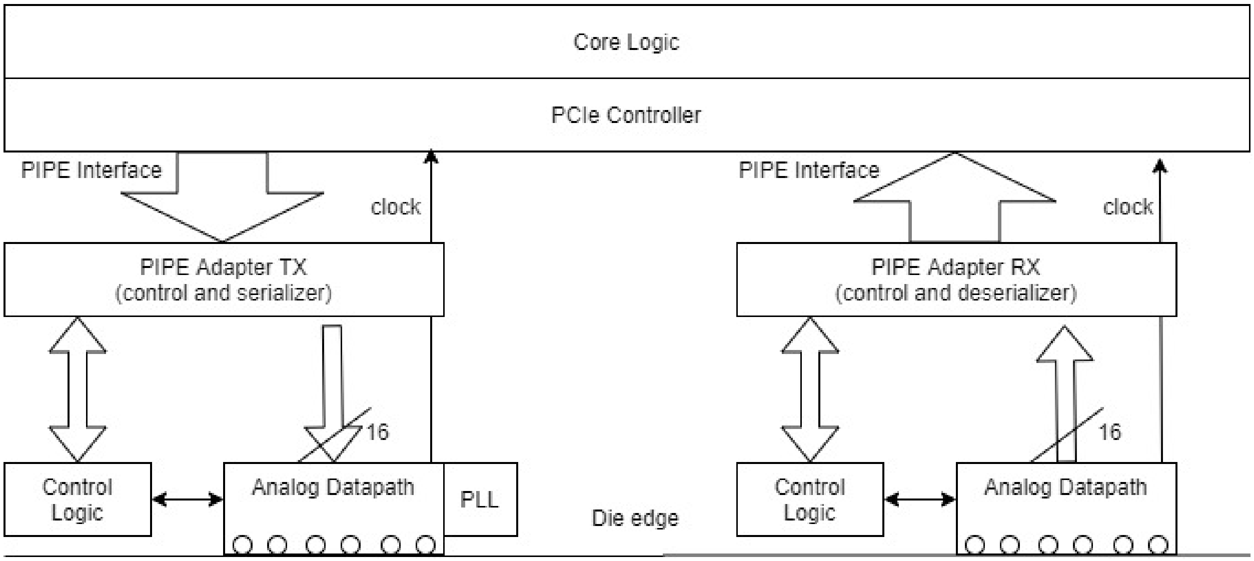

Figure 15. BoW with a PIPE adapter for PCIe transactions

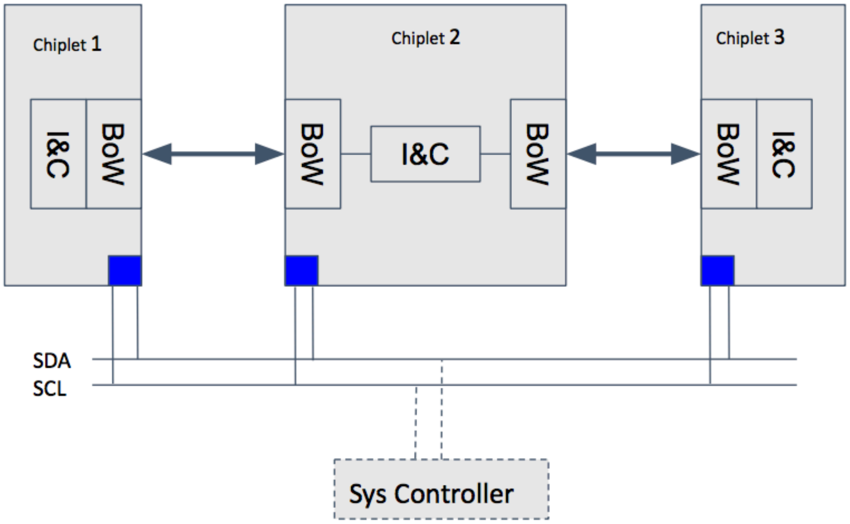

Figure 16. Example BoW System Configuration

commands from the LC.

An example topology is shown in Figure 16.

10.2. Initialization Sequence

A Tx-Rx Link shall be brought up as follows:

1. The Link Controller (LC) performs any needed configuration of the PHY slices via the APB interface. This is implementation dependent
2. The LC de-asserts Reset to the Tx PHY slices
3. Once its Tx clock stabilizes, each Tx PHY slice asserts Ready to the LC
4. When all Tx PHY slices are ready, the LC signals the Tx Link Layer to send the clock training pattern specified by the Rx PHY
5. The LC de-asserts Reset to the Rx PHY slices
6. Each Rx PHY slice performs clock and data alignment and signals Ready to the LC when done
7. When all Rx PHY slices are ready, the LC signals the Tx and Rx Link Layers that they can proceed with channel bonding

Not specified:

- Whether the up and down links are initialized one at a time or in parallel
- How the signals from the the LC get from the I2C to the PHY Ready and Reset pins of the PHY
- How the Link Layer performs channel bonding or start of data transmission
- Any PHY registers required to implement this process - these are an implementation choice.

10.3. Other Unspecified Areas

- There is no low-power standby mode defined.
- There is no specification for when a PHY should de-assert its Ready pin. PLL or DLL losing lock are possible causes.
- There is no definition of what occurs if the PHY does de-assert Ready
- There is no definition of what should be done with unused PHYs (that are on the chip but have no partner on another chiplet)
- There is no definition of addressing chiplets, Links or slices

11. Configuration

PHY configuration is implementation dependent. It may include:

- Tx vs Rx for configurable slices
- PLL, DLL, DCC or similar circuit configuration

PHY configuration may be hardwired in the chiplet implementation, or it may be programmable.

11.1. Link Training

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

12. Control Register Mapping

The interface control registers are implementation dependent. The registers shall be fully documented in the PHY datasheet.

13. Testability

13.1. Test Patterns

- Should we make any or all of these test patterns mandatory to facilitate interoperability?
- Syncing a receiver to a PRBS-31 pattern is slow without an agreed start time. How do other specs do this?

Test patterns should be programmable to repeat indefiniately.

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

- Isolated 1 and 0 pattern to test DC wander and single bit response:

  - [‘0’] X 10 + ‘1’ + [‘0’] X 10 + [‘1’] X 10 + ‘0’ + [‘1’] X 10 + [‘0’] X 10
  - This can be prepended to a PRBS pattern:

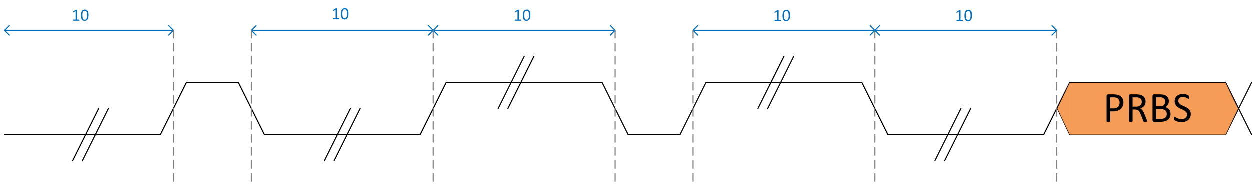

Figure 17. Stress Test Pattern

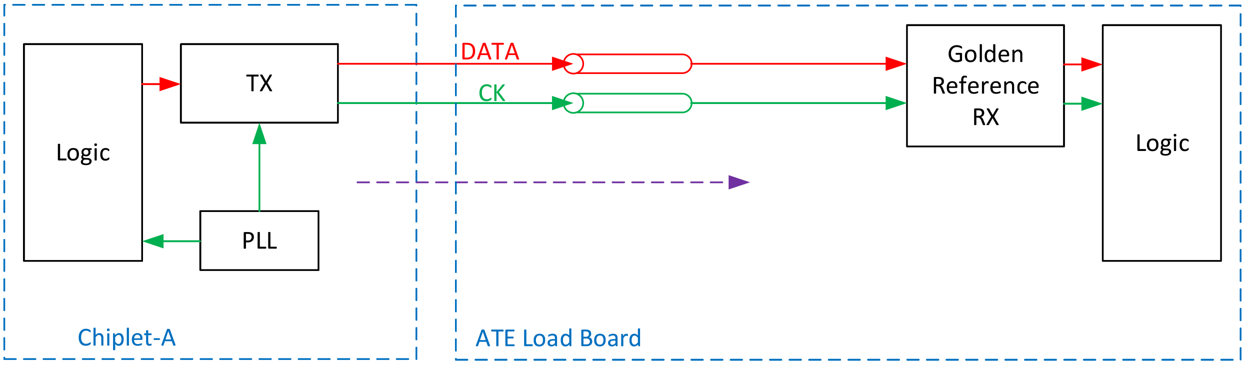

Figure 18. Open loop Tx wafer test

13.2. Loopback Test

A BoW interface may implement loopback testing for several use cases: at chiplet wafer-sort test, post-assembly package test, 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 need open-loop testing. In Tx open loop testing, shown in Figure 18, 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 19, 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.

The logic for generating and testing the PRBS sequences is outside the PHY, e.g., in the Link Layer.

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

- slice-to-slice short loopback
  - Data is looped back within the chip from a Tx slice to an Rx slice using on-chip switching (shown in Figure 20). The short loopback path is configured by the ATE using implementation-dependent registers.
  - Loopback can be implemented before the PHY serializer, between the serializer and the output buffer, and/or at the bumps.

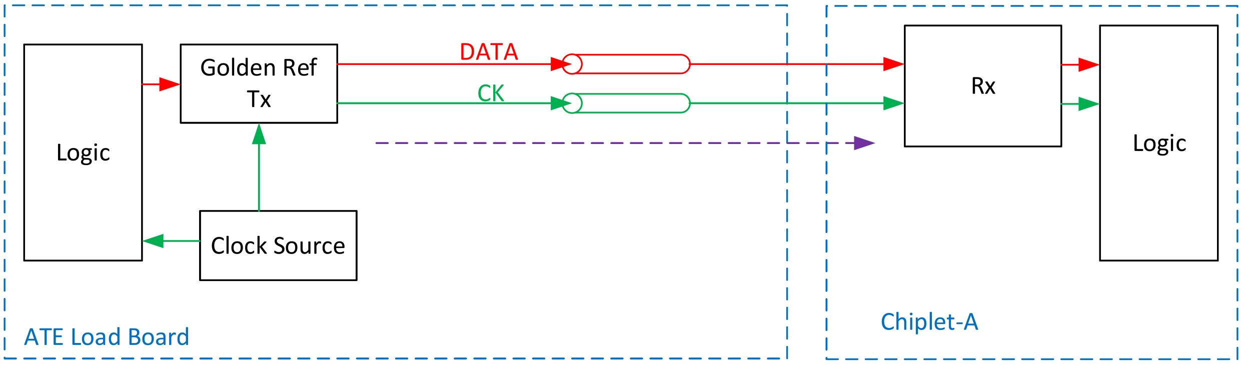

Figure 19. Open loop Rx wafer test

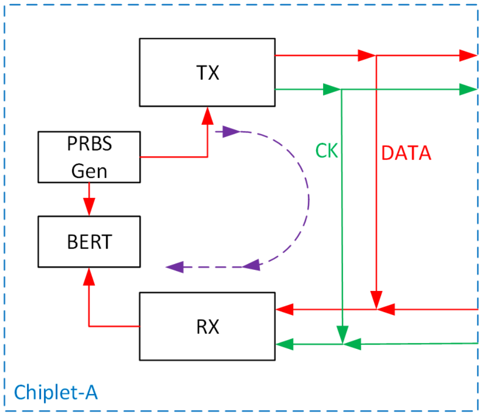

Figure 20. Short loopback testing

- intra-slice short loopback
  - A single slice containing both Rx and Tx paths sharing the same bumps can perform on-chip loopback testing simply by turning on both the Rx and Tx paths at once. This has more on-chip circuitry, but allows loopback testing with no switches or extra lines connected to the bumps other than the Tx driver tristate switches. Figure 20 applies, except there is only one shared set of bumps for a Tx/Rx slice.
- long loopback
  - 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 21). 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.

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

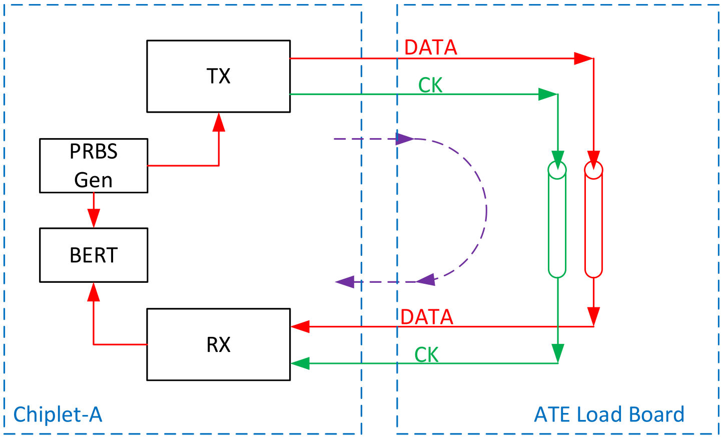

Figure 21. Long loopback testing

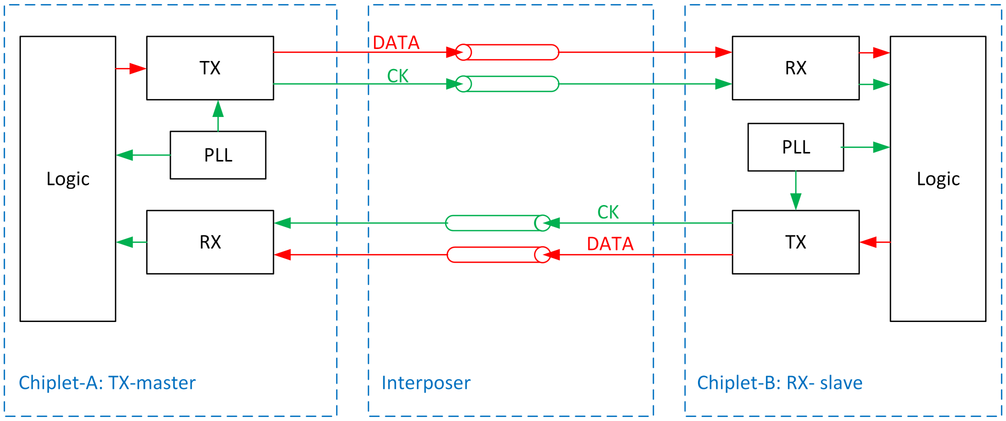

Figure 22. Chiplet-to-chiplet long loopback

purposes.

Figure 22 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.

Created with Madoko.net.