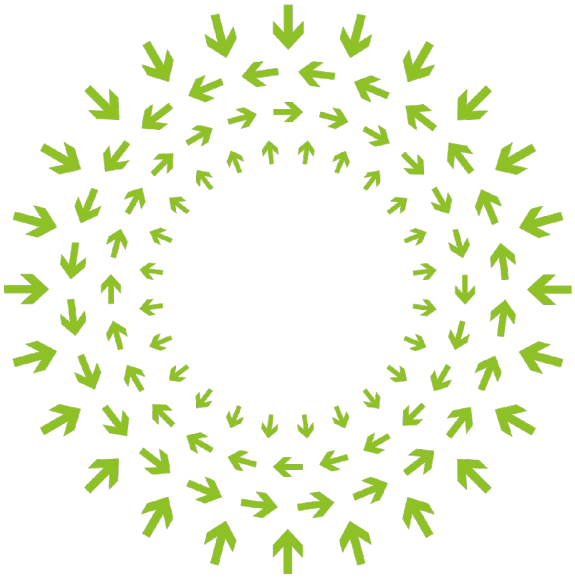
.



OPEN
Compute Project

BoW Memory Addendum to the Bunch of Wires (BoW) PHY Specification

The Open Domain-Specific Architecture BoW Workstream

Version 1.0

Author: Kevin Donnelly, Eliyan Corporation

Version History

Note: Refer to the OCP Contribution Versions, Revisions and Errata best practices documentation. Versions and revisions are made to this document and logged here. Errata is a separate document such that the contribution specification document it refers to was not revised. Ex: Version 1, Errata E1

Date	Version #	Author	Description
08/18/2025	1.0	Kevin Donnelly	1.0 Release version

Table of Contents

Version History 2
Purpose of this Addendum 5
Background 5
1. License 6
1.1. Open Web Foundation (OWF) CLA 6
2. Acknowledgements 7
3. Compliance with OCP Tenets 8
3.1. Openness 8
3.2. Efficiency 8
3.3. Scale 8
3.4. Impact 9
3.5. Sustainability 9
4. Scope 9
4.1. BoW Specification Repository 10
4.2. BoW Memory Addendum Repository 10
5. BoW Overview 11
5.1. Key Features and Conformance 11
5.2. BoW Slice 11
6. Signal Definitions 12
6.1. Directionality 12
6.2. Die-to-Die Signals (wires) 12
6.3. Slice Logic (Link Layer) Interface for a Unidirectional PHY 13
6.4. Slice Logic (Link Layer) Interface for a Bidirectional PHY 13
7. Modes and Reach 14
8. Physical Configuration 15
8.13 Use Cases for BoW D2D Links in Memory Applications 15
8.14 Example Reference Link for BoW Memory Controller Chiplets 16
8.15 Example Reference Link for BoW Memory Chiplet 17
8.16 Example BoW Memory Reference Link for Bidi Memory 18
8.17 Connecting Bidirectional Slices for BoW Memory 19

8.18 Example BoW Memory Reference Link for SPHBM 21
9. PHY Electrical Specifications 22
10. PHY Timing Specifications 23
10.1. Bit Ordering 23
10.2. Clocking 23
10.2.1 Symmetric Clocking for Memory Controller Chiplet 23
10.2.2 Asymmetric Clocking for Memory Chiplet 24
10.2.3 Clocking for Dynamic Bidirectional in SPHBM 25
10.3. Clock and Data Specifications 26
10.3.11 Clock and Data Timings for Memory Chiplets 26
10.3.12 Timing Specifications for Dynamic Bidirectional BoW PHYs 27
10.3.13 Timing for Bus Turnaround 27
11. Power Management 27
12. Chip-to-Chip Channel Specifications 28
13. Redundancy 28
14. Reset and Initialization 28
14.1. External Facilities 28
14.2. Initialization Sequence 28
14.2.1. Initialization Sequence for Memory Controller Chiplets 28
14.2.2. Initialization Sequence for Memory Chiplets 28
14.2.3. Initialization Sequence for BoW Bidirectional Memory 30
14.3. Bidirectional Slice Turnaround 30
15. Configuration 31
16. Control Register Mapping 31
17. Testability 31
18. Sideband Slices 31
19. BoW in an ODSA Design 31

Purpose of this Addendum

The purpose of this BoW Memory Addendum to Bunch of Wires (BoW) PHY Specification Version 2.0 is to provide optional enhancements that enable the BoW Die-to-Die (D2D) interface to be more efficiently used to directly connect memory devices to ASICs. This Addendum is intended to be read along with the BoW Specification.

Background

In some applications, an ASIC is connected to a memory controller chiplet that is then connected to memory, such as DDR DRAM. Because such connections are between two logic devices, the existing BoW interface can be (and has been) used to connect such interfaces. The existing BoW specification 2.0 already allows for bidirectional I/Os that are configured as either Tx or Rx at initialization, and for asymmetric traffic (for example, to support a 2:1 ratio of read-to-write data).

However, in other applications, a D2D interface like BoW can be used to connect an ASIC directly to a memory device, without an intermediate chiplet. The interfaces of memory devices, such as DDR, LPDDR, GDDR, and HBM, can either read or write data, but not both at the same time; therefore, these interfaces employ dynamic bidirectional (half-duplex) interfaces. The specification aims to optimize efficiency by preventing the waste of pins and bandwidth typically needed for simultaneous read and write operations, because memory devices can only support one at a time. While supporting this dynamic read/write data, memory interfaces also support some fixed unidirectional control signals (for Command and Address). In addition, the data interfaces of memory devices generally provide additional pins to ensure interface data integrity by enabling data encoding and error correction, or ECC. Therefore, memory interfaces generally are non-binary, providing for example, 18 pins for 16 data, or 68-72 pins for 64 data, etc.

The following is a list of features that can enable ASICs to interface directly with memory devices using a derivative of the BoW PHY described in this Addendum:

- Add support for dynamic (half-duplex) bidirectional data
- Keep some lanes as fixed unidirectional to support CMD/ADD
- Increase slice width to 18-bits (16 data + 2 ECC)
- Keep the timing and signaling specifications the same as BoW 2.0, with some additional options for clock & data alignment for memory applications

An example use case for BoW memory is to support the bandwidth of HBM4 (2TB/s read or write) along with overhead bandwidth for ECC and control signals, and fit in a beachfront of <10mm.

1. License

1.1. Open Web Foundation (OWF) CLA

Contributions to this Specification are made under the terms and conditions set forth in Modified OWF-CLA-1.0.2 (As of June 1, 2023) ("Contribution License") by:

- Eliyan Corporation

Usage of this Specification is governed by the terms and conditions set forth in **Modified OWFa1.0.2 Final Specification Agreement (FSA) (As of June 1, 2023) ("Specification License").**

You can review the applicable Specification License(s) referenced above by the contributors to this Specification on the OCP website at http://www.opencompute.org/participate/legal-documents/. For actual executed copies of either agreement, please contact OCP directly.

Notes:

The above license does not apply to the Appendix or Appendices. The information in the Appendix or Appendices is for reference only and non-normative in nature.

NOTWITHSTANDING THE FOREGOING LICENSES, THIS SPECIFICATION IS PROVIDED BY OCP "AS IS" AND OCP EXPRESSLY DISCLAIMS ANY WARRANTIES (EXPRESS, IMPLIED, OR OTHERWISE), INCLUDING IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, FITNESS FOR A PARTICULAR PURPOSE, OR TITLE, RELATED TO THE SPECIFICATION. NOTICE IS HEREBY GIVEN, THAT OTHER RIGHTS NOT GRANTED AS SET FORTH ABOVE, INCLUDING WITHOUT LIMITATION, RIGHTS OF THIRD PARTIES WHO DID NOT EXECUTE THE ABOVE LICENSES, MAY BE IMPLICATED BY THE IMPLEMENTATION OF OR COMPLIANCE WITH THIS SPECIFICATION. OCP IS NOT RESPONSIBLE FOR IDENTIFYING RIGHTS FOR WHICH A LICENSE MAY BE REQUIRED IN ORDER TO IMPLEMENT THIS SPECIFICATION. THE ENTIRE RISK AS TO IMPLEMENTING OR OTHERWISE USING THE SPECIFICATION IS ASSUMED BY YOU. IN NO EVENT WILL OCP BE LIABLE TO YOU FOR ANY MONETARY DAMAGES WITH RESPECT TO ANY CLAIMS RELATED TO, OR ARISING OUT OF YOUR USE OF THIS SPECIFICATION, INCLUDING BUT NOT LIMITED TO ANY LIABILITY FOR LOST PROFITS OR ANY CONSEQUENTIAL, INCIDENTAL, INDIRECT, SPECIAL OR PUNITIVE DAMAGES OF ANY CHARACTER FROM ANY CAUSES OF ACTION OF ANY KIND WITH RESPECT TO THIS SPECIFICATION, WHETHER BASED ON

BREACH OF CONTRACT, TORT (INCLUDING NEGLIGENCE), OR OTHERWISE, AND EVEN IF OCP HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

2. Acknowledgements

The Contributors of this Specification would like to acknowledge the following for their feedback:

Individual	**Company**
Elad Alan	Blue Cheetah Analog
Anu Ramamurthy	Microchip
Kash Johal	YorChip
Morgan Whately	Infineon
Peicheng Ju	Eliyan
Boon Chong Ang	self
Marek Hempel	Analog Devices
James Wong	Palo Alto Electron
Shahab Ardalan	Enosemi

3. Compliance with OCP Tenets

The BoW Memory Addendum to the BoW Specification defines a versatile, open and interoperable physical interface between two chiplets or chip-scale-packages (CSP) in a common package, and is fully backwards compatible with the BoW specification. This document specifies the BoW interface PHY layer, and defines a set of die-to-die parallel interfaces that implementers / adopters have the flexibility to trade off throughput/chip-edge for design complexity, cost, and packaging technology. The use of BoW memory is expected to be confined to connect die (one of which may be a memory device) 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 OCP tenets as follows:

3.1. Openness

- Available for download without restrictions or cost
- Unencumbered by technology license costs

3.2. Efficiency

- Very low peak power (< 0.5 - 1 pJ/bit for doubly terminated modes, <0.25 - 0.5 pJ/bit for unterminated)
- Energy proportional down to ~10% activity
- Very low latency (<2-4 ns without FEC, <15 ns with FEC from logic interface to logic interface)
- High throughput density (up to 2 - 12+ Tbps/mm-chip-edge)
- Support for new features for memory including dynamic bidirectional signaling (half-duplex)
- Support for key BoW features such as redundancy, sideband signaling, and configurable directionality
- Fully backwards compatible with BoW 2.0

3.3. Scale

- Flexible to support both laminate and advancing packaging technologies
- Portable across multiple bump pitches
- Portable across IC process nodes ranging from 65 nm to 2 nm and beyond

3.4. Impact

The BoW specification provides several key advantages for chiplet-based systems:

- Can be implemented in legacy as well as state-of-the-art process nodes
- Can be implemented in any packaging technology while retaining a single logical definition
- Provides design/implementation flexibility while retaining interoperability, allowing chiplet interfaces to be optimized for a given product/application

Compared to SerDes, BoW uses a lower data rate/wire with more wires. However, 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.

3.5. Sustainability

- The very low peak power (< 0.5 - 1 pJ/bit for doubly terminated modes) reduces power consumption of systems and minimizes environmental harm

4. Scope

The scope of this BoW Memory Addendum to the BoW Specification has several levels.

1. The BoW Memory Addendum includes these requirements:
    a. Use of existing BoW signals to support ECC
    b. Timing and electrical specifications for a memory interface
    c. Signals at the logic (Link Layer) interface

2. The BoW Memory Addendum includes these recommendations:
    a. Reference bump patterns for example interfaces
    b. Example arrangement of multiple slices to form memory links
    c. Use of dynamic bidirectional data (half-duplex on each wire)
    d. Arrangement of wires in laminate packaging
    e. Signal integrity of the wire channel

3. The following are outside the scope of this document:
    a. Simultaneous bidirectional data (full duplex on each wire)
    b. Specific implementations of the interface
    c. Integration of the interface with system-level data flow
    d. The use of this interface outside of a package or entirely inside a chip
    e. Definition of protocols for logical data transfer

4.1. BoW Specification Repository

The BoW Specification Revision 2.0 is available on the OCP Contribution website: https://www.opencompute.org/chiplets/26/bunch-of-wires-bow-phy-specification-20

4.2. BoW Memory Addendum Repository

This BoW Memory Addendum to the BoW Specification will be available on the OCP Contribution website.

5. BoW Overview

Section 5 of the BoW Specification provides an overview of BoW Physical Interface (PHY) and its use in multi-chiplet design. This Addendum describes additions and changes to support a BoW memory interface.

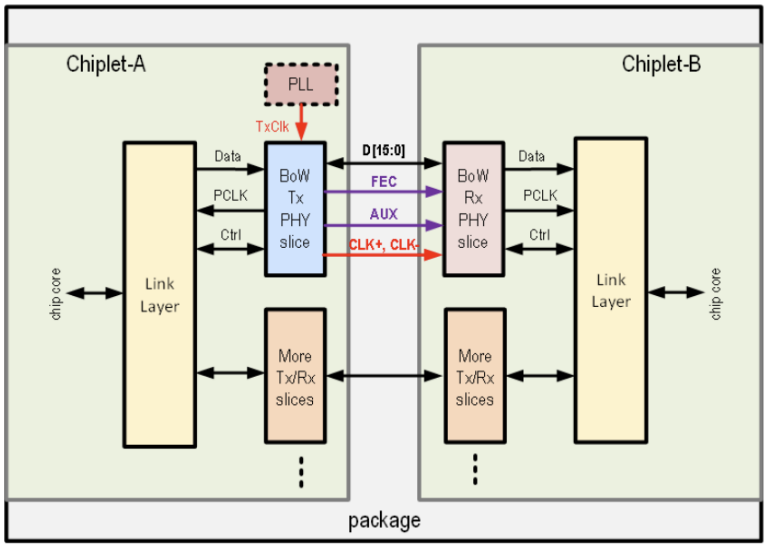
5.1. Key Features and Conformance

There are no changes to section 5.1 BoW Specification 2.0.

5.2. BoW Slice

A BoW is an energy-efficient, easy-to-use PHY interface between a pair of die inside a single package as shown in the figure below. The BoW PHY is defined as a single unidirectional slice. Multiple slices are combined to create links of the desired throughput. A link can be symmetric, asymmetric or unidirectional. The BoW PHYs between two dies 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 Specification 2.0 specifies the protocol for a BoW PHY slice. This Addendum outlines the physical layer changes for BoW PHYs applied to memory interfaces, shows examples of BoW memory PHY building blocks, and shows an example aggregation of multiple BoW PHYs into complete memory interfaces.



Addendum Figure 1: BoW Overview
(A copy of Figure 1 from the BoW Specification 2.0)

As shown in **Addendum Figure 1**, a BoW PHY slice either transmits or receives 16 bits of data between dies. 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 signals designated FEC (for Forward Error Correction) and AUX, for other optional functions such as Data Bus Inversion (DBI). For BoW memory PHYs, these two additional signals (FEC and AUX) can be used for ECC data to make an 18-bit interface (16 data + 2 ECC) that is typically used for memory applications. The data for the ECC signals is generated by memory controller logic outside the PHY, stored in the memory when data is written, and checked by the memory controller when data is later read.

There are no changes to the rest of section 5 of the BoW PHY Specification 2.0.

6. Signal Definitions

Section 6 of the BoW Specification specifies the control data signals into and out of device logic and package for BoW RX and TX slices.

6.1. Directionality

- A BoW **link** may be bidirectional (each side of a lane contains both RX and TX slices) or unidirectional (each chiplet's side contains only TX or RX slices)
- A BoW PHY **slice** is said to be bidirectional if it can be configured to operate in TX or in RX
- For BoW memory PHYs, each BoW slice is unidirectional at any given time when in operation, but can be operated as either RX or TX in a time-multiplexed (half-duplex) bidirectional manner
- This spec does not include full-duplex operation (simultaneous TX and RX over the same wires)

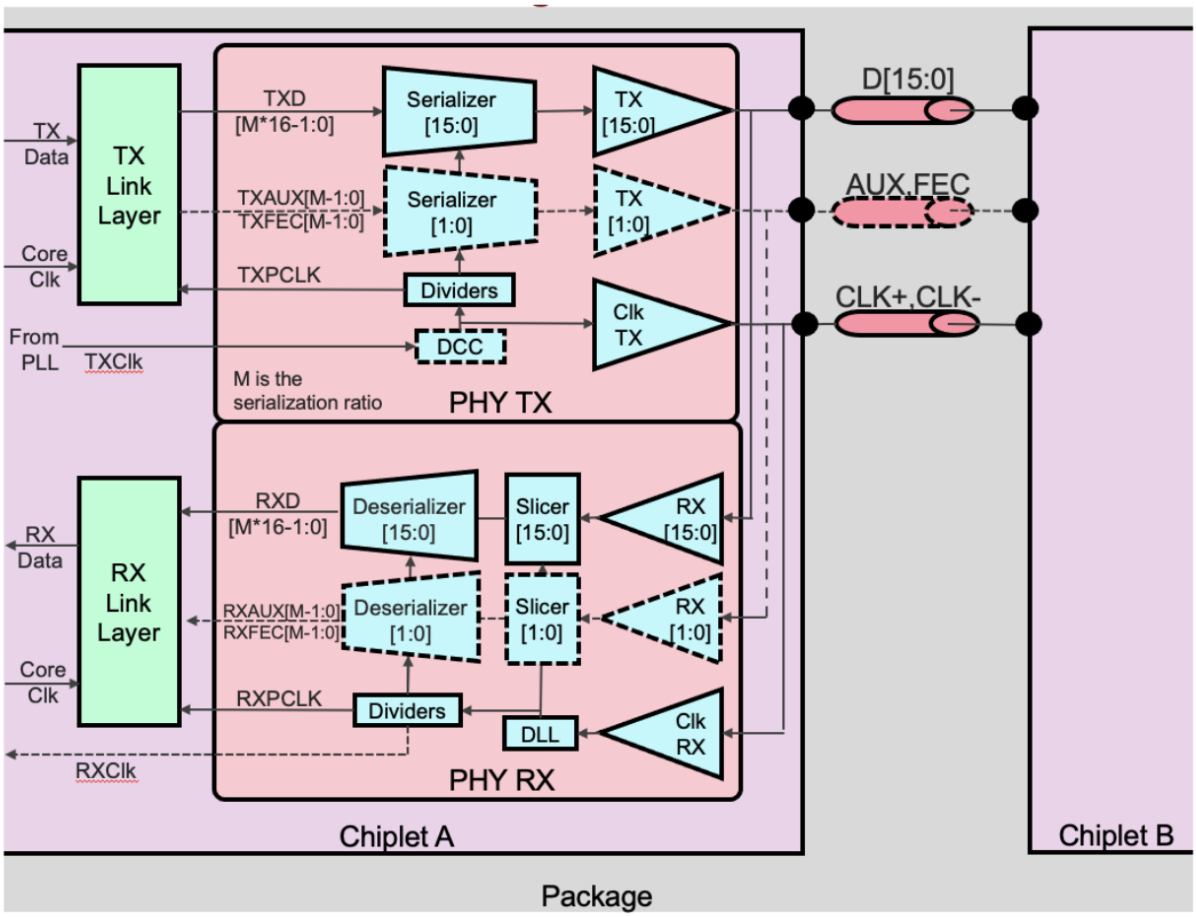
6.2. Die-to-Die Signals (wires)

As shown in **Addendum Figure 2**, each BoW slice consists of a differential clock pair, 16 single-ended data wires, and an optional pair of wires FEC and AUX. When BoW is applied to memory interfaces as BoW memory PHYs, the FEC and AUX wires can be operated as ECC data running in conjunction with the 16 data wires to provide 18 total lanes (16 data + 2 ECC).

6.3. Slice Logic (Link Layer) Interface for a Unidirectional PHY

There are no changes to section 6.3 of the BoW Specification 2.0. To clarify, if the FEC and AUX wires are operated as ECC data, they appear as extra data inputs at the parallel interface to the Link Layer.

6.4. Slice Logic (Link Layer) Interface for a Bidirectional PHY



Addendum Figure 2: BoW Bidirectional Slice Block Diagram
(A copy of Figure 4 from the BoW Specification 2.0)

As shown in **Addendum Figure 2**, a bidirectional slice for BoW memory shall have external bidirectional signal wires and unidirectional clock wires, along with two sets of internal unidirectional signals connecting to the chiplet's core logic (or Link Layer).

In mission mode for a BoW memory PHY, each slice can have both the PHY TX or the PHY RX enabled and operated in a time-multiplexed (half-duplex) manner to support read or write data. For loopback test in both BoW and BoW memory PHYs, both TX and RX can be enabled. For loopback, it is recommended to enable testing one slice at a

time to avoid drawing both RX and TX power of all the slices in the whole link at the same time.

In the BoW memory PHY, the FEC and AUX signals can be utilized as ECC data signals, operated in conjunction with the 16 data signals to provide 18 lanes (16 data + 2 ECC) of traffic between the ASIC (Chiplet A) and memory device (Chiplet B). These 2 ECC lanes are operated in a time-multiplexed (half-duplex) manner in conjunction with the 16 data lanes.

The data at the internal slice logic interface must be SDR (Single Data Rate - bit rate equal to the TXPCLK/RXPCLK frequency).

7. Modes and Reach

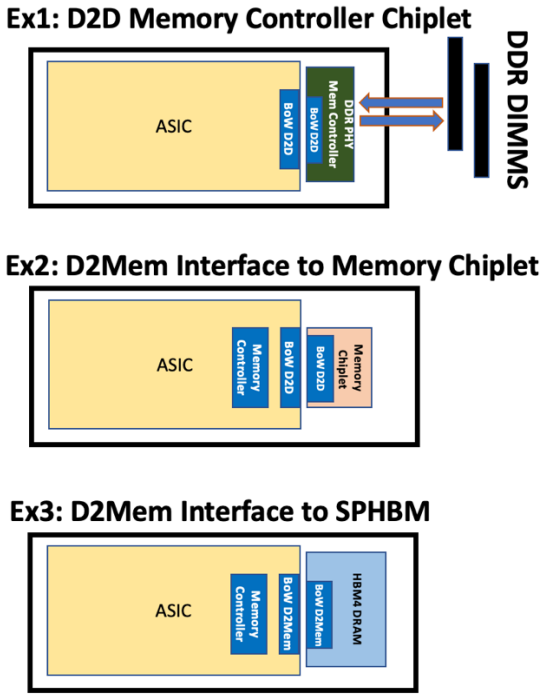
There are no changes to section 7 of the BoW Specification 2.0.

8. Physical Configuration

Section 8 of the BoW Specification 2.0 describes physical configurations of the BoW PHY. There are no changes to the existing sections 8.1 to 8.12 of the BoW Specification. The added subsections below describe physical configurations for the BoW PHY when applied to memory interfaces.

8.13 Use Cases for BoW D2D Links in Memory Applications

Three uses cases, outlined in **Addendum Figure 3**, are examples of BoW D2D PHYs applied to memory applications.



Addendum Figure 3: BoW Memory Example Use Cases

Ex1 shows BoW applied to a memory controller chiplet. In this use case, BoW is used as a logic-to-logic D2D interface from an ASIC to a chiplet containing a memory controller, and that chiplet in turn interfaces with memory (in this example, two DDR5 DIMMs). As a logic-to-logic interface, no changes to the BoW interface are required in this Addendum vs. the original BoW Specification, but this Addendum does include some descriptive examples relevant to this use case.

Ex2 shows BoW applied to a memory chiplet. In this use case, a memory chiplet is directly attached to an ASIC with a BoW memory PHY. As an example, the read/write

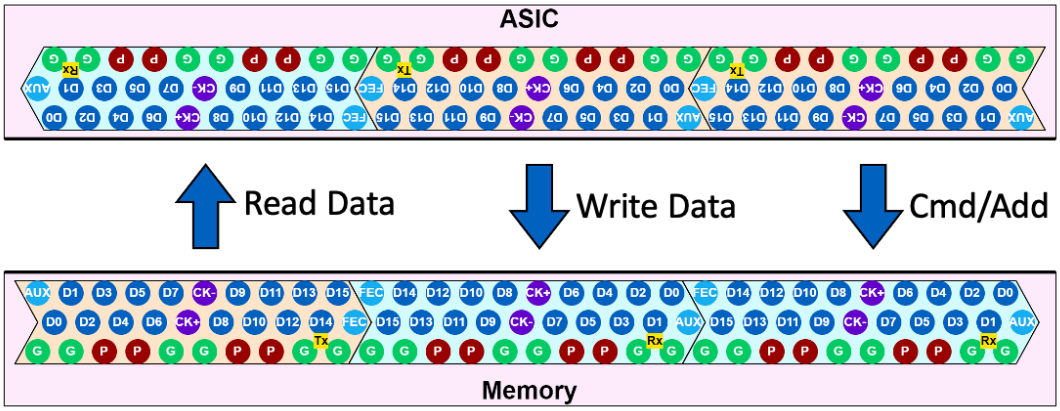
bandwidth of ~32GB/s is required in a beachfront of < 2.5mm. This use case will drive some spec changes outlined in this Addendum.

Ex3 shows BoW applied to a Standard Package HBM Chiplet (SPHBM). In this use case, an ASIC is attached to a high-bandwidth memory (HBM) chiplet on standard organic packaging using a BoW memory PHY. As an example, the read/write bandwidth of ~2TB/s is required in a beachfront of < 10mm. This use case will drive spec changes outlined in this Addendum.

8.14 Example Reference Link for BoW Memory Controller Chiplets

A memory controller chiplet example reference link is described below. This reference link is similar to Figure 6 from section 8.3 of the BoW Specification 2.0.

Memory interfaces must provide read and write bandwidth along with control information, such as commands and addresses (Cmd/Add). To provide equal read and write bandwidth to a memory, one application of BoW PHYs includes one slice dedicated for read traffic and another slice dedicated to write traffic, together with a third slice for Cmd/Add traffic. **Addendum Figure 4** shows a three-slice reference link for memory applications that provides equal read and write bandwidth with additional bandwidth for Cmd/Add.



Addendum Figure 4: Three-Slice Reference Link for BoW Memory Controller Chiplet

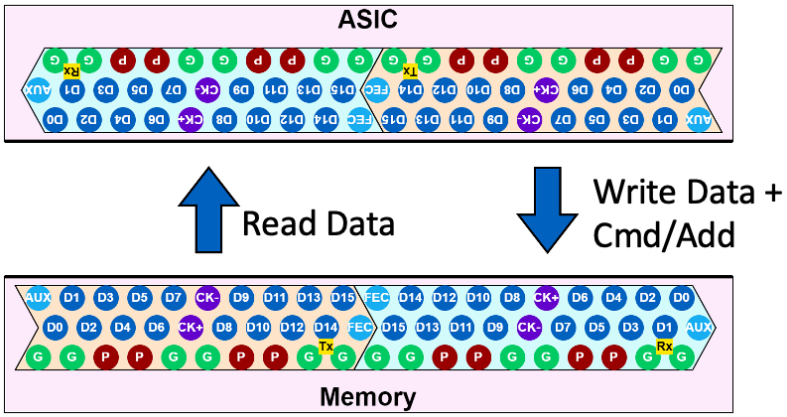
The main disadvantage of this three-slice approach is the additional beachfront and silicon area required for the extra slice. In most memory applications, the Cmd/Add bus runs at a significantly lower bandwidth than the data buses, meaning this dedicated slice is significantly underutilized. If the Cmd/Add bandwidth were <50% of read/write bandwidth, one potential way to mitigate this third-slice penalty would be to use a half-slice for the third slice dedicated to Cmd/Add information. Note that half-slice components are described in section 8.4 of the BoW Specification 2.0.

This asymmetric ASIC PHY (2 TX and 1 RX) is already allowed under the existing BoW PHY Specification. In alternate applications, users can allocate multiple read data slices for every write data slice, which is also permitted under the existing specification. It is also possible to multiplex between write data and Cmd/Add traffic in a single slice to further reduce PHY beachfront, as described below. In all cases, the FEC and AUX wires can be operated as ECC data running in conjunction with the 16 data wires, as described in section 6.2 of this Addendum.

8.15 Example Reference Link for BoW Memory Chiplet

A memory chiplet reference link is described below. This reference link is similar to Figure 6 from section 8.3 of the BoW Specification 2.0.

The reference link in **Addendum Figure 5** contains two slices: a first slice for read data, and a second slice that is multiplexed between write data and Cmd/Add traffic. This uses less beachfront and silicon area than the three-slice approach, with the tradeoff that the bandwidth available for write traffic will be less than the available read bandwidth. In one example, the write bandwidth is 50% of the read bandwidth with the remaining 50% for Cmd/Add traffic. Most memory applications result in far more read than write transactions, and a 2:1 ratio of read to write traffic is common. So, this reference link for a BoW memory chiplet PHY supports a wide range of memory chiplet applications.

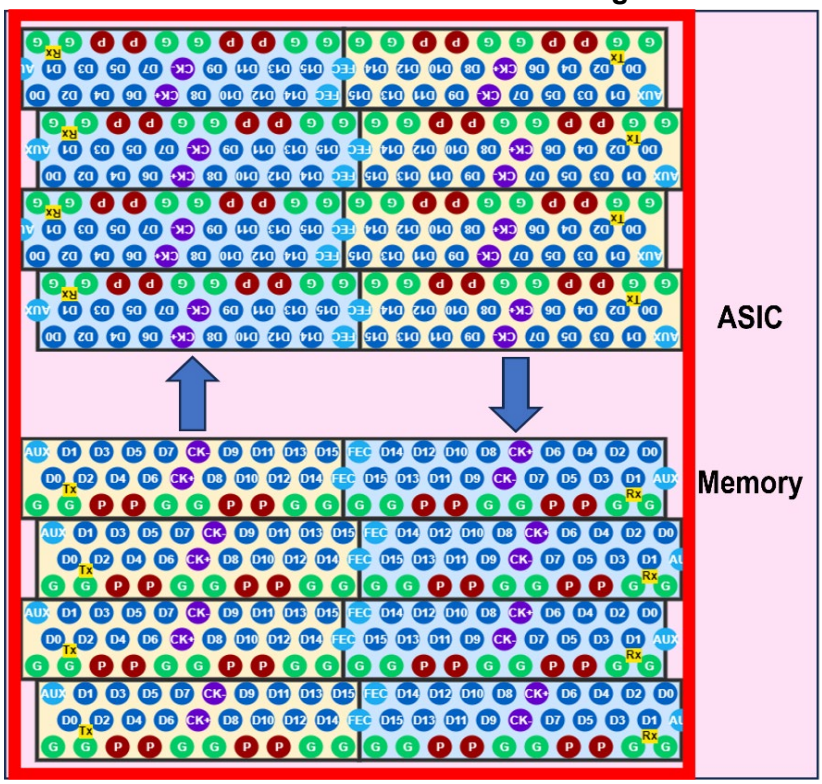


Addendum Figure 5: Two-Slice Reference Link for BoW Memory Chiplet

With the reference link above, in many memory products the FEC and AUX wires can be operated as ECC data running in conjunction with the 16 data wires, as described in section 6.2 of this Addendum.

8.16 Example BoW Memory Reference Link for Bidi Memory

To be applicable to higher bandwidth memories, BoW slices must be stacked to deliver the maximum bandwidth in a given beachfront (BW/mm). An example reference link for a standard 4-slice BoW interface is shown in **Addendum Figure 6**.

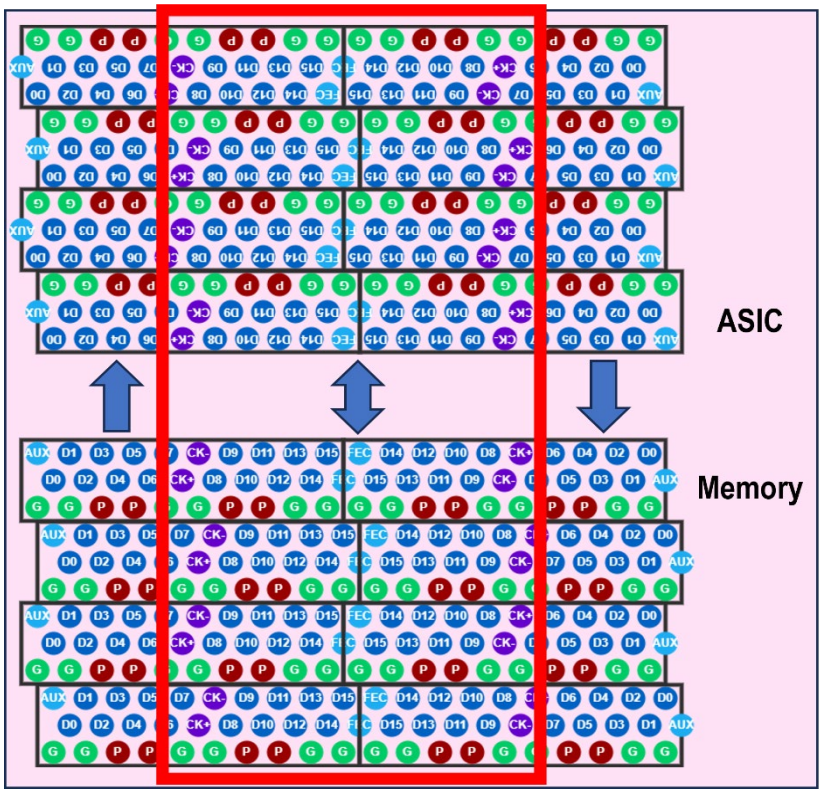


Addendum Figure 6: Reference Link for 4-Slice BoW Stack

In this standard 4-slice stacked reference link, half of the ASIC slices are configured as transmit mode and the other half in receive mode, as shown in **Addendum Figure 6**. Since each slice has 16 DQ lanes, the 4-slice stack has 64 total data lanes (+ ECC), and at 32Gbps this would deliver 1Tbps per slice. If each slice is 1mm wide, this configuration delivers 1Tbps of read bandwidth and 1Tbps of write bandwidth over 2mm of beachfront. However, DRAM devices generally cannot read and write simultaneously, as the DRAM core either accepts write data or provides read data at any given time. So, in actual memory applications, this reference link supports 1Tbps of read or write bandwidth at any given time in the 2mm of beachfront.

The BoW PHY Specification 2.0 supports bidirectional interfaces, although it specifies that the interfaces are configured in operation as unidirectional. This Addendum extends the existing interface circuitry to support half-duplex bidirectional data, where for example a BoW slice on an ASIC could either transmit (write data) or receive (read data) at any given time.

The same example reference link for a 4-slice BoW interface can be configured instead to support dynamic bidirectional data as shown in **Addendum Figure 7**.



Addendum Figure 7: Reference Link for 4-Slice Stack with Bidirectional Data

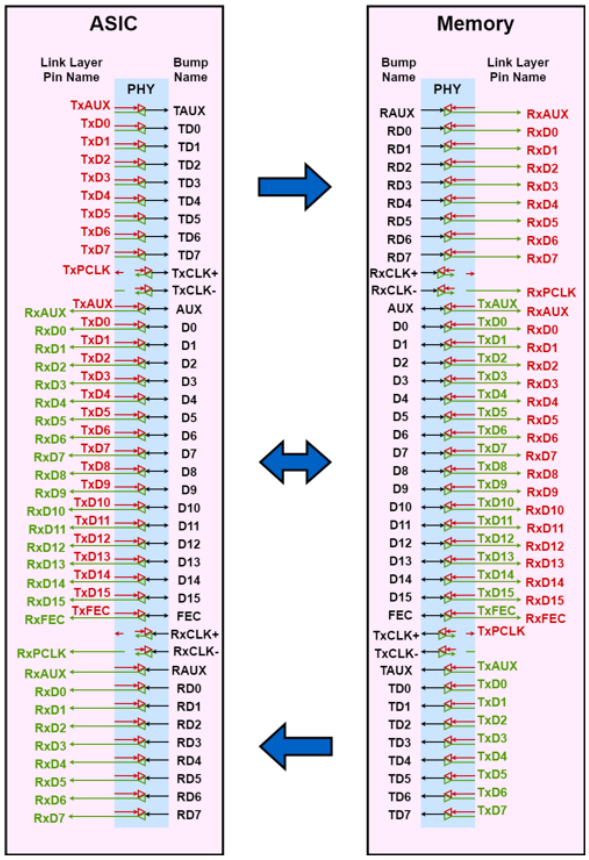
With the same above assumptions on data rate and number of lanes, the middle section of this reference link provides the same 1Tbps of read or write bandwidth in only 1mm of beachfront, doubling the effective beachfront bandwidth density (BW/mm) for memory applications. In this example reference link, the PHY can simultaneously provide unidirectional lanes to support traffic such as Cmd/Add (from ASIC to memory), and status and error information (from memory to ASIC). Note that the unidirectional lanes shown in **Addendum Figure 7** include the same number in each direction, but asymmetric numbers of lanes are allowed to support memory applications with much more outbound traffic (Cmd/Add) from ASIC to memory than return traffic.

8.17 Connecting Bidirectional Slices for BoW Memory

A bidirectional BoW memory PHY is designed to alternate in time between RX mode and TX mode, operating in either RX or TX mode at any given time. The corresponding parallel interfaces at the Link Layer must therefore support both Tx data and Rx data, although not at the same time.

Addendum Figure 8 shows how slices are connected between ASIC and memory chiplets (similar to Figure 13 of the BoW Specification 2.0). The middle section is dynamically switched between red (ASIC TX to Memory RX) and green (Memory TX to ASIC RX) depending on whether the operation is write data or read data (respectively). In contrast, the top section is always in red (ASIC TX to Memory RX) to send Cmd/Add data from the ASIC to the Memory, while the bottom section is always green (Memory TX to ASIC RX) to send status and error information back from the Memory to the ASIC.

- The Link Layer interface for a bidirectional BoW slice has a complete set of Tx wires and a complete set of Rx wires
- ASIC > Memory (red)
  - Top section configured
  - Middle section dynamic
- Memory > ASIC (green)
  - Middle section dynamic
  - Bottom section configured
- This diagram does not show the large number of wires at the link layer due to the serialization factor

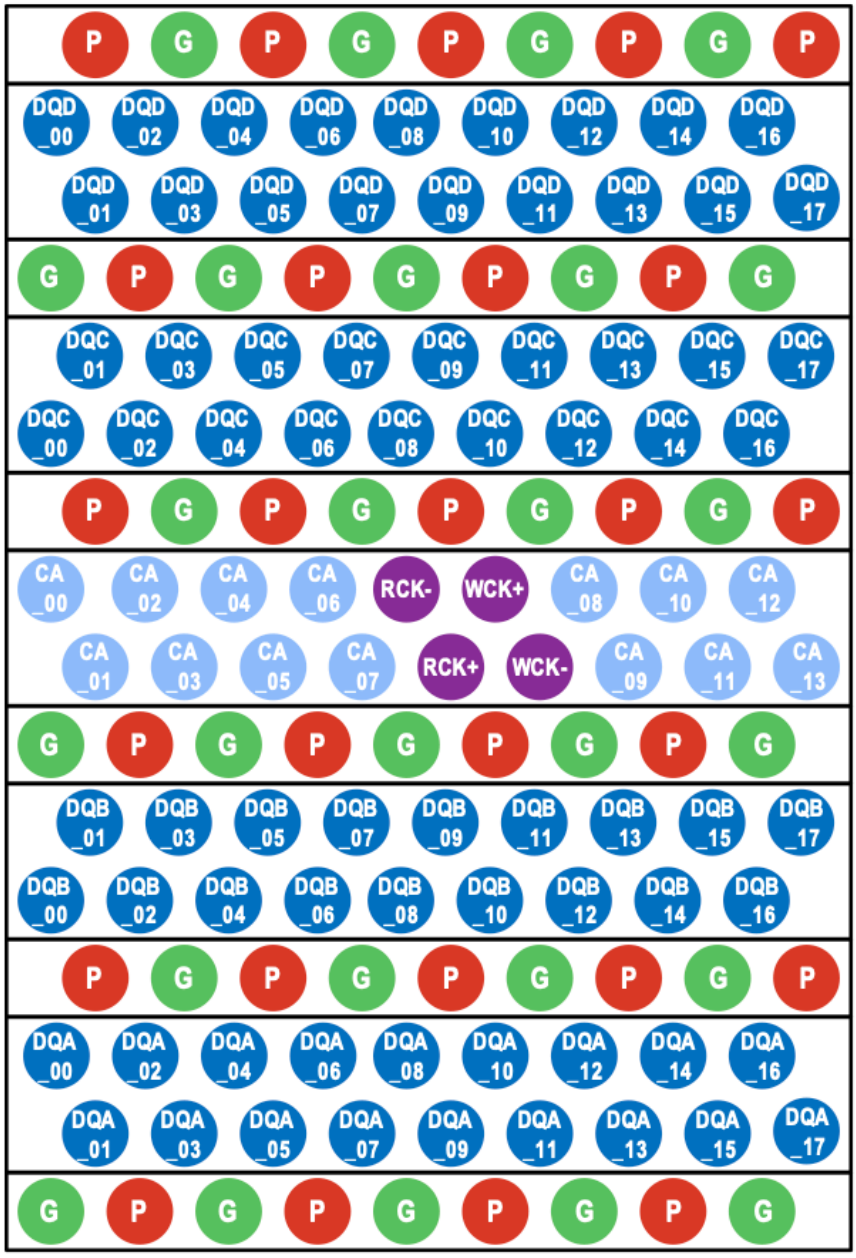


Addendum Figure 8: Pattern of Slices in a Dynamic Bidirectional Link

This figure is a reference dynamic bidirectional link, and implementations will vary. The number of dynamic bidirectional data lanes is usually greater than the number of unidirectional lanes, and the number of Cmd/Add lanes is usually greater than the number of return lanes for status and error information. For the specific bidirectional reference link for BoW applied as a Standard Package HBM interface (outlined in Section 8.18 below), there are 11 unidirectional Cmd/Add lanes from ASIC to Memory, three unidirectional Error lanes back from Memory to the ASIC, and 72 dynamic bidirectional DQ lanes between the ASIC and Memory.

8.18 Example BoW Memory Reference Link for SPHBM

A Reference Link (similar to Figure 6 of the BoW Specification 2.0) is shown below for the third use case outlined in section 8.13 above, the use case of a BoW memory PHY applied to a Standard Package HBM (SPHBM) interface.



Addendum Figure 9: Reference Link for BoW applied to Standard Package HBM

The SPHBM Reference Link in **Addendum Figure 9** contains a stack of five slices. Moving up from the bottom (chip edge), the first slice contains a set of 18 dynamic

bidirectional data lanes (DQA), the second slice contains another set of 18 dynamic bidirectional lanes (DQB), the middle slice contains 14 unidirectional command and address lanes (CA) plus two clock pairs (RCK and WCK), the fourth slice contains another set of 18 dynamic bidirectional lanes (DQC), and the fifth slices contain a last set of 18 dynamic bidirectional data lanes (DQD). Together, these DQ and CA lanes provide the control and data bandwidth necessary to support four HBM channels, referred to as a quad-channel SPHBM PHY block. A full HBM4-class memory interface with 32 total HBM channels would therefore be built using eight of these quad-channel SPHBM PHY blocks.

9. PHY Electrical Specifications

There are no changes to section 9 of the BoW Specification 2.0.

10. PHY Timing Specifications

Section 10 of the BoW 2.0 Specification outlines the BoW PHY timing specifications.

10.1. Bit Ordering

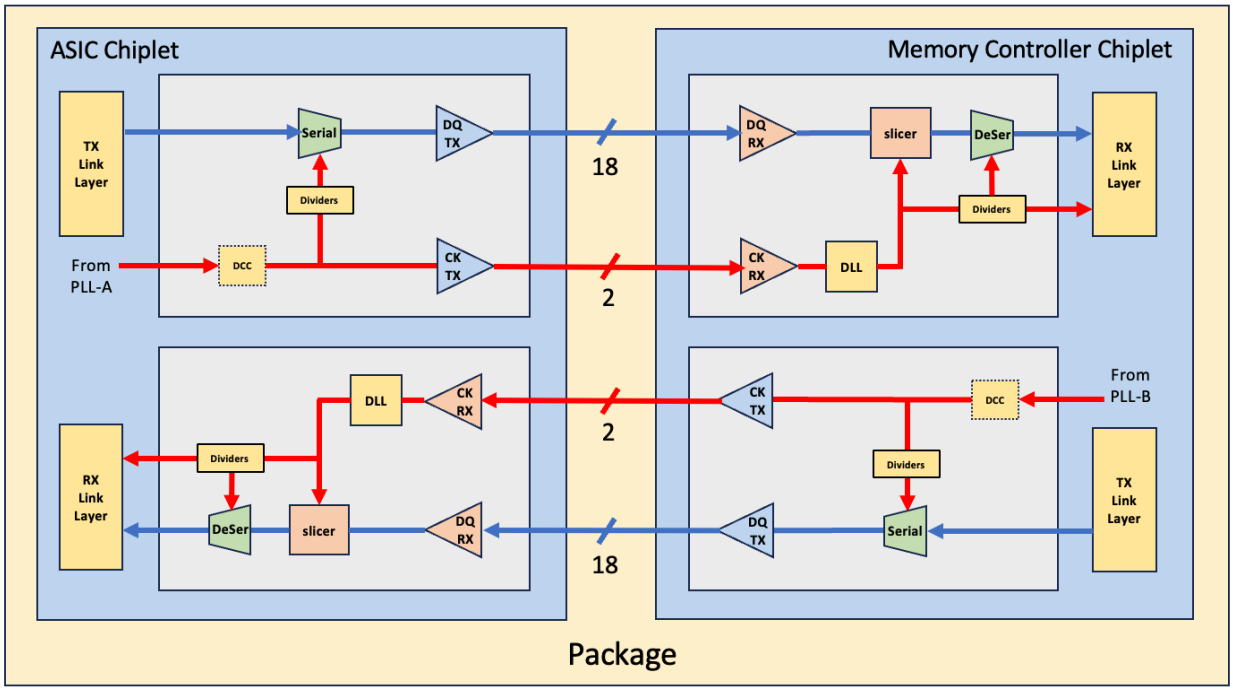
There are no changes to section 10.1 of the BoW Specification 2.0.

10.2. Clocking

Section 10.2 of the BoW Specification 2.0.describes clocking for the BoW PHY. These added subsections describe clocking for the BoW memory use cases outlined above.

10.2.1 Symmetric Clocking for Memory Controller Chiplet

For the use case of BoW D2D applied to a memory controller chiplet, the connection from the ASIC is a standard logic-to-logic D2D interfaces, and therefore implements clocking as described in Figure 16 of the BoW Specification 2.0. When applied to a link with both write and read data, as shown in **Addendum Figure 10**, the clocking is symmetric with separate PLLs on each TX side and separate DLLs (or delay lines) to adjust timing on each RX side. The TxClk is transmitted aligned with TX data, and the RX side must align the slicer clock to sample the data correctly. This block diagram is symmetric.

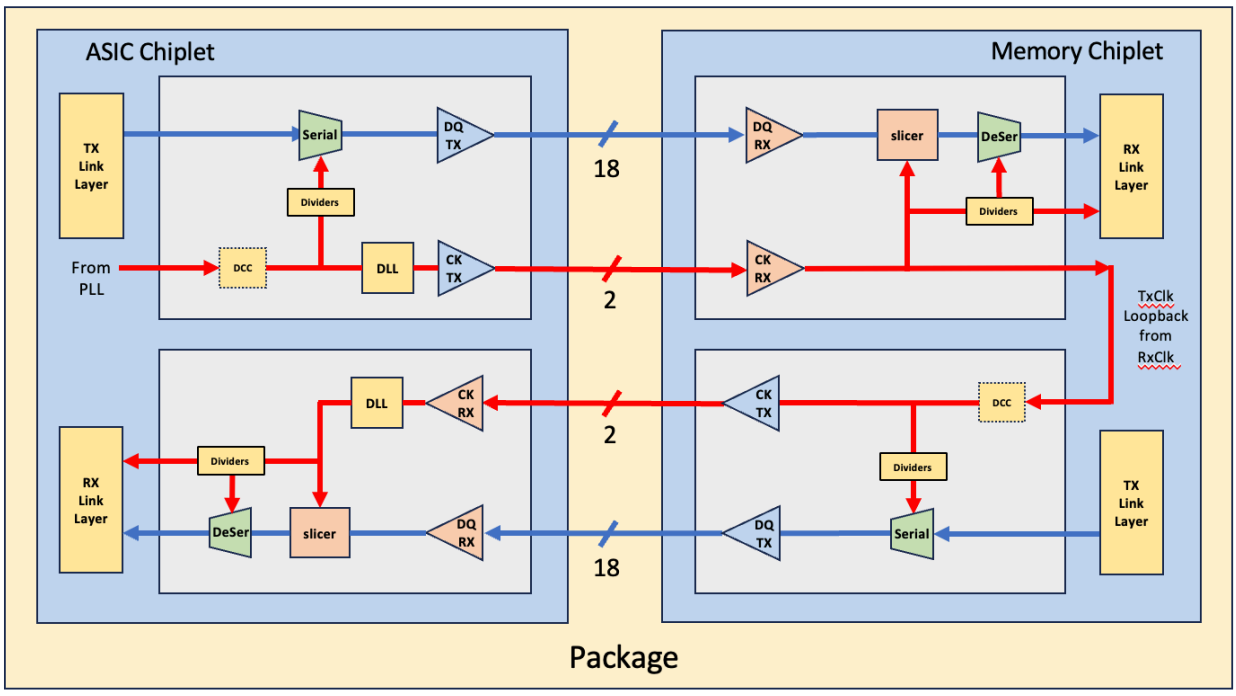


Addendum Figure 10: Symmetric Clocking for Memory Controller Chiplet Use Case

In this figure, the top Tx slice of the ASIC chiplet has an internal PLL clock that is passed through an optional DCC (duty cycle corrector) to clock transmitter. The differential clock and 18 DQ bits (16 data + 2 ECC) are transmitted to the top slice of the memory controller chiplet, with a DLL that provides 90-degree phase shift for the clock path vs. data to capture the received data. The reverse connection is symmetric, with an internal PLL clock on the memory controller chiplet that is passed through an optional DCC (duty cycle corrector) to clock transmitter. The differential clock and 18 DQ bits (16 data + 2 ECC) are transmitted to the bottom slice of the ASIC chiplet which has an internal DLL (or delay line) to adjust the received clock phase and capture the received data.

10.2.2 Asymmetric Clocking for Memory Chiplet

For the use case of BoW D2D applied to a memory chiplet, the connection from the ASIC requires a logic-to-memory D2D interface. The process technology used for memory devices is generally not as advanced as the process technology used for ASICs, and therefore, it is not desirable to implement complex logic and analog functions like PLLs in memories. As a result, memory interfaces are typically defined to be asymmetric, with the complex timing and link training functions implemented on the ASIC PHY side, which simplifies the memory PHY. When applying BoW to a memory link with both write and read data, the clocking, shown in **Addendum Figure 11** below, is utilized with both the PLL and DLL (or delay line) on the ASIC side and with no timing control on the memory side. This block diagram is therefore asymmetric.

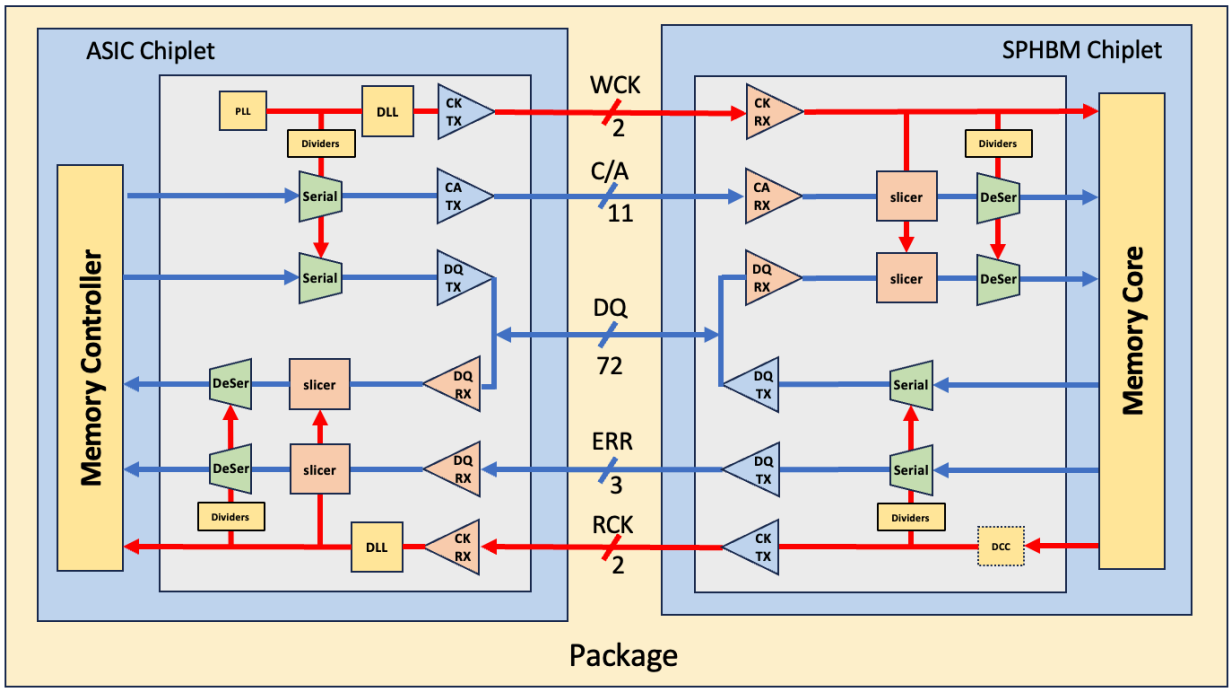


Addendum Figure 11: Asymmetric Clocking Block Diagram for Memory Chiplet Use Case

In **Addendum Figure 11**, the top Tx slice of the ASIC chiplet has an internal PLL clock that is passed through an optional DCC (duty cycle corrector) to a DLL that provides 90-degree phase shift for the clock path vs data. The differential clock and 18 DQ bits (16 data + 2 ECC) are transmitted to the top slice of the memory chiplet, which needs no timing adjustment circuitry to capture the received data. The received clock is internally looped back through an optional DCC to be the transmitted clock in the bottom slice of the memory chiplet. The differential clock and 18 DQ bits (16 data + 2 ECC) are transmitted to the bottom slice of the ASIC chiplet which has an internal DLL (or delay line) to adjust the received clock phase and capture the received data.

10.2.3 Clocking for Dynamic Bidirectional in SPHBM

For the specific application of dynamic bidirectional BoW to the use case of Standard Package HBM (SPHBM), a clocking block diagram is shown in **Addendum Figure 12**.



Addendum Figure 12: Clocking Block Diagram for SPHBM Use Case

For SPHBM, there are two clock references between the chiplets, Write Clock (WCK) from ASIC to memory, and Read Clock (RCK) from memory to ASIC. The timing reference for the Cmd/Add lanes (C/A) is WCK, while the timing reference for the returning error and status lanes (ERR) is RCK. For the dynamic bidirectional DQ lanes, the timing reference for data in write mode (ASIC to memory) is WCK, and the timing reference for data in read mode (memory to ASIC) is RCK.

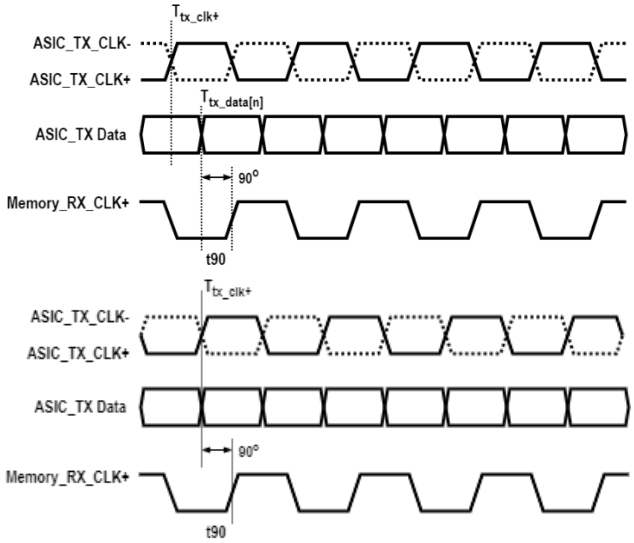
10.3. Clock and Data Specifications

Section 10.3 of the BoW Specification 2.0 describes the extensive clock and data timing specifications of the BoW PHY. There are no changes to the existing subsections 10.3.1 to 10.3.10 of the BoW Specification 2.0. Also, no timing specification changes are required for the memory controller chiplet use case.

The following new clock and data timing subsections address the other two use cases (memory chiplet and HBM memory) that are described in section 8.13 above.

10.3.11 Clock and Data Timings for Memory Chiplets

For the memory chiplet use case, the clocking is asymmetric as described in section 10.2.2 above, and there are different data and clock alignment for the ASIC chiplet vs. the memory chiplet, with all of the timing control complexity being on ASIC side.



Addendum Figure 13: Asymmetric Clock and Data Timings for Memory Chiplet Use Case

Addendum Figure 13 shows the clock and data times for the memory chiplet use case. When the ASIC chiplet transmits data to memory, the forwarded TxClk is delayed by 90-degrees with respect to the data so the memory can use this clock directly to sample the data (without timing circuitry required on the memory). However, there is no timing shift on the memory chiplet when the memory transmits data back to the ASIC chiplet, so the TxClk and data are aligned and the ASIC chiplet must delay its internal clock by 90-degrees to properly sample the incoming data.

10.3.12 Timing Specifications for Dynamic Bidirectional BoW PHYs

There are two clock and data timing options for dynamic bidirectional BoW memory PHYs:

1. Symmetric clocking where both chiplets have integrated PLLs.
2. Asymmetric clocking with clocks looped back on the memory side.

For either type of bidirectional PHY, the BoW Specification 2.0 TX timing specs are met during TX operations, and the RX timing specs are met during RX operations. So, there are no new timing specifications driven by moving from configurable bidirectional lanes (in the BoW Specification 2.0) to dynamic bidirectional lanes (in this Addendum).

10.3.13 Timing for Bus Turnaround

For a BoW memory link with bidirectional data slices, quick turnaround between TX/RX operation is required to support read/write memory traffic.

Bidirectional interfaces generally have an intermediate tri-state (hi impedance) state when switching between TX to RX or RX to TX (“bus turnaround”) to allow settling time and avoid bus contention that would result from being in both TX and RX modes simultaneously. During this bus turnaround, the PHY cannot be used for either read or write, resulting in a timing “bubble” on the memory interface. The bus turnaround bubble for BoW PHYs applied to memory interfaces is expected to be less than 1ns, equivalent to 2 parallel clock cycles at the internal link layer interface.

Memory devices with dynamic bidirectional interfaces are common in the industry, and include standards such as DDR, LPDDR, and HBM. Memory devices receive command packets that instruct them whether a write (memory RX) or read (memory TX) operation is requested. That information together with programmable settings for latency (time between command and data) instructs the memory device when it is expected to receive or transmit the requested data. But, memory devices cannot control when to switch directions on the data bus.

It is important to avoid conflicts on the bidirectional data bus, and the memory controller residing on the ASIC has the responsibility to manage the timings between commands and data, and between write and read commands & data. The memory controller is expected to include the bus turnaround bubble as part of its scheduling algorithms.

11. Power Management

There are no changes to section 11 of the BoW Specification 2.0.

12. Chip-to-Chip Channel Specifications

There are no changes to section 12 of the BoW Specification 2.0.

13. Redundancy

There are no changes to section 13 of the BoW Specification 2.0. As described, redundant lanes are generally required only when using advanced packaging, and normally the optional FEC and AUX lanes are used as the redundant lanes. If redundancy is desired and the FEC and AUX lanes are already assigned to be operated as ECC data in a memory application, then additional lanes must be added for the redundant lanes.

14. Reset and Initialization

Section 14 of the BoW Specification 2.0 describes BoW PHY reset and initialization.

14.1. External Facilities

Section 14.2.1 of the BoW Specification 2.0 describes the elements necessary to initialize a BoW PHY, such as a link controller, training pattern data, PHY reset, etc. There are no changes to this section.

14.2. Initialization Sequence

14.2.1. Initialization Sequence for Memory Controller Chiplets

Section 14.2.1 of the BoW Specification 2.0 describes the standard initialization for a BoW TX-RX link, and there are no changes to this section. This initialization sequence is applicable to the memory controller chiplet use case, which has two standard BoW PHYs with symmetric clocking as described in section 10.2.1 above. The initialization of the two BoW PHYs are independent and can take place simultaneously. For applications that support lane redundancy (as described in section 13), if any lanes fail to initialize properly, then a redundant lane can be employed.

14.2.2. Initialization Sequence for Memory Chiplets

When a BoW PHY is applied to a Memory Chiplet use case with an ASIC being directly attached to a memory chiplet, the ASIC PHY is expected to implement all of the timing control to simplify the memory chiplet PHY design with asymmetric clocking as

described in section 10.2.2 above. As a result, the initialization sequence will be different than that used for a standard BoW PHY.

Memory chiplets will typically have training managed by a link Training Register that runs a lower frequency than the data bus, for example, 0.25 Fmax. Components of the link training register can include loopback mode enable, adjustable cycles of latency, fixed data patterns, VREF trim settings, training cycle count, and management of any lane swapping if redundancy is supported.

A memory chiplet initialization sequence contains the following steps:

1. The ASIC and memory chiplet PHYs are reset as in a normal BoW PHY.
2. The Host (ASIC) releases reset to the memory device.
3. Host configures the memory device by initiating register writes at low frequency (ex: 0.25 Fmax).
4. Host configures the Training Register on the memory device by writing the information including latency, fixed data pattern, and training cycle count (i.e. # of cycles the fixed data pattern will run before existing training mode and resuming normal mode).
5. Host changes the frequency to the desired operating frequency, and then the fixed data pattern will be transmitted from the memory to the Host for the training cycle count while Host receiver timing and VREF trimming are adjusted.
    a. For all the training operating the CLK/CLK# are the timing reference and should not be changed – all timing is relative to this clock.
    b. The Host data inputs will be don't care until the training cycle count number has been reached.
    c. For applications that support lane redundancy (as described in section 13), all available lanes should be trained. If any Host lanes fail to receive the fixed data pattern, then a redundant lane can be employed.
6. Host will reduce the frequency and configure the Training Register on the memory device to enter loopback mode (the training cycle count can also be adjusted accordingly) and set the desired latency from input clock to data out.
7. Host changes the frequency to the desired operating frequency and transmits data to the memory device for loopback, which the memory device then receives and transmits data back to the Host with the configured data latency setting.
    a. In a scenario utilizing two unidirectional BoW PHYs (one for write data and another for read data), the memory device can receive the loopback data and transmit it back to the Host based on the data latency settings without needing to store the data in the memory array.
    b. In a scenario with a BoW PHY utilizing dynamic bidirectional data lanes, the Host issues a Write command and transmits write data, and the memory device receives and stores the data in a write buffer or memory location. The Host issues a Read command to the memory location

where the write data is stored, the data bus is then turned around from write to read data, and the memory device transmits the stored data back to the Host based on the data latency settings.

8. Host receives the loopback data from the memory device and adjusts the Host Tx data timing relative to the clock to identify the best window for the memory device receiver.
    a. For applications that support lane redundancy (as described in section 13), all available lanes should be trained. If any memory device lanes fail to receive the loopback data pattern from the Host, then a redundant lane can be employed.
9. The Host can repeat the process if the memory device supports VREF trimming to find the best data eyes.
10. The Host and memory then enter normal operation mode.

14.2.3. Initialization Sequence for BoW Bidirectional Memory

When a BoW PHY using dynamic bidirectional signaling is applied to an application such as the Standard Package HBM (SPHBM) use case with clocking described in section 10.2.3 above, separate initialization sequences are needed for ASIC-to-memory traffic (Cmd/Add and write data) and for memory-to-ASIC read data traffic. The initialization sequence for each direction will be similar to that described for a standard BoW link in section 14.2.1 of the BoW 2.0 Specification, with implementation-specific procedures detailed in the relevant memory product specification. Most memory products include an IEEE 1500 port to facilitate memory device initialization, training, and register configuration. For applications that support lane redundancy (as described in section 13), if any lanes fail to initialize properly, then a redundant lane can be employed.

14.3. Bidirectional Slice Turnaround

As described in the BoW 2.0 Specification, in order to reverse direction on a standard BoW PHY link with bidirectional slices on both ends, it is expected that a full reset, reconfiguration and initialization of the slices will be performed. This full procedure is expected to require microseconds to milliseconds, but will be implementation dependent.

In contrast, for a BoW memory link, quick bidirectional slice turnaround is required to support read/write memory traffic. As described in section 10.3.12, the bus turnaround timing for BoW PHYs applied to memory interfaces is expected to be less than 1ns, equivalent to two parallel clock cycles at the internal link layer interface.

15. Configuration

There are no changes to section 15 of the BoW Specification 2.0.

16. Control Register Mapping

There are no changes to section 16 of the BoW Specification 2.0.

17. Testability

There are no changes to section 17 of the BoW Specification 2.0.

18. Sideband Slices

There are no changes to section 18 of the BoW Specification 2.0.

19. BoW in an ODSA Design

There are no changes to section 19 of the BoW Specification 2.0.