



OPEN

Compute Project

OpenHBI-L Specification for Laminate substrate

V1.0 RC2

Authors:

Kenneth Ma – AMD (Chair)

Andy He – Google (Vice-Chair)

Alphawave – Dustin Dunwell, Letizia Giuliano, Bill Beale, Alan Chan

AMD – Millind Mittal, Ge Chang, Xiaobao Wang

Arm – Javi DeLaCruz, Mikael Rien, Jean-Claude Duby, Fabio Toni Braz

Cadence – Jeff Bostak, Vikas Agrawal, Sue-Hung Fung

SiFive – Juan-Carlos Calderon

Synopsys – Peter Nyasulu, Octavian Beldiman, Brian Worobey, Tri Vo

Teradyne – Ed Seng

Table of Contents

1.	License (OCP OpenHBI Specification)	8
1.1	Open Web Foundation (OWF) CLA	8
1.2	Acknowledgements	8
2.	OCP Tenets Compliance	9
2.1	Openness	9
2.2	Efficiency	9
2.3	Impact	9
2.4	Scale	9
3.	Revision Table	10
4.	Scope	11
4.1	In Scope	11
4.2	Out of Scope	11
5.	Overview	12
5.1	Key Features	13
5.2	OpenHBI-L Interface Unit (DWORD or DW)	13
5.3	DWORD Signal Descriptions	14
5.4	OpenHBI-L Scalability	14
5.5	Multi-layered Interface	15
6.	OpenHBI-L Bump Map and Orientation	17
6.1	Physical Orientation Convention	17
6.2	DWORD Bump Map	17
6.3	SAME orientation	19
6.4	ROTATED Orientation	20
6.5	Peer-to-Peer Use Case	20
6.6	IP Reuse	21
6.7	Bit Reordering enables seamless Orientation-Independent routing	22
6.8	Single IP Design	23
7.	PHY Layer	24
7.1	PHY Layer Functions	24
8.	Logical PHY Layer	26
8.1	Bit Reordering	27
8.2	Data Bus Inversion – Enhanced 9-bit DBI mode	27
8.3	Framing and Alignment	28
8.4	Parity	30
8.5	Options and Modes	30
8.6	Interface Pin Assignments per Mode	31
9.	Chiplet Configuration interface and Sidebands	32

9.1	MIPI I3C as Configuration Interface.....	32
9.2	OpenHBI-L Sideband Signals	32
9.3	LFSR circuits for Testing and Training patterns.....	33
9.4	OpenHBI-L Training and Testing	34
10.	OpenHBI-L Routability and Reference Stack-Up	38
10.1	Bump Pitch.....	38
10.2	Routing Strategy Examples	38
11.	OpenHBI-L IO Electricals	42
11.1	Transmitter Electrical Parameters	43
11.2	Receiver Electrical Parameters	44
11.3	Channel Parameters	45
11.4	REFCLK Specifications	47
11.5	BER spec.....	47
12.	Clocking, Initialization and Configurations.....	48
12.1	Clocking.....	48
12.2	Initialization Flow.....	51
12.3	Capability, Control and Status (CCS) Registers	52
12.4	Calibration and Training	57
12.5	Continuous Training during operation.....	58
13.	Power Management.....	60
14.	References	61
Appendix A - Requirements for IC Approval (to be completed Contributor(s) of this Spec)		62
Appendix B-_____ - OCP Supplier Information (to be provided by each Supplier of Product)		63

Glossary of Terms

This section provides glossary used in this specification. Note that it is not organized in alphabetical order but in sequential order to best understand the terms and definitions of OpenHBI.

Abbreviation	Term	Description
OpenHBI	Open High Bandwidth Interface	An optimized chiplet interconnect comprises of multiple DWORD channels. In this specification, if the term OpenHBI is used without Version number or -L suffix, the statement will apply to all specifications in OpenHBI family of specifications.
OpenHBI-L	OpenHBI Specification for Laminate substrate	Name of this specification. OpenHBI-L provides the specification of the features to be used with lower cost and broadly available laminate substrate.
Standard Profile	Standard Profile	OpenHBI-L Anchor and Chiplet shall support Standard Profile data rates of 4 Gbps, 8Gbps and 16 Gbps per lane at BER $\leq 1e-25$
Enhanced Profile	Enhanced Profile	OpenHBI-L Anchor and Chiplet may optionally support the Enhanced Profile data rates of 10, 20, 32 or 40 Gbps per lane at BER $\leq 1e-15$
DWORD or DW	Data Word	DWORD is the basic building block of OpenHBI interface. Each DWORD consists of up to 42 data-carrying signals, RX clock pair, TX clock pair. DWORD and DW will be used interchangeably in this specification.
Beachfront or BF	Beachfront	The length along the die edge that is occupied by the OpenHBI C2C interface
C2C, D2D, Chiplet interface	Chip-to-Chip Die-to-Die Chiplet Interface	The C2C, D2D and Chiplet interface terms will be used interchangeably in the OpenHBI specification depending on the contexts.
C4 bump	C4 bump	C4 bump is an abbreviation of “Controlled Collapse Chip Connection” which is a method for interconnecting dies on packaging substrate.
Y-pitch	Y-pitch	OpenHBI-L C4 bump-to-bump pitch along the beachfront
X-pitch	X-pitch	OpenHBI-L C4 bump-to-bump pitch towards the center of the anchor or chiplet die (typically perpendicular to the beachfront)
Anchor	Anchor die	Anchor die refers to a chiplet that can provide configuration, management, test services and/or can support multiple chiplets on one or more of its die edges. The notion of Anchor is optional if it uses an external controller to provide the configuration and management services.

PHY	Physical Layer	The low-level hardware interface of a high-speed input/output protocol
Logical PHY	Logical PHY layer	A thin layer above PHY layer to provide optional functions and services including Parity, Framing, DBI and Bit Reordering.
Link Layer	Link Layer	Link layer is a layer above Logical PHY layer. It supports optional DWORD aggregation function, flow control and data integrity and reliability services.
IDL interface	Inter-Die Link (IDL) interface	IDL abstracts the C2C transport. C2C transport includes the PHY layer, Logical PHY and Inter-Die Link (IDL) Layer.
IDL adaptation layer	IDL adaptation layer	IDL adaptation layer adapts the OpenHBI C2C transport to the IDL interface for mapping to the protocol layer(s).
DBI	Data Bus Inversion	Data Bus Inversion is an optional normative service to minimize signal toggling on the physical wire. If the DBI service is enabled, it counts the number of data transitions to ensure no more than 50% of the data wires (including DBI signal) will toggle to reduce power and improve signal integrity.
CCS Registers	Capability, Control & Status Registers	A set of reference Capability, Control and Status Registers.
Gearbox		Speed/width conversion function in a SERDES
SERDES		SERializer - DESerializer
TX	Transmit	Prefix for Transmit-side; e.g., TX_Clk, TX DWORD
RX	Receive	Prefix for Receive-side; e.g., RX_Clk, RX DWORD
UI	Unit Interval	The time period of a single data bit on the interface bus (= ½ of TX_Clk / RX_Clk period)
Gbps or Gb/s		Gigabits per second – unit of data rate
Tbps or Tb/s		Terabits per second – unit of data rate
Gbps/mm		Gigabits per second per mm – unit of bandwidth density
Tbps/mm		Terabits per second per mm – unit of bandwidth density
IO or IOs	Input / Outputs	Input receiver and Output driver
pJ/bit or pJ/b		Picojoule / bit – unit of energy per bit transferred
PDN	Power Delivery Network	The power delivery network is the current path from the power supply (or power supplies) through packaging substrate to the silicon die.
PPA		Power, Performance and Area. This term describes some key metrics in silicon or IP design considerations.
FIT	Failure In Time	Number of failures that can be expected in 1 billion (1e9) device-hours of operation.

GDSII	GDSII stream format	A database file format which is the de facto industry standard for data exchange of integrated circuit or IC layout.
-------	---------------------	--

Language

OpenHBI specification adopts the use of the words “shall”, “must”, “should”, “may” and “can” as defined below:

Term	Description
“shall” or “must”	It indicates a normative requirement strictly to be followed. Failure to meet the requirement results in non-conformance.
“should”	It indicates a strong suggestion or recommendation, but not a requirement. Failure to implement the suggestion does not result in non-conformance.
“may”	It indicates “it is permitted to” implement or take a course of action.
“can”	It indicates “it is able to” or used as a statement of possibility and capability.

1. License (OCP OpenHBI Specification)

1.1 Open Web Foundation (OWF) CLA

Contributions to this Specification are made under the terms and conditions set forth in the **modified** Open Web Foundation Contributor License Agreement (“OWF CLA 1.0”) (“Contribution License”) by:

ALPHAWAVE, AMD, ARM, CADENCE, GOOGLE, SIFIVE, SYNOPSYS, TERADYNE

You can review the signed copies of the applicable Contributor License(s) for this Specification on the OCP website at <http://www.opencompute.org/products/specsanddesign>

Usage of this Specification is governed by the terms and conditions set forth in the **modified Open Web Foundation Final Specification Agreement (“OWFa 1.0”) (“Specification License”)**.

Notes:

1) The following clarifications, which distinguish technology licensed in the Contribution License and/or Specification License from those technologies merely referenced (but not licensed), were accepted by the Incubation Committee of the OCP:

“None”

2) 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.

1.2 Acknowledgements

The Contributors of this Specification would like to acknowledge the following companies and members for their feedback (list in alphabetical order of the member companies) :

Cadence – Ashish Agrawal

Synopsys – Khanh Pham, Bao Anh Nguyen

2. OCP Tenets Compliance

2.1 Openness

OpenHBI-L is part of the OpenHBI family of specifications to build an open, public and interoperable OpenHBI IP and Chiplet ecosystem. All WG members agreed to grant the Necessary Claims by executing the modified OWFa 1.0 CLA and FSA to enable user implementation of the OpenHBI-L specification.

To enable cross-vendor interoperability, the spec has provisions to enable KGD (known good die) testing and post-packaged final test utilizing standard MIPI I3C and JTAG as configuration and test interfaces.

2.2 Efficiency

Bandwidth efficiency: OpenHBI-L enables high-performance chiplet applications on low-cost organic laminate substrate. It can deliver up to 2.4 Tbps/mm in Enhanced profile at BER $\leq 1e-15$ and up to 960 Gbps/mm in Standard profile with increased data reliability of BER at $\leq 1e-25$.

Routing efficiency: OpenHBI-L provides the flexibility based upon design rules to route the full signal density in a 6-2-6 laminate, a 8-2-8 laminate or a greater layer count with full coplanar ground shield support for enhanced signal integrity.

Design Efficiency: OpenHBI-L enables a single IP design for use in Anchor and Chiplet dies and can support any beachfront orientation seamlessly, for example, in homogeneous peer-to-peer use case.

2.3 Impact

We believe OpenHBI-L is the highest performance chiplet standard for organic laminate substrate that support up to 2.4 Tbps/mm. OpenHBI-L enables the advantage of routability in a 6-2-6 at the full beachfront bandwidth with lower cost packaging substrate.

2.4 Scale

OpenHBI-L performance is scalable in beachfront height and depth up to 2DW-deep. A typical instance consisting of 8DW supports between 1.34 to 12 Tbps bandwidth, enough to drive extremely high performance applications. The bandwidth scales linearly which is only limited by size of the die edge.

3. Revision Table

Date	Revision #	Author	Description
2/28/2022	0.5	OHBI WG	Initial Draft. Updated beachfront scalability guidelines Section 6: DW Bump Map and Orientation Section 12 update
3/31/2022	0.7	OHBI WG	Standard and Enhanced Profiles definitions Section 6: DW bump map with PDN and symmetry. Section 9: Removed SETLRR, GETLRR Section 10: Routability and Reference Stack-Up - 6-2-6 and 8-2-8 Routing examples - Reference Stackup Section 11: IO Electrical section with TBD parameters Section 9 and Section 12 - LFSR, Testing, Calibration and Training
5/4/2022	0.9RC1	OHBI WG	Section 9.3: Add Programmable Pattern mode Section 9.4: Updated Training and Testing steps Section 9.4.3: Add example implementation to generate 80-bit per clock with the 40-bit Galois LFSR Section 11: IO Electrical section update Section 12.3.2.4: Add "Programmable Pattern mode" to generate pre-programmed 40-bit pattern continuously Section 12.3.2.5: Add "LFSR Initial Value Register"
5/13/2022	0.9RC2	OHBI WG	Section 11: Added Insertion Loss and Crosstalk masks
5/29/2022	0.9RC3 / 1.0RC1	OHBI WG	Incorporated review feedback and edits by Teradyne
6/29/2022	1.0RC2	OHBI WG	Updated diagrams and spec with Inter-Die Link (IDL) nomenclature

4. Scope

4.1 In Scope

This document defines the technical specifications for the OpenHBI-L under OCP Server project, ODSA subproject. Scope of OpenHBI-L includes but not limited to the following:

- DWORD and Bump signals definitions
- PHY layer
- Logical PHY layer
- DWORD Bump pattern and signal ordering
- Beachfront Orientation considerations
- Beachfront Scalability
- Bump pitch (Min-Max range)
- IO electricals, Data Rates and Clocking
- Chiplet Configuration interface
- Test pattern generator and signal mapping
- Reference laminate substrate media and channels
- Reference Stackup and routing examples
- Reference Initialization, Calibration and Training steps and register set.

4.2 Out of Scope

- Upper layer definition (Refer to Reference [4] ODSA OpenHBI Inter-Die Link Layer Specification)
- Use of OpenHBI-L outside of a package
- Security features such as Authentication, Integrity Check, Encryption

5. Overview

The ODSA OpenHBI-L or Open High Bandwidth Interface specification for Laminate substrate defines a high-performance die-to-die or chiplet-to-chiplet interconnect on a medium such as organic laminate packaging or similar technologies.

The OpenHBI-L defines an optimized C2C interface specification which is comprised of DWORD channels. It is intended for low cost, high bandwidth IO expansion (e.g., SerDes, ADC/DAC, networks, optical chiplets), Accelerators and Memory expansion (e.g., PCIe/CXL chiplets) and as the transport of popular on-die protocols for applications in Datacenter, HPC, AI, networking and other applications that demands high performance, low latency and energy efficient C2C interconnect.

OpenHBI-L “Standard Profile” defines IO electrical specification that enables a high reliability interface at BER (Bit Error Ratio) $\leq 1e-25$. Anchor and Chiplet shall support Standard Profile data rates of 4 Gbps, 8 Gbps and 16 Gbps per lane.

OpenHBI-L “Enhanced Profile” is optional normative and supports increased data rate of up to 40 Gbps per lane at a BER $\leq 1e-15$ for protocols and applications requiring highest bandwidth possible that support their own data reliability and integrity mechanism. Anchor and Chiplet may optionally support the Enhanced Profile data rates of 10 Gbps, 20 Gbps, 32 Gbps and/or 40 Gbps. Note that, if a chiplet supporting 40 Gbps data rate shall also support 20 Gbps and 10 Gbps data rates. Similarly, a chiplet supporting 20 Gbps data rate shall also support 10 Gbps data rate.

OpenHBI-L defines guidelines for beachfront scalability, configuration interface, interoperability and testability between heterogenous chiplets, especially for chiplets from different vendors.

Figure 5-1 illustrates an example OpenHBI-L chiplet-based device with an Anchor die with chiplets for Datacenter connectivity such as networking, switch or backplane, for Optical interconnect, and PCIe/CXL connectivity for Accelerator and pooled memory expansion use cases.

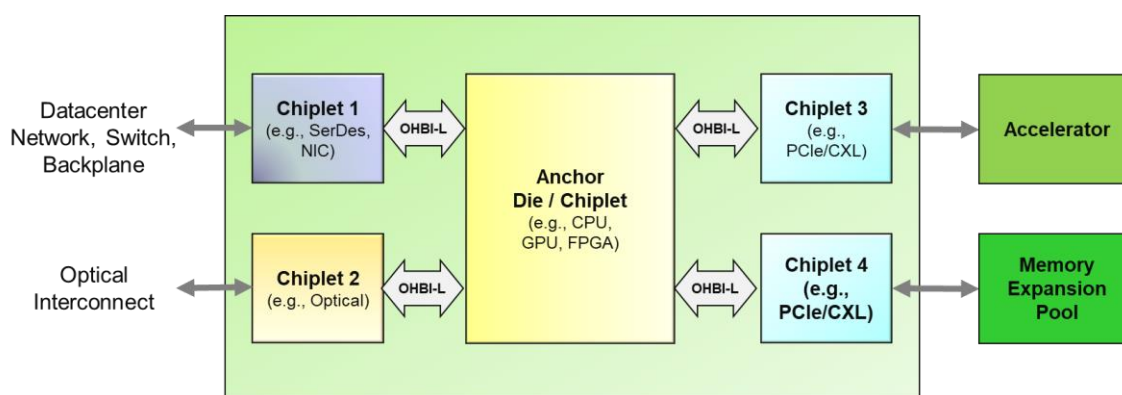


Figure 5-1: Example OpenHBI-L chiplet-based device and applications

5.1 Key Features

- Supports low cost, broadly available laminate packaging substrate, e.g., organic laminate
- Linear (beachfront) bandwidth density up to ~ 1Tbps per mm (TX+RX bandwidth)
- Data Rates and BER target
 - Standard Profile: 4, 8 and 16 Gbps per lane at BER $\leq 1e-25$
 - Enhanced Profile: 10, 20, 32 or 40 Gbps per lane at BER $\leq 1e-15$
- Standard Profile is defined to deliver a reliable link (< 1 FIT) at a reach up to 20mm trace reach
- Multi-channel chip-to-chip interface
 - Optional programmable DW direction (e.g., supports unbalanced read/write traffic)
- Optimized IO Electricals
 - RX-terminated I/O (50 ohm nominal)
 - 0.75 I/O supply rail
 - Power target ≤ 0.5 pJ/bit (average energy per bit transferred including TX PHY/IO + RX IO/PHY)
- Orientation-Independent Routing
 - Seamless C2C interface routing with any combinations of anchor and chiplets C2C beachfront orientation
- Scalable
 - Bandwidth and beachfront (number of DWORDS) to match target use cases.
- Chiplet Configuration interface
- C2C Interface Test features

5.2 OpenHBI-L Interface Unit (DWORD or DW)

The basic OpenHBI-L communication unit is **DWORD** or **DW** for short. A DWORD consists of up to 42 data wires and 4 clock/strobe wires.

A DWORD may support either configurable (bidirectional) or fixed (pre-configured TX or RX) directionality.

A bidirectional DWORD is configurable to operate as TX (output) or RX (input) DWORD at initialization, or optionally during run-time, via configuration bus. The choice applies to all 42 DWORD data signals.

An Anchor shall implement configurable bidirectional DWORDS unless it is designed to only support chiplets of known and fixed configurations.

A chiplet may implement fixed direction DWORDS only if it has a known configuration and/or it will be connected to Anchor with configurable bidirectional DWORDS.

The DWORD structure is symmetric in the sense that there is no difference in its design between Anchor or chiplet roles (i.e., no notion of 'Manager' or 'Subordinate').

In operation, each DWORD is unidirectional. A chiplet TX DWORD transmitting data will drive the data signals and the differential transmit clock pair TX_Clk to the chiplet on the receiving side. A chiplet receiving data will receive the data signals and will accept the clock pair via the receive clock inputs RX_Clk. Clock edges shall be phase-aligned with D[41:0] data signals relative to the data eye.

5.3 DWORD Signal Descriptions

Table 5-1 describes OpenHBI-L DWORD signals and their high-level functions.

Table 5-1: OpenHBI-L DWORD Signals

Symbol	OpenHBI-L functions	TX-side	RX-side	Transport Data
D[41:0]	Data signals	Out	In	Yes
TX_Clk	TX_Clk / Differential TX clock pair	Out	Not used	No
RX_Clk	RX_Clk / Differential RX clock pair	Not used	In	No

D[41:0] are data signals that nominally carries data between two chiplets.

TX_Clk is differential output clock pair on TX DWORD.

RX_Clk is differential input clock pair on the RX DWORD.

In normal configuration, TX_Clk differential output clock pair on TX side drives the RX_Clk differential input clock pair on the RX side.

The RX_Clk pair on TX DWORD and TX_Clk pair on RX DWORD are not needed typically for known TX and RX DWORD configuration, and may be left as “No Connect” as long as the RX_Clk pair inputs on TX side are pulled to proper valid state and not floated. For use cases where the DWORD directions are configurable during initialization or at run-time, both the TX_Clk and RX_Clk pairs must be connected to enable dynamically re-configurable operations.

5.4 OpenHBI-L Scalability

OpenHBI-L defines DWORD as basic building block of the C2C interface.

Each DWORD can support raw bandwidth up to 672 Gbps (42 lanes x 16 Gbps) in Standard Profile or up to 1680 Gbps (42 lanes x 40 Gbps) in Enhanced Profile.

The OpenHBI-L C2C interface bandwidth can be scaled up by using multiple DWORD units in both TX and RX directions.

The OpenHBI-L C2C interface may consist of an arbitrary number of DWORDs, with a minimum of one DWORD.

To facilitate interoperability, routability and configuration management, the specification recommends the following guidelines when constructing the C2C beachfront unless the Anchor or chiplet is designed to only support chiplet(s) or Anchor(s) of known and fixed configurations:

Beachfront guidelines:

- OpenHBI-L beachfront should support minimum of 2DW with any number of 2DW-increment.
- OpenHBI-L beachfront should be either 1DW- or 2DW-deep and should not exceed 2DW-deep.

Interoperability Rules and Guidelines:

- An Anchor shall implement configurable bidirectional DWORDs unless it is designed to only support chiplets of known and fixed configurations.
- 2DW-deep Anchor shall support Chiplets of either 1DW- or 2DW-deep

5.5 Multi-layered Interface

To enable flexible and modular architecture, OpenHBI family of specifications adopts a reference multi-layered interface model compatible with ODSA stack concept as shown in Figure 5-2.

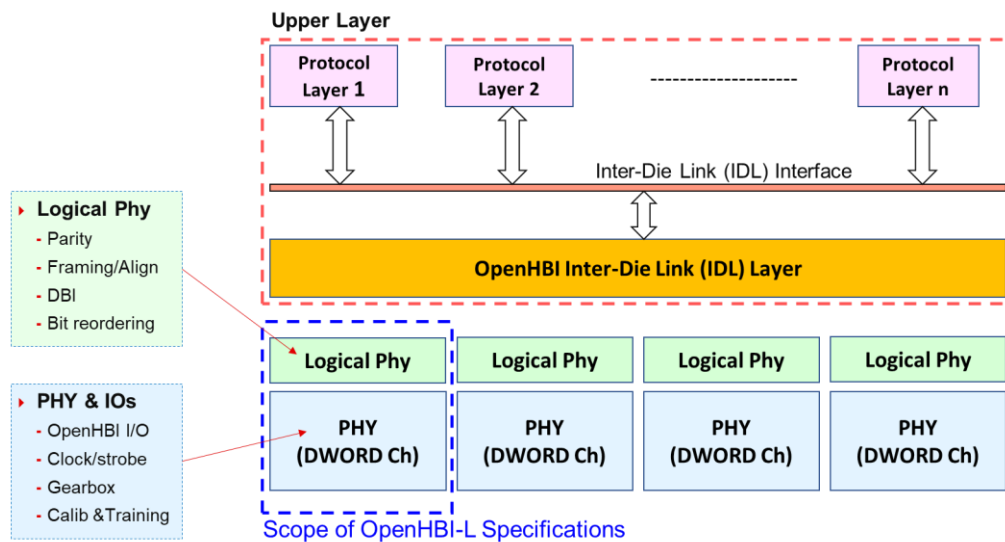


Figure 5-2: OpenHBI-L Reference Multi-Layered Model

The OpenHBI-L defines two layers, PHY and Logical PHY layers:

5.5.1 PHY Layer

This layer defines OpenHBI-L PHY which performs the transmit, receive, clocking, serializing/deserializing (aka Gearbox) functions during operations, calibration and the data-to-clock training functions during initialization.

5.5.2 Logical PHY Layer

This layer supports optional-normative functions such as bit reordering, DBI (data bus inversion), framing and parity generation/check. The bit reordering function is to ease the routing between two chiplets in ROTATED orientation.

The OpenHBI-L C2C transport can support various high-level protocols (e.g., CXS, AXI, PCIe, CXL) via the “Upper Layer” which may consist of IDL interface, IDL adaptation layer and protocol layer.

The Link layer supports optional DWORD aggregation function, flow control and data integrity and reliability services. The IDL adaptation layer adapts the OpenHBI C2C transport to the IDL interface for mapping to the protocol layer(s).

Figure 5-3 illustrates conceptual OpenHBI Anchor and chiplet operations in layered architecture.

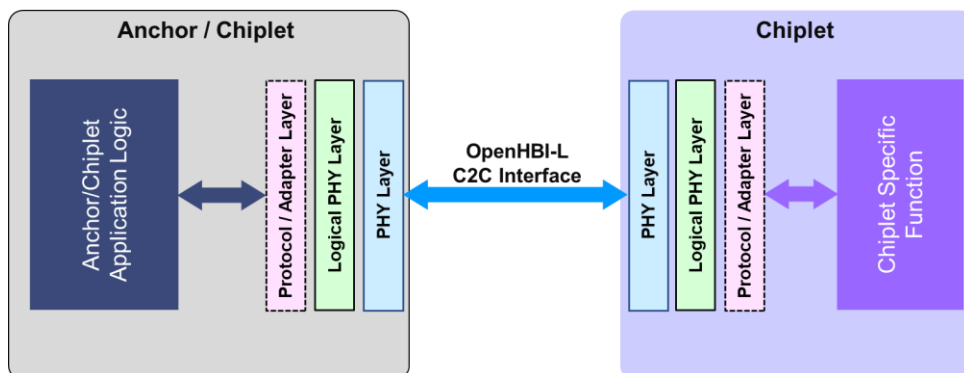


Figure 5-3: OpenHBI Anchor and chiplet layered architecture

Different function chiplets may implement the layers only required by their functions. The minimum set of layers required for normal operation are PHY and Logical PHY layers. It is therefore desirable to optimize the design, complexity, latency, and PPA (Power, Performance and Area) of these 2 layers.

6. OpenHBI-L Bump Map and Orientation

6.1 Physical Orientation Convention

This specification allows OpenHBI-L interfaces along any of the 4 die edges of an Anchor, referred to as Top, Right, Bottom and Left edges. As illustrated in the example in Figure 6-1, OpenHBI-L specification defines the beachfront bump pattern shall rotate 90° clockwise when traversing the Top → Right → Bottom → Left edges of an Anchor.

For clarity of the specification, the subsequent orientation descriptions and examples refer to a side-by-side chiplet orientation with OpenHBI-L signals flowing horizontally between chiplets. Unless otherwise noted (and without loss of generality), the descriptions and examples will apply to any one of the 4 die edges and any direction of the signal flow and trace routes.

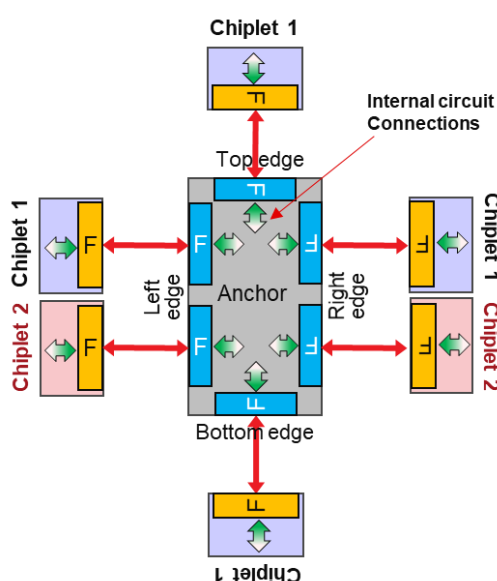


Figure 6-1: Example OpenHBI-L Anchor with chiplets on Top, Right, Bottom and Left edges

6.2 DWORD Bump Map

OpenHBI-L specification defines a reference DWORD bump map.

OpenHBI-L DWORD signal ordering is normative, that is, the Anchor or Chiplet implementation of the order of DWORD signal bumps shall follow the reference bump map in this section to enable optimal routability on optimal number of routing layers.

Designers are strongly recommended to also follow the PDN (VDD and GND) bumps of the DWORD bump map but it is not mandatory as long as the power integrity and signal integrity requirements are met and can meet the BER target at specified max data rate.

OpenHBI-L bump patterns have the notions of “True” and “R180” (Rotated 180-degree) orientations. The specification defines four bump map orientations as below:

- Chiplet “True” orientation is the native bump map of Chiplet DW facing Right die edge
- Chiplet “R180” orientation is the native bump map of Chiplet DW facing Left die edge
- Anchor “True” orientation is the native bump map of Anchor DW facing Left die edge
- Anchor “R180” orientation is the native bump map of Anchor DW facing Right die edge

Figure 6-2 illustrates “Chiplet-True”, “Chiplet-R180”, “Anchor-True” and “Anchor-R180” reference bump maps. Note the bump locations relative to the Chiplet and Anchor die edges in each orientation below.

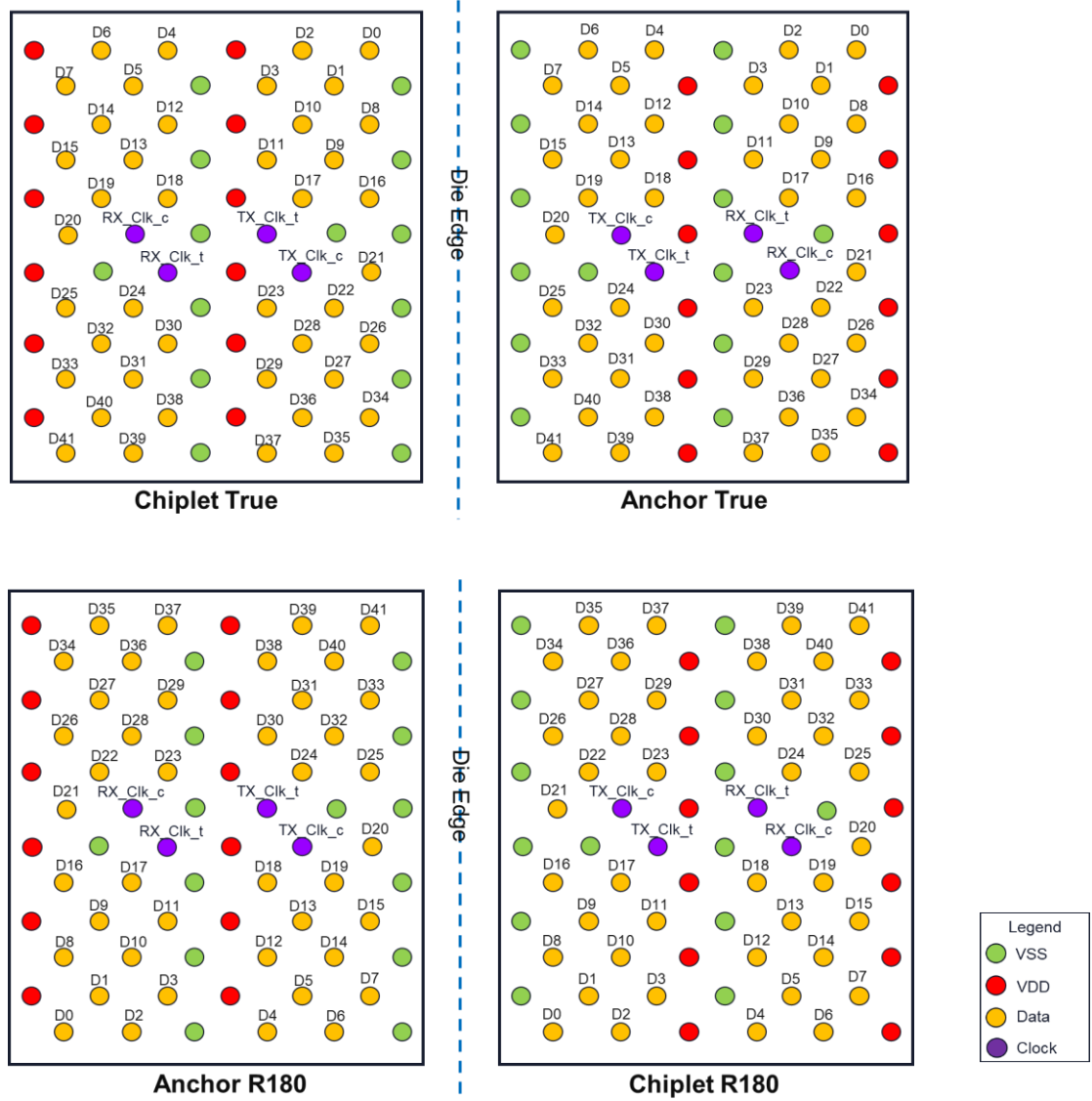


Figure 6-2: OpenHBI-L Chiplet True/R180 and Anchor True/R180 reference bump maps

Each DWORD bump map has 12 rows by 12 columns of C4 bumps in checkerboard arrangement with 72 C4 bumps containing 46 signals, VCC and GND rails.

- 42-bit Data (D[41:0])
- Differential TX_Clk_t/_c output clock pair
- Differential RX_Clk_t/_c input clock pair

The “Chiplet True” and “Chiplet R180” orientation are simply 180-degree rotation of each other.

Similarly, the “Anchor True” and “Anchor R180” are simply 180-degree rotation of each other.

A Chiplet shall be designed to support “Chiplet True” C2C orientation.

An Anchor shall be designed to support “Anchor True” C2C orientation.

It is important to note that the TX_Clk_t/_c and RX_Clk_t/_c bumps of the Anchor and Chiplet are arranged to be “rotation-invariant” so that they will always connect seamlessly in straight and length-matched connections regardless of SAME orientation or ROTATED orientation.

6.3 SAME orientation

Two C2C beachfronts are considered to be in “SAME orientation” when they are connected in True-to-True or R180-to-R180 orientation.

In SAME orientation, all signals connect seamlessly (straight connections).

Typically, by following the convention and rules as defined in Section 6.1, the Anchor and Chiplet C2C connections will always be in the SAME orientation with the exceptions described in Section 6.4 “ROTATED Orientation” below.

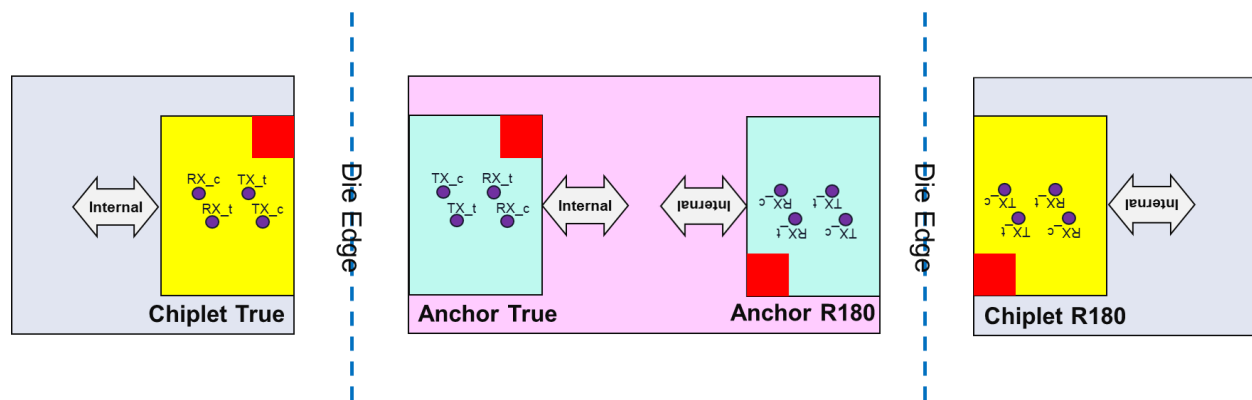


Figure 6-3: SAME orientation connection examples

6.4 ROTATED Orientation

Two C2C beachfronts are considered to be in “ROTATED orientation” when they are connected in True-to-R180 or R180-to-True orientation.

For OpenHBI-L compliant Anchor or chiplet, ROTATED orientation will only happen in the following two cases:

- (1) Peer-to-Peer Use Case: When an Anchor or chiplet can support both Anchor and chiplet roles and connecting to itself to form a multi-dies cluster. Refer to Section 6.5.
- (2) IP Reuse: The same IP is used in both Anchor and chiplet and connecting to each other. Refer to Section 6.6.

OpenHBI-L supports seamless straight connections independent of SAME or ROTATED orientation, enabled by the Bit-Reordering feature. In ROTATED orientation, there is no need to connect the data signals by their names (e.g., D[0] to D[0], D[1] to D[1].....D[41] to D[41]) which would result in a crisscrossed routing. Refer to Section 6.7 for more details.

6.5 Peer-to-Peer Use Case

In Peer-to-Peer use case, an Anchor or chiplet is connected to itself and the C2C and DWORD orientation inevitably rotated by 180° facing each other, as illustrated in Figure 6-4.

One of the two dies (Anchor or chiplet) will act as configuration “Manager” and the other die (Anchor or chiplet) will act as configuration “Subordinate”. The mechanism to configure the Manager and Subordinate roles is out of scope of the OpenHBI-L specification.

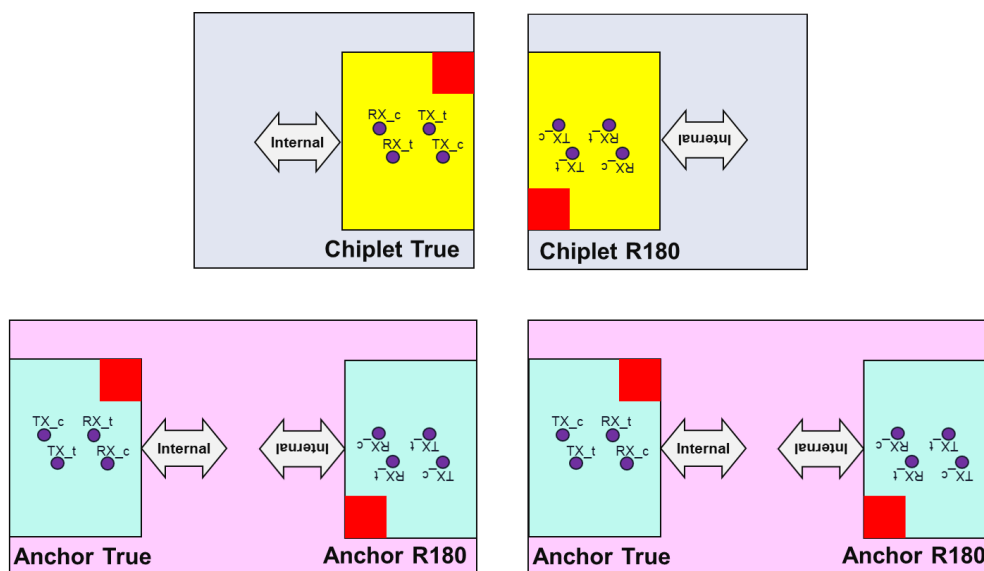


Figure 6-4: ROTATED orientation in Peer-to-peer use cases

6.7 Bit Reordering enables seamless Orientation-Independent routing

The Bit Reordering feature enables seamless, straight and length-matched traces in the ROTATED orientation. The Bit Reordering feature is optional-normative.

Table 6-1 shows the connection map if the TX and RX DWORD signals are connected in straight routes in ROTATED orientation.

Table 6-1: ROTATED Orientation Connection Map

TX DW	RX DW	TX DW	RX DW	TX DW	RX DW	TX DW	RX DW	TX DW	RX DW
d0	d41	d10	d31	d20	d21	d30	d11	d40	d1
d1	d40	d11	d30	d21	d20	d31	d10	d41	d0
d2	d39	d12	d29	d22	d19	d32	d9	TX_CLK_t	RX_CLK_t
d3	d38	d13	d28	d23	d18	d33	d8	TX_CLK_c	RX_CLK_c
d4	d37	d14	d27	d24	d17	d34	d7	RX_CLK_t	TX_CLK_t
d5	d36	d15	d26	d25	d16	d35	d6	RX_CLK_c	TX_CLK_c
d6	d35	d16	d25	d26	d15	d36	d5		
d7	d34	d17	d24	d27	d14	d37	d4		
d8	d33	d18	d23	d28	d13	d38	d3		
d9	d32	d19	d22	d29	d12	d39	d2		

Figure 6-6 illustrates the sequence of events from TX DWORD PHY outputs (on the left) towards RX DWORD PHY inputs (on the right). The RX D[x] inputs will be carrying TX D[41-x]. The Bit Reorder Mux (“F”) will perform the bit shuffling function as shown in Table 6-2 by reordering the input “E” to output “G” so that the RX D[x] will present the corresponding TX D[x] to the next layer transparently.

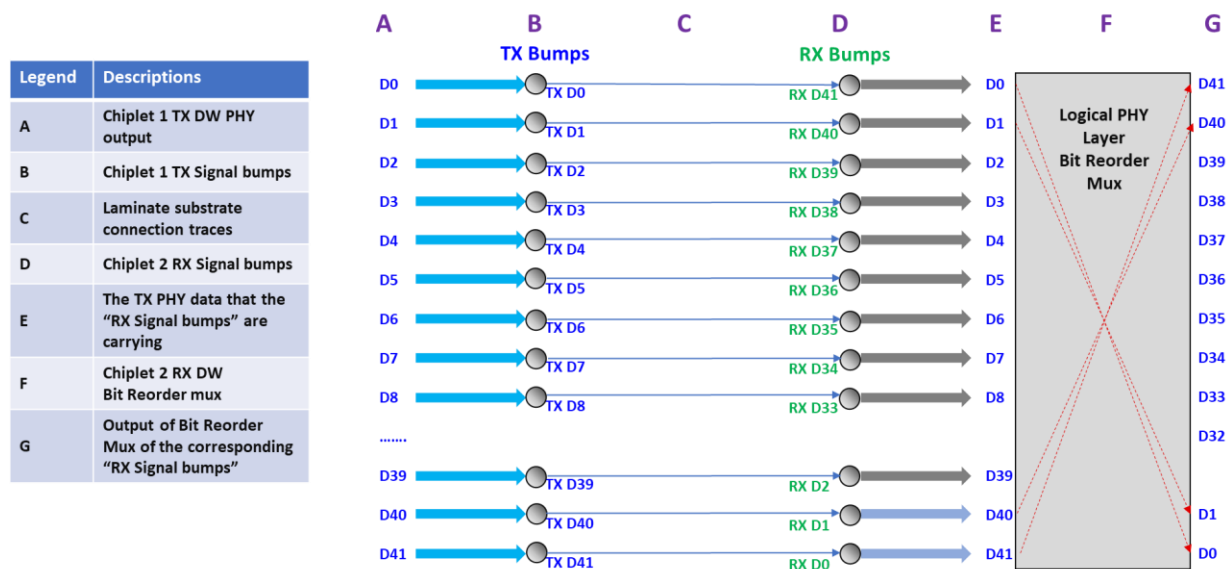


Figure 6-6: Event sequence from TX DW PHY to RX DW PHY with Bit Reordering

If the Anchor and Chiplet desire to support ROTATED orientation use case, the RX DWORD Logical PHY must support the (Optional Normative) Bit Reordering function in Table 6-2 in order to be routable.

Table 6-2: Bit reordering performed by the Bit Reorder Mux (“F” in Figure 6-6)

In “E”	Out “G”	In “E”	Out “G”	In “E”	Out “G”	In “E”	Out “G”	In “E”	Out “G”
d0	d41	d10	d31	d20	d21	d30	d11	d40	d1
d1	d40	d11	d30	d21	d20	d31	d10	d41	d0
d2	d39	d12	d29	d22	d19	d32	d9		
d3	d38	d13	d28	d23	d18	d33	d8		
d4	d37	d14	d27	d24	d17	d34	d7		
d5	d36	d15	d26	d25	d16	d35	d6		
d6	d35	d16	d25	d26	d15	d36	d5		
d7	d34	d17	d24	d27	d14	d37	d4		
d8	d33	d18	d23	d28	d13	d38	d3		
d9	d32	d19	d22	d29	d12	d39	d2		

6.8 Single IP Design

OpenHBI-L specification provides the benefits that a single IP design can be used in both Anchor and chiplet, and with the Bit Reordering function, it can also support Peer-to-Peer use cases seamlessly.

7. PHY Layer

7.1 PHY Layer Functions

As shown in Figure 7-1, the PHY layer contains the TX_Clk and RX_Clk pairs, up to 42 data signals and it performs the following functions:

- Speed/width conversion function (aka Gearbox function)
- Calibration and training function

As the OpenHBI-L IO data rate (e.g., Phy_Bitrate = 16 Gbps) can be substantially higher than the Logical PHY layer or core logic operating frequency (e.g., 1-2 GHz), the PHY layer can perform R:1 serialize / deserialize (Gearbox) functions and convert the frequency between two domains, where the ratio $R = \text{Phy_Bitrate} / \text{LogPhy_Clk}$ and must be an 2^n integer value.

The number of signals between PHY layer and Logical PHY or Upper Layer will be at least $42 * R$ signals. For example, if Phy_Bitrate = 16 Gbps and LogPhy_Clk = 2 GHz, then $R = 8$.

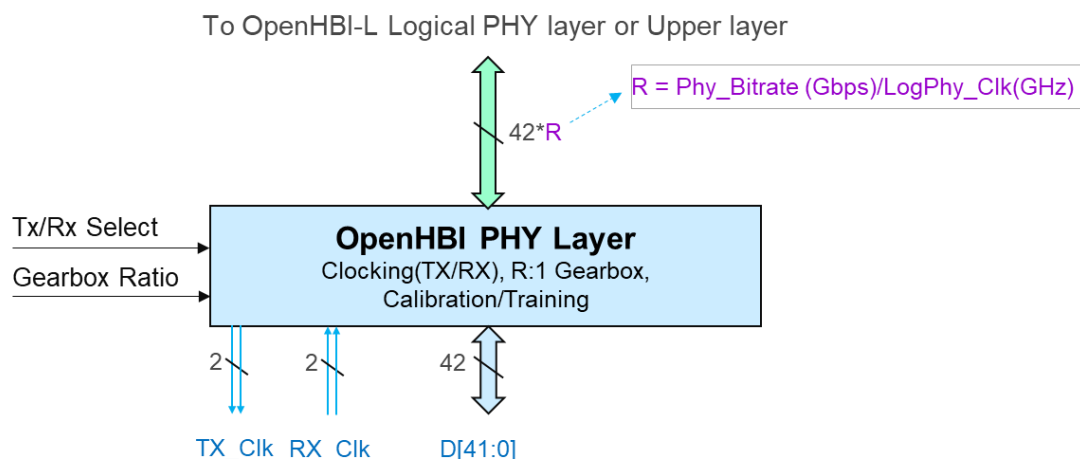


Figure 7-1: OpenHBI-L PHY layer functions

7.1.1 Data I/O and Clock/Strobe

OpenHBI-L adopts DDR (double-data rate) signaling, i.e., the data rate doubles the TX_Clk and RX_Clk frequency.

The OpenHBI-L PHY layer generates TX_Clk (output) and uses RX_Clk (input) to sample the input D[41:0].

The TX_Clk shall be in phase-aligned relationship with DWORD D[41:0].

The RX_Clk shall be in phase-aligned relationship with DWORD D[41:0].

7.1.2 Speed/Width Conversion

The PHY layer may support 2:1, 4:1, 8:1, 16:1 or 32:1 gearbox conversion ratios, programmable at initialization time. Therefore, on the Logical PHY or Upper Layer side, the resulting data bits width will be 84, 168, 336, 672 and 1344 bits respectively.

All gearbox ratios are optional. Anchor or Chiplet on the link can employ different Gearbox ratios which will not affect interoperability. The latency due to Gearbox function is (typically) one Logical PHY layer clock period, independent of the Gearbox ratio employed.

Table 7-1 illustrates a few example PHY Bitrates, Logical PHY or Upper Layer clock frequency and the resulting Gearbox ratio (R) and the pin count between PHY and the Logical PHY / Upper Layer.

Table 7-1: Example Gearbox Ratio and # of signals between PHY and Logical PHY / Upper Layer.

Gearbox Ratio (R)	OpenHBI-L IO Bitrate [Gbps]	Logical PHY and/or Upper Layer clock	# of Data signals / DW	# of signals to Logical PHY or Upper Layer (= 42*R)
4	8	2GHz	42	168
8	16	2GHz	42	336
16	20	1.25GHz	42	672
16	32	2GHz	42	672

7.1.3 Serialization Order

The OpenHBI-L PHY shall serialize data in contiguous groups starting at the LSB (least significant bit). Table 7-2 below shows an example serialization order of 8:1 gearbox ratio with all 42 data signals D[41:0] are transporting data. The LSB of each data beat shall be transmitted on OpenHBI-L D[0] signal wire.

Table 7-2: Example Serialization Order for 8:1 gearbox ratio

Serialized data beat	Logical PHY layer bits transmitted
1 (first)	Bit [41:0]
2	Bit [83:42]
3	Bit [125:84]
4	Bit [167:126]
5	Bit [209:168]
6	Bit [251:210]
7	Bit [293:252]
8 (last)	Bit [335:294]

7.1.4 Calibration and Training

Typically, the IO calibration and Clock-to-Data training and deskewing are performed in the PHY layer. Refer to Section 12.4 for more details.

7.1.5 PHY Operation

Once enabled and trained, the OpenHBI-L PHY layer shall provide continuous data flow per direction (TX_Clk will toggle continuously).

A Link layer may provide optional flow control and valid data / idle transfer indication as needed, but the PHY layer always sends/receives a word (up to 42 data bits per DW) every clock cycle.

8. Logical PHY Layer

This section describes OpenHBI-L Logical PHY layer and related services to work with PHY layer to form an OpenHBI-L transport for chiplet applications.

A typical bidirectional Logical PHY layer function is programmable at initialization time to operate in either TX or RX mode. Fixed function chiplets or designs with known configurations may implement fixed direction Logical PHY interface units.

Refer to Figure 8-1, this layer defines the following Optional-normative functions:

- Bit Reordering for ROTATED orientation
- DBI (data bus inversion) for power savings
- Framing and alignment
- Parity generation/check

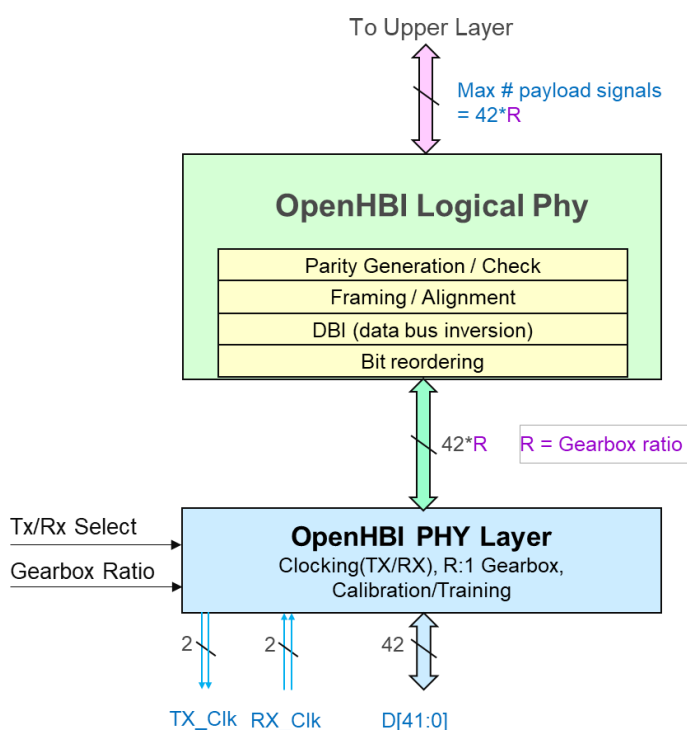


Figure 8-1: OpenHBI-L Logical PHY layer functions

The interface between PHY layer and Logical PHY layer consists of $42 \cdot R$ signals (R =gearbox ratio) plus local clock and optional implementation-specific signals.

Each Logical PHY layer service is optional-normative and can be disabled. When DBI, Framing and Parity functions are enabled, 6 of the 42 data pins of a DWORD are reserved for Logical PHY layer functions, as follows:

- 1 pin for Framing/Alignment (D[41])
- 1 pin for Parity (D[40])
- 4 pins for DBI (D[39:36])
- 36 pins for data payload (D[35:0])

Disabling a service frees up the associated pin(s) and increases the number of signals available to the Upper Layer. For example, disabling DBI feature will provide $4 \times R$ more signals to Upper Layer, where R = Gearbox ratio. The maximum number of data payload signals between Logical PHY layer and Upper Layer is $42 \times R$.

8.1 Bit Reordering

This Bit-reordering function enables straight line routing between two OpenHBI-L chiplets in ROTATED orientation. Refer to Section 6.7 for details.

It is important to note that the Bit Reordering function, if required, must be performed before any other Logical PHY functions can be performed.

8.2 Data Bus Inversion – Enhanced 9-bit DBI mode

OpenHBI-L supports enhanced 9-bit DBI function. DBI is an optional normative service that ensures no more than 50% of the 9 data signals + DBI signal (total 10 signals per DBI signal) will toggle on the DWORD physical wires, which can reduce the energy/bit with the added benefits to reduce the SSO (Simultaneous Switching Output) noise and improve the signal integrity of the C2C interconnect.

If DBI is enabled, 4 pins are dedicated to DBI, each pin is associated with 9 data pins.

- D[36] = DBI0 covers D[8:0]
- D[37] = DBI1 covers D[17:9]
- D[38] = DBI2 covers D[26:18]
- D[39] = DBI3 covers D[35:27]

If DBI is not enabled, the DBI signals can be used as data-carrying signals D[39:36] or those signals can be disabled (or do not toggle) to save power.

Framing/D[41] and Parity/D[40] bits are not covered by DBI functions.

On the TX side, the DBI function counts the number of data transitions within the associated 9-bit word. This involves comparing current bit value to previous bit value. If 5 or more bits change value, then the DBI pin will be asserted ($=1$) and all 9 data bits will be inverted. Otherwise, DBI is not asserted ($=0$) and the data bits are unchanged. This ensure there will always ≤ 4 transitions (out of 9-bit word) in any transfer.

On the RX side, the incoming DBI bit, when asserted, is used to invert the 9 incoming data bits.

When DBI is enabled, the maximum number of transitions per DWORD is $4 \times (4 \text{ (data)} + 1 \text{ (dbi)}) + 2 \text{ (d41, d40)} = 22$, when compared with 42 transitions without DBI.

Note that on the TX side, DBI state shall be reset at initialization time. The first transfer may contain more than 4 transitions, but the subsequent signal toggling counts will generate the correct DBI state accordingly.

8.3 Framing and Alignment

Logical PHY layer may support optional framing and alignment services for Intra-DW Framing (within a DWORD) and Inter-DW Alignment (with optional alignment service between multiple DWORDs by the Upper Layer above the Logical PHY).

When configured, D[41] becomes a dedicated “Framing” signal sent along with the data D[40:0]. The Framing signal shall be active High, 1UI wide pulse concurrent with the data phase of D[40:0] that corresponds to the first beat (least significant word) in a serialized sequence generated by TX gearbox.

The RX Logical PHY is responsible to detect, achieve and maintain proper Framing boundary. Goal is to ensure first “beat” in a burst sequence carries the “Framing bit = 1” signaling.

Alignment Error can be reported via (informative) CCS Status registers. Reporting and handling of Framing errors is out of scope of OpenHBI-L specification. Refer to Section 12.3 CCS Registers.

8.3.1 Intra-DW Framing

On TX side, the TX PHY layer gearbox serializes the wide data word into R data beats (e.g., R=4:1) carried by the TX DWORD data signals.

On RX side, the sampled “Framing” signal is expected to be 0b000....001, where Framing bit is the LSB (least significant bit), and it shall have (R-1) “zeros” trailing the “Framing bit = 1”.

Framing Error should be reported if:

1. If sampled Framing bit value is not as expected, i.e., LSB = 0 → Framing Error
2. If sampled Framing bit has 1 in LSB (least significant bit) but the # of trailing zeros is different from (R-1) → Framing Error

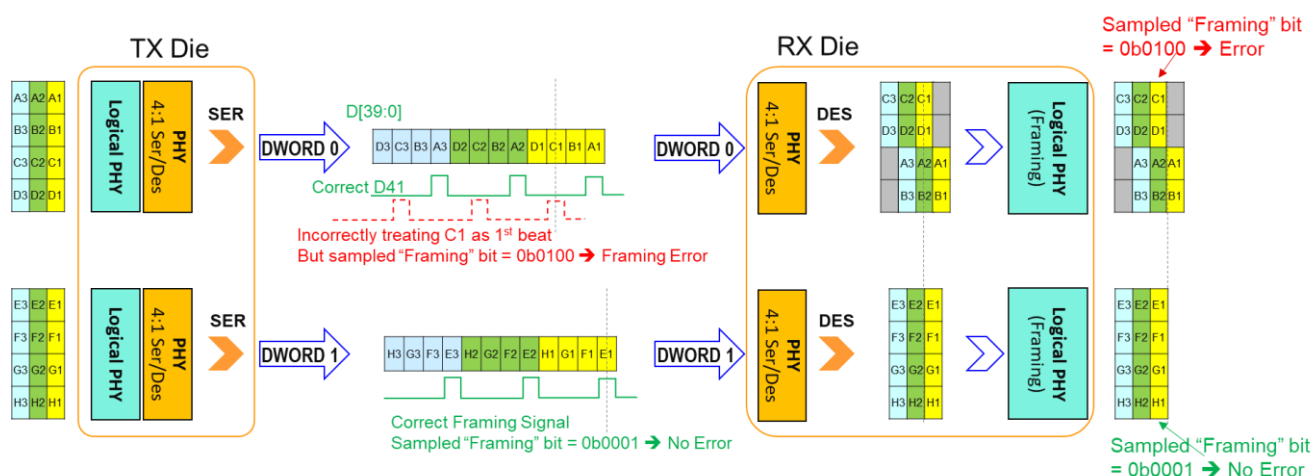


Figure 8-2: Example diagram showing correct and incorrect Intra-DW framing

8.3.2 Inter-DW Framing/Alignment

Figure 8-3 below illustrates the Upper Layer can optionally utilize the “Framing” signal to align or group multiple DWORDs for use cases that require DWORD channel aggregation.

All grouped / aligned DWORD channels must be in same clock domain.

Example shown is two DWORDs (or more) where the skews between DWORD’s Clock/Data/Framing signals are Non-Zero.

Each Logical PHY will perform framing normally and pass the sampled “Framing” bit 0b000.....01 to Upper Layer which can use to align the First Beat of traffic on each DWORD Channel.

Note: This is an Optional feature that belongs to Upper Layer above “Logical PHY” and is outside of OpenHBI-L scope.

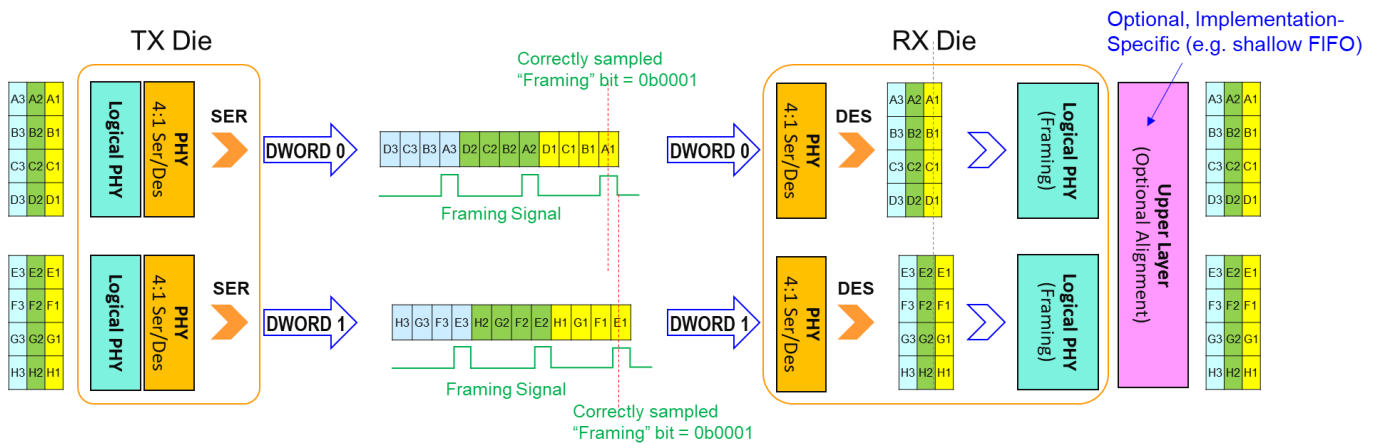


Figure 8-3: Example diagram showing correct Inter-DW framing

If Inter-DW skew is larger than Framing pattern period (e.g. 1ns @1GHz LogPHY clock), the Upper Layer can align two (or more) DWORDs incorrectly as illustrated in the Figure 8-4 below.

This can be solved by optional programmable extended range of repeating pattern of “one followed by (N-1) zeros” where N = 32, 64, 128.

- where “N” can be determined by max Inter-DW skews to ensure no ambiguity in Inter-DW Alignment boundary`

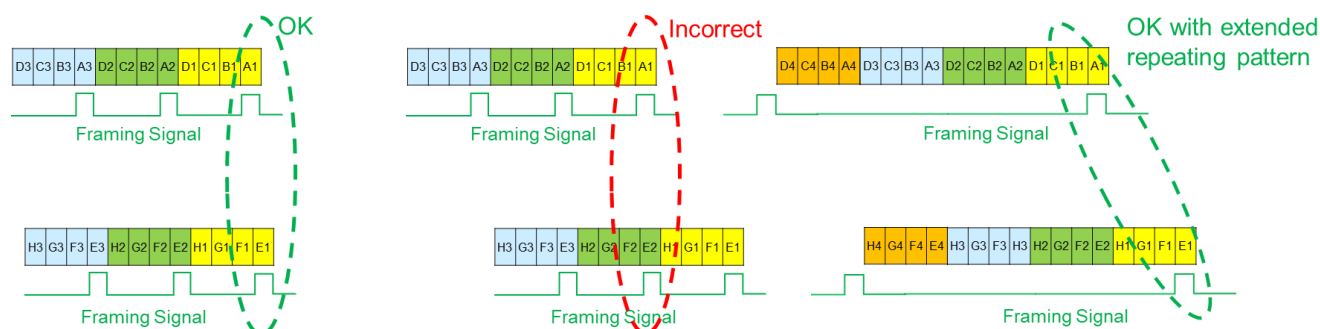


Figure 8-4: Example diagram showing how extended repeating pattern can handle large Inter-DW skew

The key benefit is to accommodate Inter-DW (DW-to-DW) skews. A small benefit is less toggling and slightly less power on “Framing” (D[41]) bit.

Note: This is an Optional feature that belongs to Upper Layer above “Logical PHY” and is outside of OpenHBI-L scope.

8.4 Parity

OpenHBI-L Logical PHY layer provides EVEN parity protection. If enabled, D[41, 39:0] is protected with 1 parity bit (D[40]) which shall be aligned with the data.

On the TX side, parity is computed on all 41 bits after DBI. On the RX side, parity is computed on all 41 bits before DBI. RX side then performs parity checking and error logging on the receive (RX) side. The error(s) will be logged via error log / status registers.

OpenHBI-L Parity error reporting and handling are implementation-specific. If an OpenHBI-L chiplet supports Parity error reporting and notification and the feature is enabled, when the Chiplet RX DWORD detects a parity error, it shall generate an I3C IBI (In-Band Interrupt) with MDB to the Anchor, and the firmware / interrupt handler will check the interrupt source through I3C and handles the parity error accordingly.

The Logical PHY layer shall support the following modes

- Parity is enabled
- Parity is disabled, and the parity bit is inactive
- Parity is disabled, and the parity bit carries user/sideband data (not covered by DBI)

8.5 Options and Modes

Of the Logical PHY layer services, the following services consume interface bits and therefore affect the user-available data payload bus width

- Framing
- Parity
- DBI

These services are optional and can be disabled, freeing up additional useable payload bits. However, not all disabled service combinations (modes) are supported. Only the following Modes 0-4 in Table 8-1 are supported.

Table 8-1: Logical PHY layer modes

Mode	Framing	Parity	DBI	Payload bits
0 (default)	Enabled	Enabled	Enabled	36
1	-	-	Enabled	38
2	Enabled	Enabled	-	40
3	Enabled	-	-	41
4 (Logical PHY Bypass)	-	-	-	42

Mode 0 is the default mode in which all services are enabled. Mode 4 has all services disabled except for the optional Bit-reordering function can be controlled via CCS in Logical PHY layer bypass mode.

Note: Bit-reordering function can be independently enabled or disabled for all Mode 0-4. It does not affect the number of bits available to the Upper Layer.

8.6 Interface Pin Assignments per Mode

Depending on the mode, the pin assignment visible to the Protocol layer will be different, as illustrated in the table below. The Logical PHY layer is responsible for multiplexing the bits as needed to present this simple interface to the protocol layer.

Table 8-2: Pin assignment per mode

Mode	Gearbox Ratio				
	2:1	4:1	8:1	16:1	32:1
0 (Frm+Par+DBI)	Bit [71:0]	Bit [143:0]	Bit [287:0]	Bit [575:0]	Bit [1151:0]
1 (DBI)	Bit [75:0]	Bit [151:0]	Bit [303:0]	Bit [607:0]	Bit [1215:0]
2 (Frm+Par)	Bit [79:0]	Bit [159:0]	Bit [319:0]	Bit [639:0]	Bit [1279:0]
3 (Frm)	Bit [81:0]	Bit [163:0]	Bit [327:0]	Bit [655:0]	Bit [1311:0]
4 (Bypass)	Bit [83:0]	Bit [167:0]	Bit [335:0]	Bit [671:0]	Bit [1343:0]

9. Chiplet Configuration interface and Sidebands

9.1 MIPI I3C as Configuration Interface

OpenHBI-L specification adopts MIPI I3C v1.1.1 (or future backward-compatible version) as the default configuration interface and shall support HDR-TSP mode up to 33Mbps (effective).

The I3C interface shall support 1.2V operating voltage. It shall support I3C IBI (In-Band Interrupt) with a Mandatory Data Byte (MDB).

Implementation Note: Anchor or Chiplet is allowed to support I3C Basic, which is interoperable with I3C but at a lower data rate. If MIPI I3C Basic is used, it shall support SDR mode up to 12.5Mbps.

During initialization, the configuration firmware/software shall switch I2C FM+ mode (1Mbit/s) to the highest data rate mode commonly supported between the Anchor and Chiplet as soon as possible before performing the chiplet configuration.

OpenHBI-L specification does not define a common I3C configuration command set. The Anchor and Chiplet I3C controllers shall support Direct Read/Write, Direct Write and Direct Read CCC.

9.1.1 I3C In-Band Interrupts (IBIs)

To enhance interoperability between vendors, OpenHBI-L devices shall support I3C IBI with Mandatory Data Byte (MDB).

OpenHBI-L IBIs use the interrupt group identifier for “Non MIPI Reserved” (MDB[7:5] = 3'b011).

Vendor-specific IBIs use interrupt group identifier for “User Defined” (MDB[7:5] = 3'b000).

Table 9-1: I3C In-Band Interrupts (IBIs)

Interrupt Group Identifier (MDB[7:5])	Specific Interrupt ID (MDB[4:0])	Interrupt Name	Interrupt Description
3'b000	5'h00 – 5'h1F	RSVDV	Reserved for vendor specific.
3b001 – 3'b010	5'h00 – 5'h1F	MIPI	Reserved for MIPI.
3'b011	5'h00	INITDONE	Indicates that PHY initialization completed without errors.
	5'h01	INITDONERR	Indicates that PHY initialization completed with errors.
	5'h02	TRNDONE	Indicates that training completed without errors.
	5'h03	TRNDONERR	Indicates that training completed with errors.
	5'h04	TXRQST	Request TX to transmit data. The length of the transmitted data is controlled by burst length register (BLR).
	5'h05	RXDERR	Indicates there was a parity error on the received data.
	5'h06 – 5'h1F	RSVDO	Reserved for OpenHBI.
3'b100	5'h00 – 5'h1F	RSVDV	Reserved for vendor-specific timing information.
3'b101 – 3'b111	5'h00 – 5'h1F	MIPI	Reserved for MIPI.

9.2 OpenHBI-L Sideband Signals

Table 9-2 shows the OpenHBI-L defined (normative) sideband signals for configuration and test. All OpenHBI-L Sideband signals shall be nominal 1.2V IO.

Anchor can use “PwrGood = high” to indicate to the Chiplet that power is good and can also use it as power-on / cold reset signal. Both PwrGood and Reset can functionally reset the OpenHBI-L interface and the Chiplet-specific functions. One primary difference between the two signals is that certain sticky registers/bits or certain status (e.g., security or persistent memory status) etc., may persist through assertion of “Reset” but cannot persist when PwrGood is de-asserted.

OpenHBI-L adopts I3C for configuration and management as described in Section 9.1. The JTAG interface and the I3C interface can be used during wafer testing of chiplets and post-packaged boundary scan. JTAG_TRST is optional normative signals.

Table 9-2: OpenHBI-L defined signals [Normative] for Chiplet configuration and test

Signal Descriptions	Direction		Notes
	Anchor	Chiplet	
PwrGood	Out	In	High = PwrGood
Reset	Out	In	High = Reset
RefClk (default)	Out	In	
I3C_CLK	Out	In	
I3C_DATA	I/O	I/O	
JTAG_TCK	Out	In	
JTAG_TMS	Out	In	
JTAG_TDI	Out	In	
JTAG_TDO	In	Out	
JTAG_TRST	Out	In	Optional normative signal
Total = 10			

OpenHBI-L does not define the physical or relative locations of the Sideband signals to the DWORD channels.

9.3 LFSR circuits for Testing and Training patterns

A 40-bit Galois LFSR (linear feedback shift register) circuit is defined for OpenHBI-L DWORD for testing and training of the die-to-die signal interface.

Each DWORD shall support four 40-bit Galois LFSR logic (or equivalent) as defined below.

$$f(x) = X^{40} + X^{38} + X^{21} + X^{19} + 1$$

Figure 9-1 illustrate (part of) the implementation of each 40-bit Galois LFSR. The L[39:0] are the 40 output bits of the circuit. For clarity, the circuit to initialize the seed (initial value) of the LFSR is not shown.

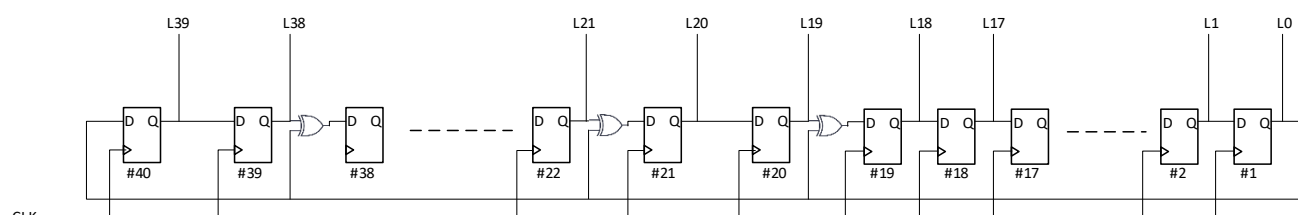


Figure 9-1: Example implementation of 40-bit Galois LFSR circuits

TX and RX LFSRs shall be programmable to any 40-bit value as initial value (seed).

In LFSR mode, both TX and RX LFSRs shall be reset to default seed value of 0xAAAAAAAAAA. Optionally, other seed values can be used if both TX and RX LFSRs are programmed to the same seed value accordingly.

In Programmable Pattern mode, the TX and RX LFSRs can be initialized to any 40-bit value but the TX and RX LFSRs do not advance its state with TXClk and RXClk. This enables a repeating pattern of any 40-bit value to be continuously transmitted which can be useful for some training algorithms. Refer to Section 12.4 and Figure 12-5 for more details.

Refer to Figure 9-2, RX DWORD shall support “LFSR Compare mode” to detect if incoming bits from TX DWORD matches the output of the local matching LFSR inside RX DWORD. Design should have error registers to capture any mismatch. It is optional to capture additional details of the mismatches such as error counts, errors per lane and/or other statistics to assist in testing and system verification. The error registers can optionally be sticky that can persist through assertion of “Reset” but cannot persist when PwrGood is de-asserted.

OpenHBI-L chiplet can optionally support on-die BERT to help with BER performance evaluation and is implementation-specific.

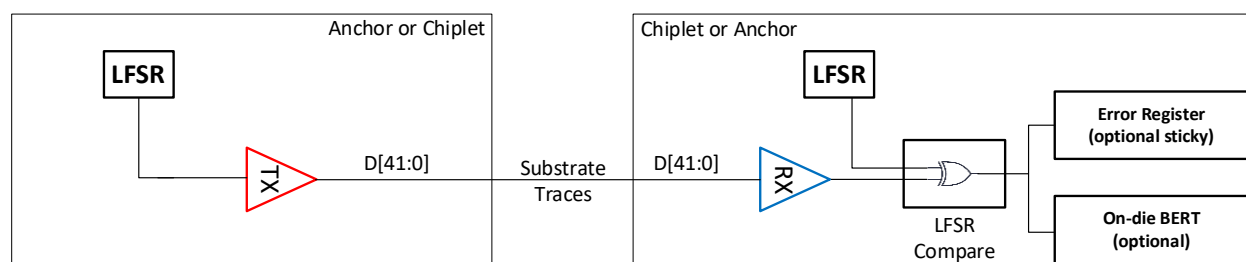


Figure 9-2: Example LFSR Compare with Error register and Optional BERT

9.4 OpenHBI-L Training and Testing

This section describes the testing, calibration and training steps of Anchor TX DWORDs to Chiplet RX DWORDs. Similar sequence can be used for testing, calibration and training steps of Chiplet TX DWORDs to Anchor RX DWORDs.

On the Anchor side, the configuration steps below are performed locally using proprietary mechanism.

On the Chiplet side, the configuration steps below are performed by firmware or system software through the I3C configuration interface.

For the sake of clarity, the sequence describes one TX DWORD to RX DWORD only. It is strongly recommended all DWORDs can be configured to be trained or tested in parallel to reduce initialization time, but it is allowed to perform each DWORD or groups of DWORDs sequentially.

9.4.1 Data-to-RX_Clk Training Steps

Below describe the Data-to-RX_Clk training steps to train all data lanes of the OpenHBI-L DWORD to operate at the target data rate and the target BER.

1. Configures Anchor TX DWORD to stop the TX_Clk and reset the TX LFSR
2. Configures Chiplet RX DWORD to reset RX LFSR
3. Enables the RX “LFSR Compare” circuits.
4. Preset both TX- and RX-side LFSR to seed value of 0xAAAAAAAAA
5. Configures the number of TX LFSR patterns to be transmitted, and RX DWORD expected to receive.
6. Enables TX LFSR to transmit the programmed number of LFSR patterns OR until software configures TX LFSR to stop transmission.
 - RX-side detects LFSR transmission started by first rising edge of RX_Clk.
7. RX-side performs search for optimal EQ setting and training (implementation-specific algorithms).
8. After the DWORD training process is complete (either successful or fail or the programmed number of TX LFSR patterns has been sent):
 - If training is successful, RX will set RX-TTCR “RX initialization done” bit = 1 and “RX training error” bit = 0
 - If training is unsuccessful, RX will set RX-TTCR “RX initialization done” bit = 1 and “RX training error” bit = 1. Software can adjust the parameters and retry the training steps above.
9. When the training of all RX DWORDs on a chiplet is done, the chiplet may send an IBI with MDB with “TRNDONE” or “TRNDONERR” Interrupt ID to the Anchor to notify the training has been completed.

9.4.2 C2C Interface Testing Steps

Below describe the C2C interface testing steps conducted at target data rate after training has been completed.

1. Configures Anchor TX DWORD to stop the TX_Clk and reset the TX LFSR
2. Configures Chiplet RX DWORD to reset RX LFSR
3. Clear the LFSR Compare error registers and the optional on-die BERT (Bit Error Rate Tester) logic.
4. Enables the RX “LFSR Compare” circuits.
5. Preset both TX- and RX-side LFSR to seed value of 0xAAAAAAAAA
6. Configures the number of TX LFSR patterns to be transmitted, and RX DWORD expected to receive.
7. Enables TX LFSR to transmit the programmed number of LFSR patterns OR until software configures TX LFSR to stop transmission and RX to stop comparing LFSR patterns and logging errors.
 - RX-side detects LFSR transmission started by first rising edge of RX_Clk.
 - RX-side pattern comparators (XOR) detect and log any mismatch between the incoming data and the patterns generated by the local RX DWORD LFSRs into the error registers (which can optionally be sticky and is implementation choice).
8. After TX LFSR transmissions stopped, system software can examine any errors in the error registers and the optional BERT.

9.4.3 40-bit Galois LFSR mapping D[41:0]

Conceptually, each DWORD contains four “Bytes” or Byte[3:0]. In the context of OpenHBI, the definition of a “Byte” is different from the conventional 8-bit data size. Each Byte[x] within a DWORD contains 10 Data signals (where x = 3..0).

Each Byte[x] implements the 40-bit LFSR circuit. Each DWORD requires four 40-bit LFSR circuits. The LFSR circuits of each Byte works independently and simultaneously across the Bytes.

When enabled, the 40-bit output, L[39:0], of the LFSR are shifted out serially over 4 UI (2 rising and 2 falling edges), depicted as UI[0] to UI[3] in Table 9-3 below on each of the 10 signals of the respective Byte[x], and then continue with the next LFSR state's pattern.

For example:

- D0 of Byte[0] will carry the L0, L10, L20 and L30 output of the 40-bit LFSR circuits of Byte[0] in UI[0], UI[1], UI[2] and UI[3] respectively.
- D18 of Byte[1] will carry the L7, L17, L27 and L37 output of the 40-bit LFSR circuits of Byte[1] in UI[0], UI[1], UI[2] and UI[3] respectively.

The LFSR has 40 bits output while there are 42 data signals in the DWORD. Due to the special mapping convention to [Reference 1], the D5 and D36 are treated differently in the mapping.

D5 and D36 also require a deterministic pattern for testing and training. For simplicity, OpenHBI-L specify D5 and D36 to use of LFSR bit pattern of its neighbor C4 bump, D4 and D37 respectively while maintaining rotation symmetry as shown in Table 9-3 such that:

- D5 duplicates the LFSR bit pattern of D4
- D36 duplicates the LFSR bit pattern of D37

Table 9-3: 40-bit LFSR output L[39:0] mapping to D[41:0] within a DWORD over UI[0] to UI[3]

Byte[3:0] of DWORD				UI [3:0] of respective Byte[3:0]			
Byte 0	Byte 1	Byte 2	Byte 3	UI 0	UI 1	UI 2	UI 3
D0	D11	D21	D31	L0	L10	L20	L30
D1	D12	D22	D32	L1	L11	L21	L31
D2	D13	D23	D33	L2	L12	L22	L32
D3	D14	D24	D34	L3	L13	L23	L33
D4, D5	D15	D25	D35	L4	L14	L24	L34
D6	D16	D26	D36, D37	L5	L15	L25	L35
D7	D17	D27	D38	L6	L16	L26	L36
D8	D18	D28	D39	L7	L17	L27	L37
D9	D19	D29	D40	L8	L18	L28	L38
D10	D20	D30	D41	L9	L19	L29	L39

It is recommended each DWORD contains its own independent 40-bit LFSR, LFSR Compare and related circuits to enable parallelized training and test operations. But it is implementation-choice if each DWORD contains its own four LFSRs or the four LFSRs are shared between multiple DWORDs to reduce circuit area. The latter will cause the longer training, calibration, and test time.

As defined, the LFSR will generate 40 bits per clock and run at a frequency that is 1/4th of the data rate. For example, the LFSR runs at 2GHz for data rate at 8Gbps. However, design running at higher C2C data rate can employ additional logic to implement the LFSR to generate the equivalent of 80 bits or 160 or 320 bits per clock so the LFSR will be running at 1/8th or 1/16th or 1/32nd of the data rate respectively. Actual designs of such LFSR is implementation-specific and is out of scope of OpenHBI-L specification.

Figure 9-3 illustrated an example implementation of the 40-bit Galois LFSR that can generate 80-bit per clock. The design is equivalent to advancing the LFSR with the same polynomial defined in Section 9.3 by 2 states per clock (instead of 1 state per clock). For clarity, the circuit to initialize the seed (initial value) of the LFSR and the clock connections of the flops are not shown.

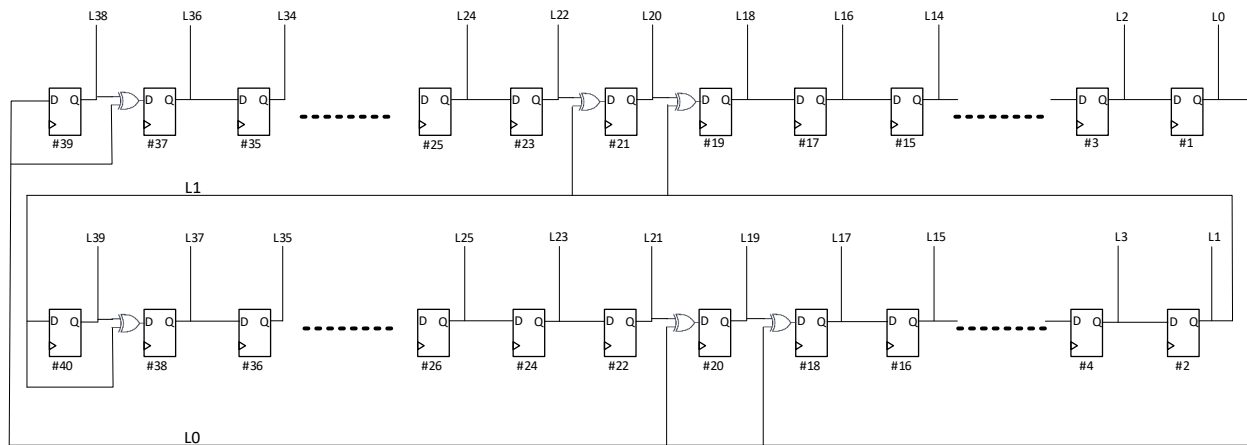


Figure 9-3: Example LFSR implementation that can generate 80-bit per clock.

The lower 40-bit LFSR[39:0] is taken directly from the L[39:0] output of the LFSR circuit. The upper 40-bit LFSR[79:40] is derived from the lower 40-bit with 3 additional XOR gates as described below:

$$\text{LFSR [39:0]} = \{ L[39], L[38], \dots, L[1], L[0] \}$$

$$\text{LFSR [57:40]} = \{ L[18], \dots, L[1] \}$$

$$\text{LFSR [58]} = L[19] \text{ XOR } L[0]$$

$$\text{LFSR [59]} = L[20]$$

$$\text{LFSR [60]} = L[21] \text{ XOR } L[0]$$

$$\text{LFSR [76:61]} = \{ L[37], \dots, L[22] \}$$

$$\text{LFSR [77]} = L[38] \text{ XOR } L[0]$$

$$\text{LFSR [79:78]} = \{ L[0], L[39] \}$$

10. OpenHBI-L Routability and Reference Stack-Up

10.1 Bump Pitch

OpenHBI-L specifies the minimum bump pitch (min) to range from 110um to 130um for manufacturability considerations. In Figure 10-1 example below, the X and Y dimensions do not form equilateral triangles which provides some area optimization. Attention should be taken to understand the substrate design rules available.

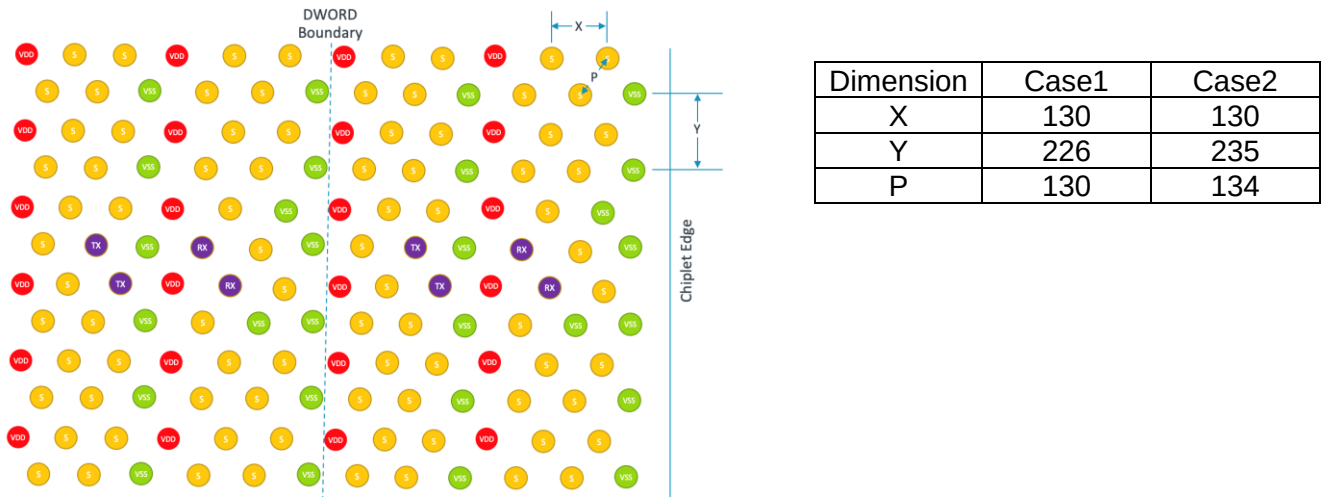


Figure 10-1: Bump pattern for two DWORDs of depth from the die edge

The bump map X-pitch ($\geq$ minimum bump pitch) can be implementation-specific and typically does not affect interoperability and routability, but different X-pitches between the Anchor and chiplet will cause minor differences in trace lengths which requires more careful routing to match the trace length.

10.2 Routing Strategy Examples

In order to achieve the desired isolation, it is recommended that each signal routing layer be in stripline transmission line mode. This includes having a VSS plane above and below each routing layer. The VDD is routed inward, away from the die edge and can be connected below the last VSS layer used for the stripline reference layer. It is important to be mindful of maintaining the return paths for each signal and supply in the adjacent layers for each signal.

There are many routing strategies available. The following sections illustrate two DWORDs of depth routed in a 6-2-6 laminate (3 routing layers) or an 8-2-8 laminate (4 routing layers). These are intended to be reference examples only. Each of these can achieve a very close route length for all signals within a DWORD.

For simplicity, the colors used will represent the following net types.



10.2.1 6-2-6 Routing Example

A 6-2-6 is partially represented below. The die-to-die connectivity occurs in the upper layers so the lower layers are not shown other than to ensure that the VDD plane on the lower side is present to maintain the contiguous return path. VDD and VSS will need to be connected to balls in the lower layers which are not explicitly shown in the routing examples.

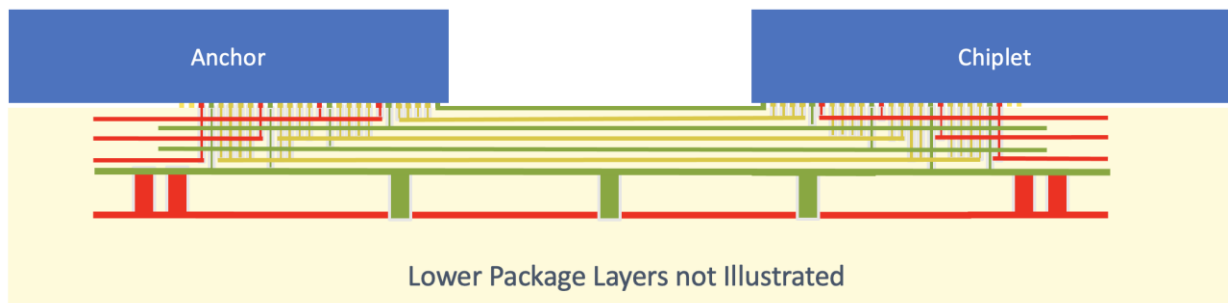
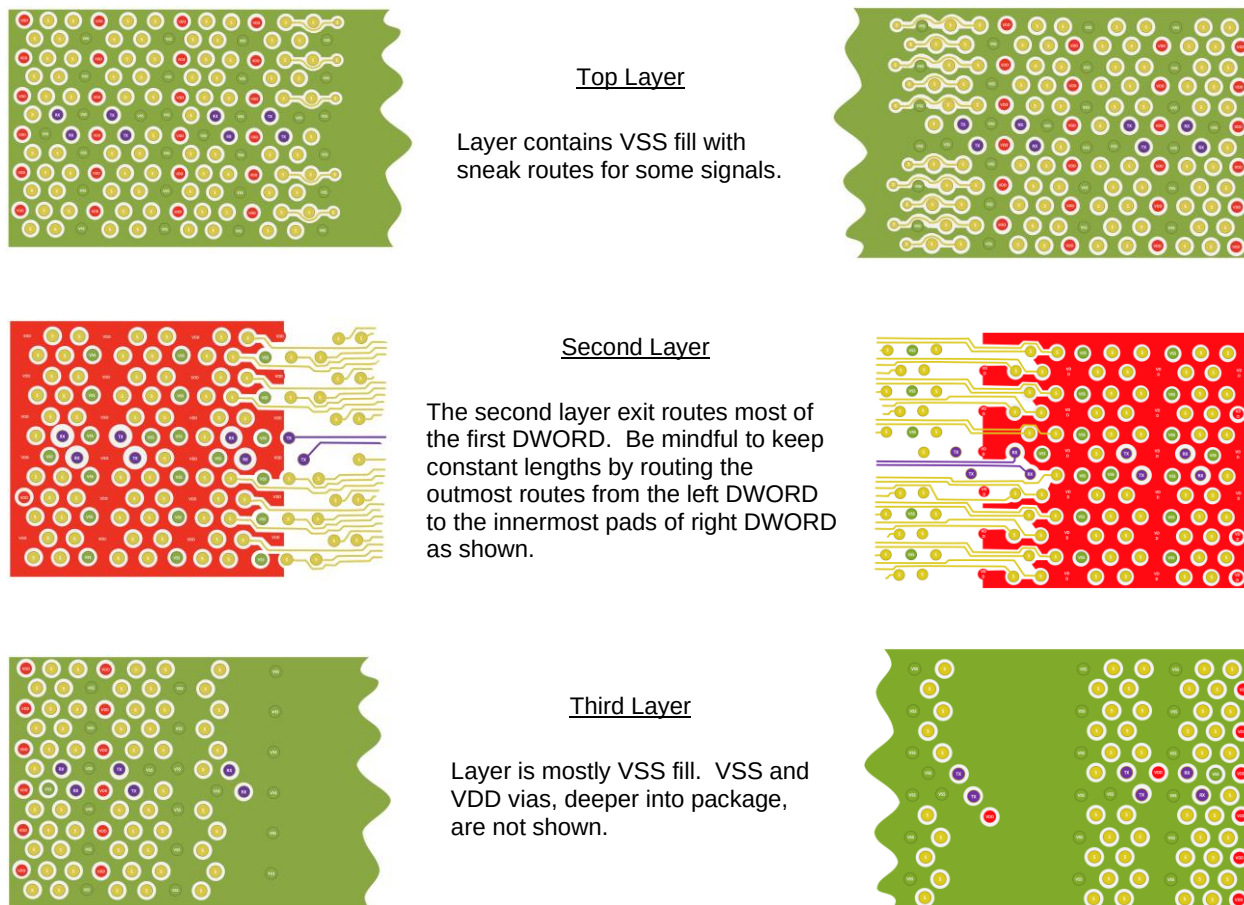
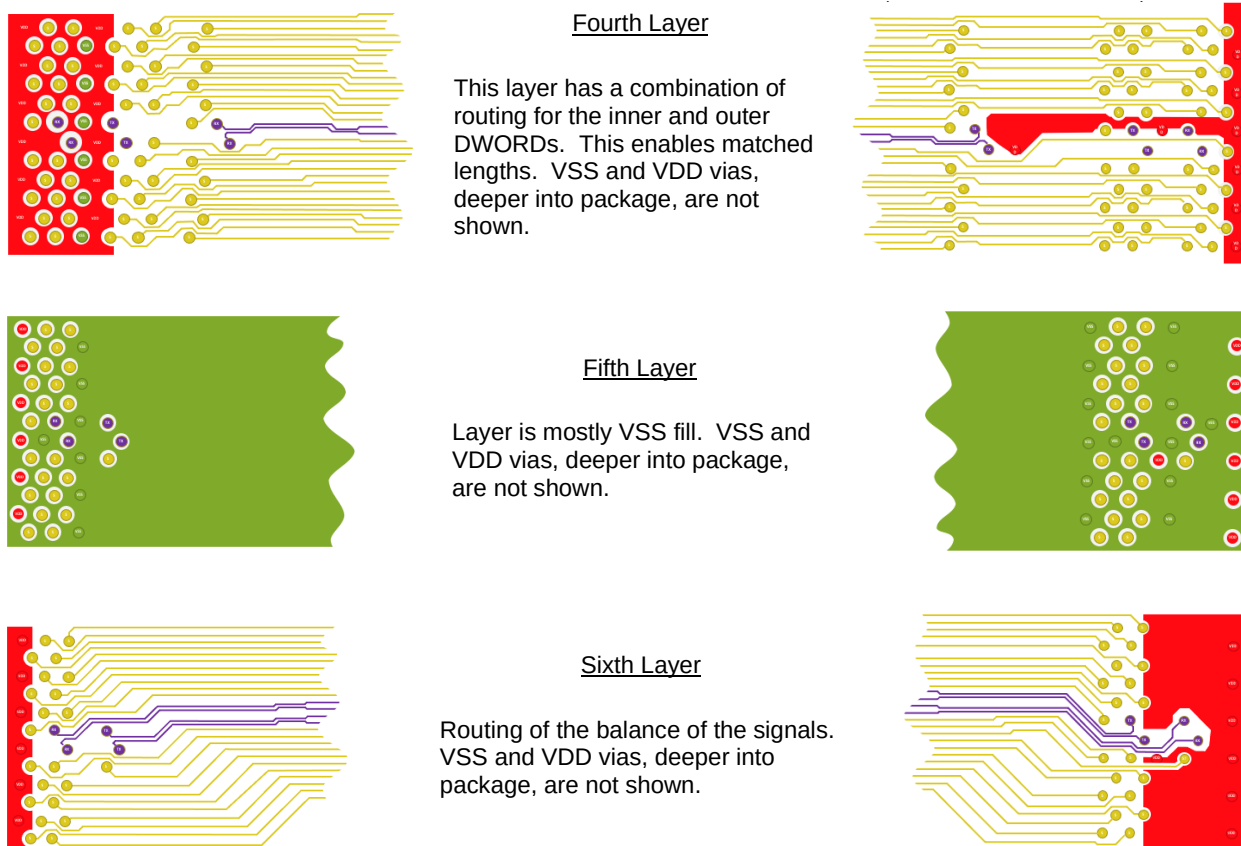


Figure 10-2: Upper layer routing assignments for a 6-2-6 stack-up

The following examples show the proposed routing strategy that allows for length matching within a DWORD. Only the upper routing layers are illustrated. The images on the left and right represent the exits routes for the two chiplets, or the chiplet and anchor.





The subsequent 7th layer is a VSS fill. As mentioned earlier, it is important to provide a contiguous VDD plane below the 7th layer for return path currents. Both the VDD and VSS nets need to also be connected to balls or pads below the routing area.

10.2.2 8-2-8 Routing Example

A 8-2-8 is partially represented below. The die-to-die connectivity occurs in the upper layers so the lower layers are not shown other than to ensure that the VDD plan on the lower side is present to maintain the contiguous return path. VDD and VSS will need to be connected to balls in the lower layers which is not explicitly shown in the routing examples.

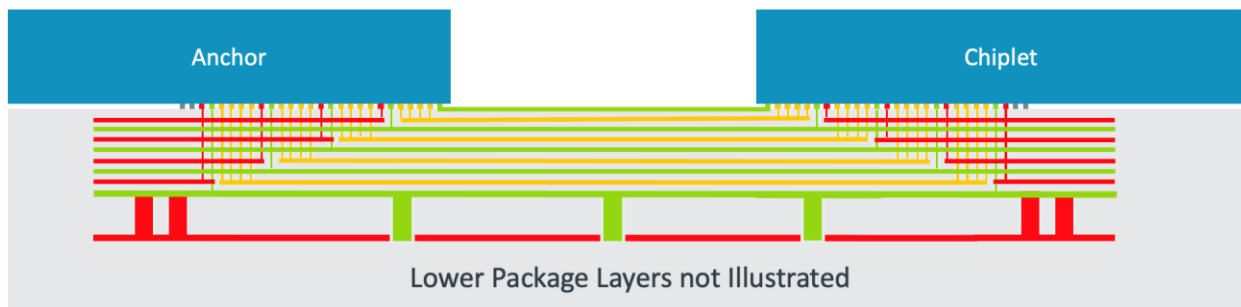
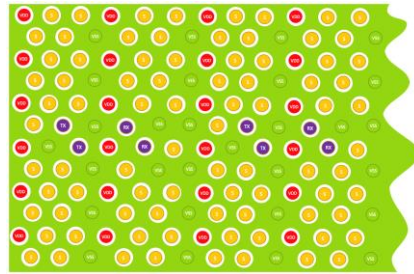


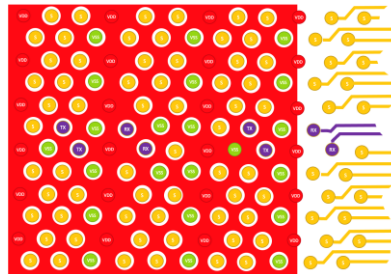
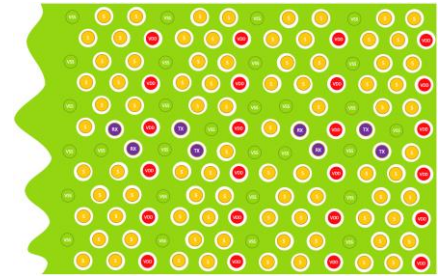
Figure 10-3: Upper layer routing assignments for a 8-2-8 stack-up

The following examples show the proposed routing strategy that allows for length matching within a DWORD. Only the upper routing layers are illustrated. The images on the left and right represent the exits routes for the two chiplets, or the chiplet and anchor.



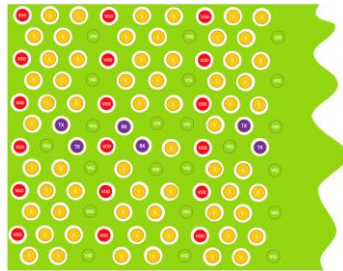
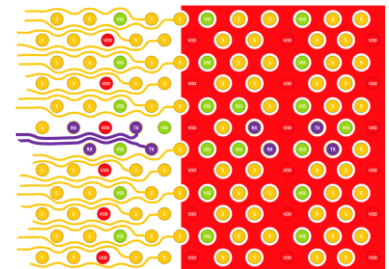
Top Layer

Layer contains VSS fill with sneak routes for some signals.



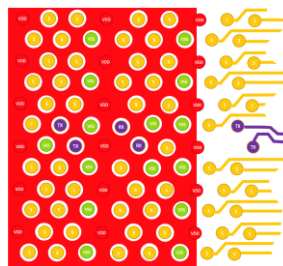
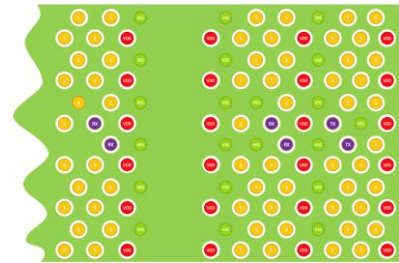
Second Layer

The second layer exit routes half of the first DWORD. The routes are relatively well aligned and straight. VSS and VDD vias, deeper into package, are not shown.



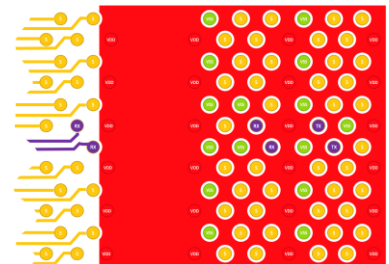
Third Layer

Layer is mostly VSS fill. VSS and VDD vias, deeper into package, are not shown.



Fourth Layer

This layer has a combination of routing for the inner and outer DWORDs. This enables matched lengths. VSS and VDD vias, deeper into package, are not shown.



The subsequent layers are a close repetition of the prior layers. For example, layer 5 is similar to the top layer. Layer 6 is similar to layer 2. Layer 7 is similar to layer 3, and so on.

11. OpenHBI-L IO Electricals

OpenHBI-L is optimized for higher data rates and energy efficiency for organic laminate substrate, while keeping the upper layer mapping identical to OpenHBI v1.0.

Figure 11-1 illustrates the OpenHBI-L Transceiver Topology concept.

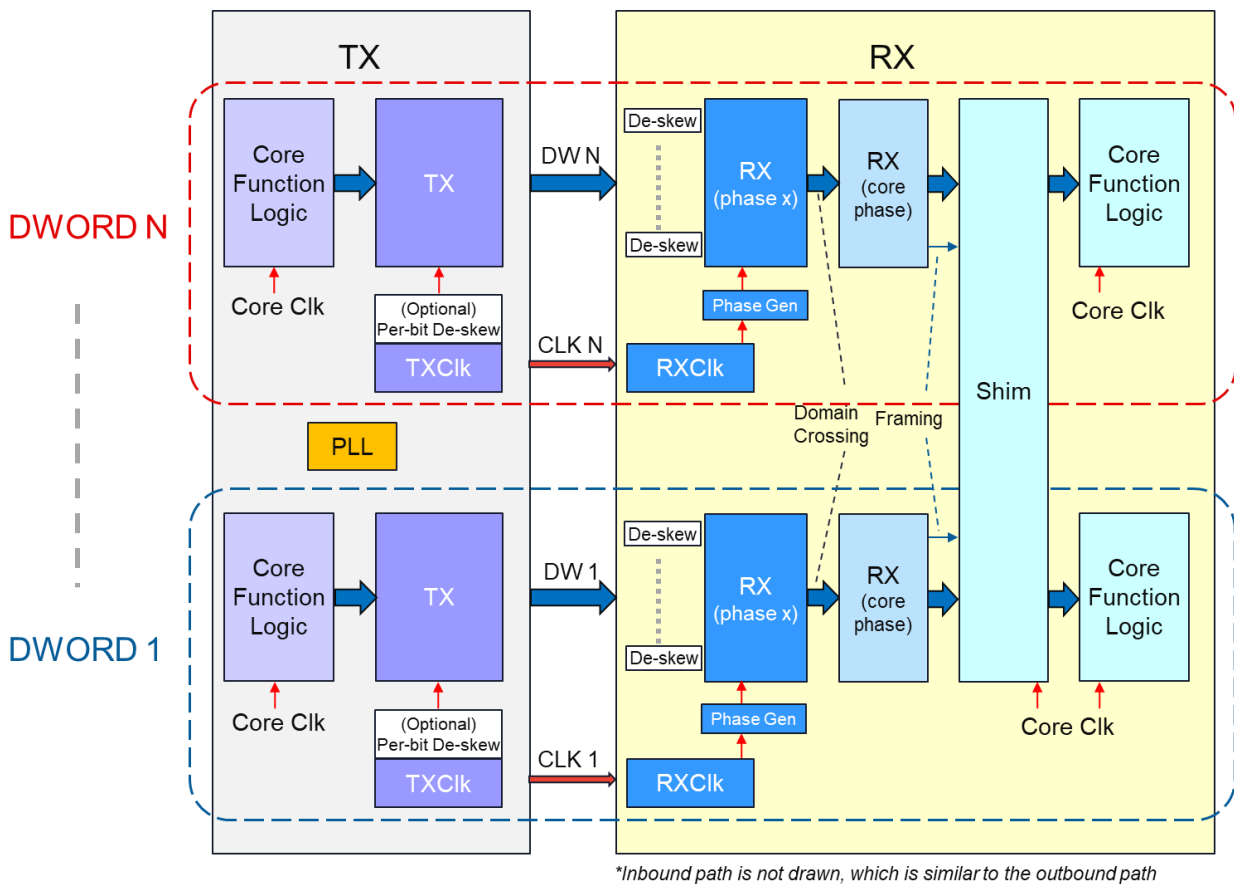


Figure 11-1: OpenHBI-L Transceiver Topology concept

The overall specs are divided into 2 categories: **Normative** and **Reference**, which are color-coded accordingly in the tables above.

Normative portions of IO electrical specifications are hard compliance to meet as standards.

Reference portions of IO electrical specifications are recommended as a reference design only, yet doesn't require to meet compliance, depending on the design implementation and trade-off.

11.1 Transmitter Electrical Parameters

Standard Profile: (4Gbps/8Gbps/16Gbps)

Table 11-1: Transmitter Electrical Parameters (Standard)

Definition	Min	Nominal	Max	Unit
IO Supply Voltage	0.75 – 5%	0.75	0.75 + 5%	V
Clock DCD (Duty Cycle Distortion)	-2.5%		2.5%	
Driver Impedance	36	40	44	Ohm
Clock-Data alignment (clock to data skew) ²	0		0.5	UI
BER (Bit Error Rate)			1e-25	
Data Lane-to-Lane Skew	-10		10	ps
TX timing total budget ¹			0.25	UI

Enhanced Profile: (20Gbps/32Gbps/40Gbps)

Table 11-2: Transmitter Electrical Parameters (Enhanced)

Definition	Min	Nominal	Max	Unit
IO Supply Voltage	0.75 – 5%	0.75	0.75 + 5%	V
Clock DCD (Duty Cycle Distortion)	-2.5%		2.5%	
Driver Impedance	36	40	44	Ohm
Clock-Data alignment (clock to data skew) ²	0		0.5	UI
BER (Bit Error Rate)			1e-15	
Data Lane-to-Lane Skew	-10		10	ps
De-emphasis ³	0	3		dB
TX timing total budget ¹			0.2	UI

Spec details:

Note 1: The major components of the TX timing budget include data/clock skew(after training), TX data-clock relative jitter effect and TX DCD (duty cycle distortion after correction) effect, etc. Please note that the random jitter of the PLL clocking is accounted for on the RX side. The spec aims to grant maximum design flexibility across the individual components listed, while assuring the link's interoperability.

Note 2: Clock is expected to transmit earlier than data with up to 0.5UI differences as shown in the table above. Clock shall not be transmitted later than data.

Note 3: The de-emphasis targets 3dB as nominal spec with 0dB as minimum setting (no TX EQ). No max value is specified.

11.2 Receiver Electrical Parameters

Standard Profile: (4Gbps/8Gbps/16Gbps)

Table 11-3: Receiver Electrical Parameters (Standard)

Definition	Min	Nominal	Max	Unit
RX Termination	45	50	55	Ohm
Single Ended Eye Height (peak-to-peak)	50			mV
BER (Bit Error Rate)			1e-25	
Clock DCD (Duty Cycle Distortion)	-2.5%		2.5%	
RX horizontal total budget ²			0.4	UI
MAX PLL random jitter (1 sigma) ¹			0.4	ps

Enhanced Profile: (20Gbps/32Gbps/40Gbps)

Table 11-4: Receiver Electrical Parameters (Enhanced)

Definition	Min	Nominal	Max	Unit
RX Termination	45	50	55	Ohm
Single Ended Eye Height (peak-to-peak)	50			mV
BER (Bit Error Rate)			1e-15	
Clock DCD (Duty Cycle Distortion)	-2.5%		2.5%	
RX horizontal total budget ²			0.35	UI
Equalization ³	0	3		dB
MAX PLL random jitter (1 sigma) ¹			0.4	ps

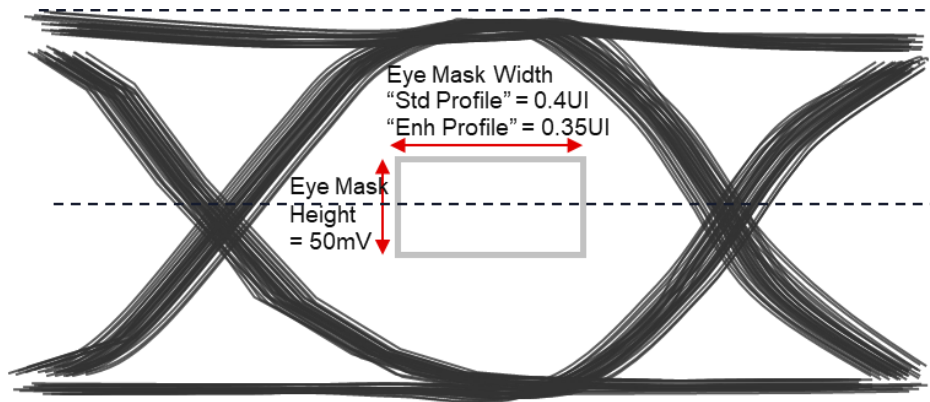


Figure 11-2: Receiver Eye Diagram

Spec details:

Equalization and deskew techniques are expected to be applied for high speed operation to recover and maximize the link margin. They can be applied to TX or RX or both sides, maximizing the design flexibility, as long as all the specifications are met.

Note 1: One of the key parameters defined is the max PLL random jitter. As specified in the TX section, it is captured and defined in the RX section and with up to 10UI only due to the nature of the source synchronization and the extra short reach/skew of the OpenHBI spec. Any jitter slower than 10UI from PLL is expected to be fully canceled out by design.

Note 2: The major components of the RX timing budget include data/clock skew(after training), RX data-clock relative jitter effect, random clocking variation, RX aperture, RX data-clock residue drift (after necessary continuous training/adaptation). The spec aims to grant maximum design flexibility across the individual components listed, while guaranteeing the link's interoperability.

Note 3: The equalization targets 3dB as nominal spec with 0dB as minimum setting (no RX EQ). No max value is specified.

11.3 Channel Parameters

Standard Profile: (4Gbps/8Gbps/16Gbps)

Table 11-5: Channel Parameters (Standard)

Definition	MAX	Unit
Channel (Crosstalk + ISI + Slew Rate) ²	0.35	UI
Pin cap (TX side) ¹	250	fF
Pin cap (RX side) ¹	250	fF

Enhanced Profile: (20Gbps/32Gbps/40Gbps)

Table 11-6: Channel Parameters (Enhanced)

Definition	MAX	Unit
Channel (Crosstalk + ISI + Slew Rate) ²	0.45	UI
Pin cap (TX side) ¹	150	fF
Pin cap (RX side) ¹	150	fF

Spec details:

The channel is characterized in the transient domain in the spec to decouple it from a specific channel /packaging medium.

Note 1: The pin cap refers to the equivalent cap value at Nyquist rate. It is expected to use inductor peaking (e.g., T-coil) or similar technology to achieve the spec showcased in the enhanced profile.

Note 2: The major components of the channel timing budget include ISI (inter-symbol-interference), crosstalk effect and slew rate effect (degradation introduced around the top/bottom of RX eye mask) The spec aims to grant maximum design flexibility across the individual components listed, while guaranteeing the link's interoperability.

11.3.1 Channel Insertion loss and Crosstalk Limit diagrams

The parameters in this sub-section is informative and is to serve as a reference/baseline for design.

The channel reach is targeting up to 20mm reach.

The channel should typically have less loss than Figure 11-3 below (hence being above the green line).

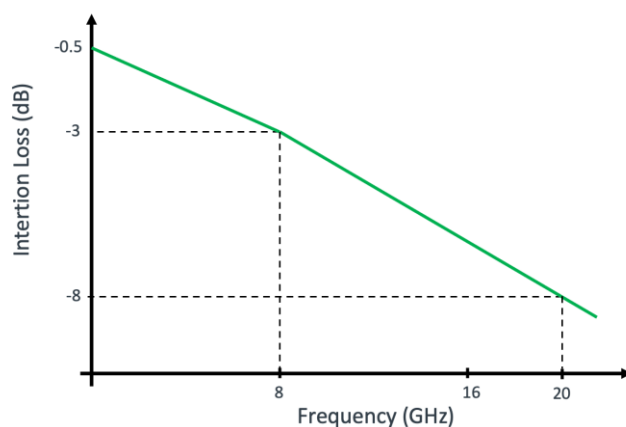


Figure 11-3: Channel Insertion Loss Limit

Similarly, the crosstalk (power-sum far-end crosstalk) should be lower than in the picture below (hence below the green line).)

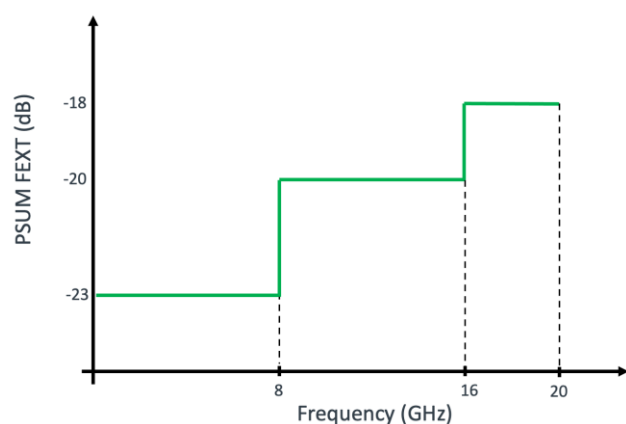


Figure 11-4: Channel Crosstalk Limit

11.4 REFCLK Specifications

For best interoperability, the REFCLK should support following spec:

Table 11-7: REFCLK Specifications

Definition	Value	Unit
Frequency Range	100~300	MHz
Signaling	Single-Ended	
Jitter spec	N/A	

Spec details:

The reference clock is expected to be 100MHz~300MHz and single-ended only to ensure maximum interoperability. The jitter spec is not specified due to the nature of low pass effect from PLL and high pass effect of the source synchronization link.

11.5 BER spec

The BER (Bit Error Ratio) for Standard profile is defined as single-lane spec and the target value is 1e-25 or better. This BER is to be guaranteed as long as the design is fully compliant with the electrical spec.

OpenHBI-L IP designed based on this BER is expected to be good enough in data reliability for most of the use cases, if not all, including ultra large-scale AI systems and mission critical designs, without the need of any error protection and/or correction.

Nonetheless, if, for any reason, users find error protection/correction is useful or required, they might add them in the upper layer of OpenHBI for further enhancement of error-free operation.

The BER for Enhanced profile is defined as single-lane spec and the target value is 1e-15 or better. This BER is to be guaranteed as long as the design is fully compliant with the electrical spec.

CRC-like error detection is strongly recommended to be incorporated in the link layer design to ensure highly reliable performance. Other schemes, such as re-try or FEC, could be considered depending on the users requirements.

12. Clocking, Initialization and Configurations

12.1 Clocking

This OpenHBI-L Clocking section is informative and is intended to illustrate a few high-level clocking example use cases.

Refer to Section 5.2, the TX_Clk and RX_Clk clock edges shall be edge-aligned with D[41:0] relative to the data eye.

Three different clocking scenarios are illustrated below:

- Mesochronous mode: Multiple or All DWORDs in a beachfront (Section 12.1.1)
- Sync mode: Single DWORD (Section 12.1.2)
- Sync mode: Two or more DWORDs (Section 12.1.3)

The Anchor and Chiplets should support Mesochronous clocking as baseline to ensure interoperability and broader use case support. Fixed function chiplets with known configurations may implement a specific clocking mode(s) as needed.

12.1.1 Mesochronous Clocking: All DWORDs can run at same clock

Figure 12-1 illustrates an example employing mesochronous clocking mode.

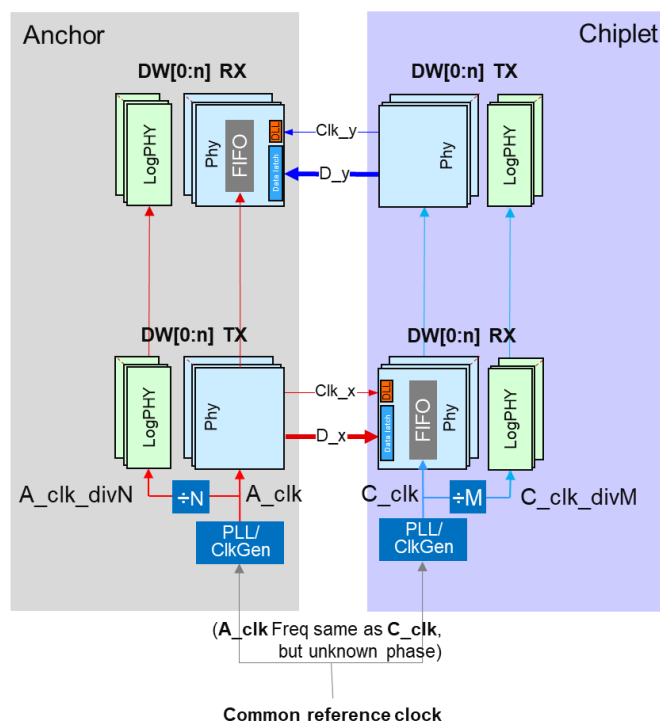


Figure 12-1: Example of Mesochronous clocking mode

Details:

- Anchor and Chiplet has its own PLLs to generate local A_Clk and C_Clk respectively, which has the same frequency but unknown relationship, based on a common clock source.
 - Note: The “Common reference clock” shown can be either fed by an external clock generator to both the Anchor and Chiplet through package C4 bumps or the RefClk (Anchor: RefClk Output, Chiplet: RefClk Input).
- Anchor and Chiplet PHY layers operate as a mesochronous network
- Anchor and Chiplet each have a local clock divider to generate a local LogPHY clock. The divider factors N and M can either be the same or 2^n -multiple relationship (where n = Integer).
- As shown, all DWORD[n:0] on Anchor-side and Chiplet-side run at same PHY clock ($A_clk = C_clk$).
 - The divider N and M will not affect mesochronous network operation
- A FIFO in RX PHY is needed (as shown) to resolve phase uncertainty and absorbs the maximum long-term jitters between the two PLLs.

The advantage of mesochronous clocking is that all DWORD[n:0] are effectively synchronous with each other and FIFO effectively absorbs the jitters between Anchor and Chiplet PLLs and simplifies the alignment and aggregation logic in the Upper Layer.

12.1.2 Sync Mode: Single DWORD

Figure 12-2 below illustrates an example of fully synchronous clocking mode within a single DWORD.

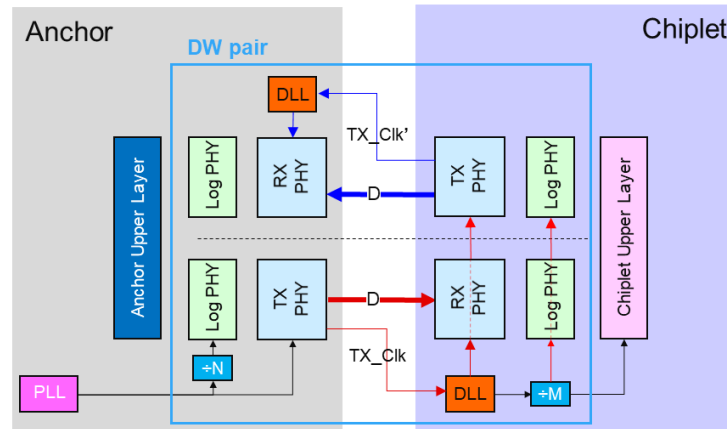


Figure 12-2: Example of Sync clocking mode: Single DWORD

Details:

- Anchor-side PLL generates a local clock which will be used to generate Anchor TX DWORD TX_Clk.
- The TX_Clk is edge-aligned with D[41:0]
- Each RX DWORD has one DLL.
- Chiplet RX DWORD trains the incoming RX_Clk with local DLL to generate a sampling clock centered with the data eye to capture the data
- Anchor TX and Chiplet RX each have a local clock divider to generate a local LogPHY clock. The divider factors N and M can either be the same or 2^n -multiple relationship (where n = Integer).
- Optionally, the Chiplet-side trained sampling clock and the “divided-by-M” LogPHY clock can be used by Chiplet TX DWORD, as shown in red dotted lines.
- Anchor RX DWORD will train the incoming RX_Clk with the local DLL to capture the incoming data.

This clocking architecture has the advantage of lower latency end-to-end, if designed properly.

Note that, in this example, the Anchor-side TX and RX DWORDs will have the same frequency, but the phase relationship will be dependent on both Anchor and chiplet-sides implementation and needs to be resolved at the Anchor Upper Layer.

12.1.3 Sync Mode: Two or more DWORDs

Figure 12-3 illustrates an example of fully synchronous clocking mode with an Upper Layer that uses two or more DWORDs.

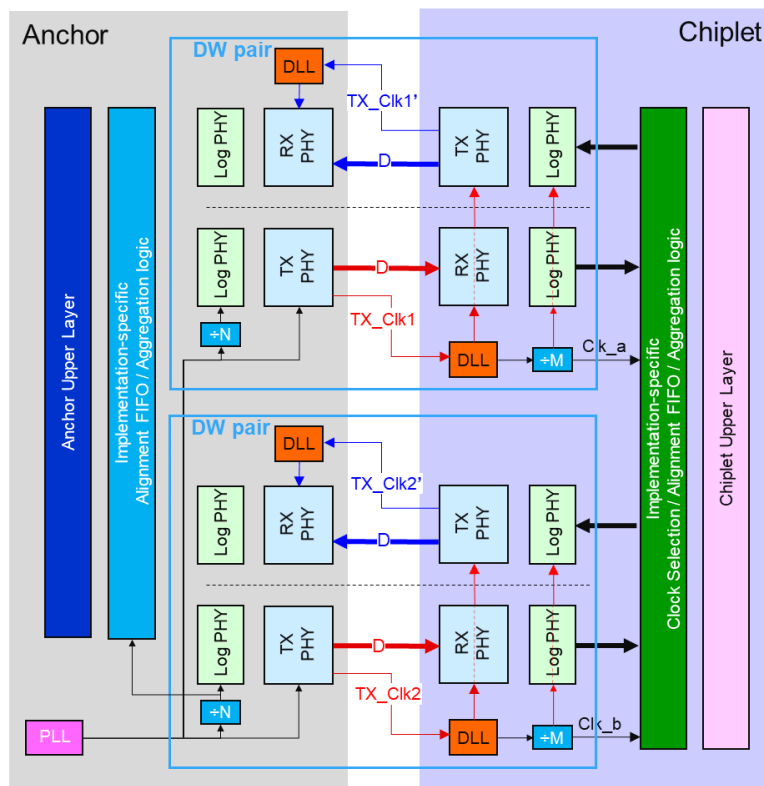


Figure 12-3: Example of Sync clocking mode: 2 or more DWORDs

Details:

- On a per-DWORD pair basis, the clocking details is the same as described in Section 12.1.2.
- Chiplet-side has two RX DWORDs which will output Clk_a and Clk_b respectively by local clock dividers. Clk_a and Clk_b can have a certain amount of skew.
- Chiplet-side has the Clock Selection logic (to select between Clk_a and Clk_b to be used for alignment and aggregation), Alignment FIFO and DWORD Channel Aggregation logic before presenting to the Chiplet Upper Layer.
- Similarly, Anchor side has an Alignment FIFO and DWORD Channel Aggregation logic before presenting to the Anchor Upper Layer.

This example is an extension of Section 12.1.2 and has the same advantage of low end-to-end latency, if designed properly.

12.2 Initialization Flow

The following high-level sequence and timing must be satisfied for the OpenHBI-L instance power up and initialization. Please refer to Figure 12-4 and Table 12-1.

All DWORDS of an OpenHBI-L instance must be properly powered up and initialized.

1. Apply power to the OpenHBI-L instance supplies. During power supply ramp, PwrGood is de-asserted (LOW), Reset and all other input signals may be in an undefined state (driven LOW or HIGH, or Hi-Z).
2. After both Anchor and Chiplet power rails are stable, Anchor drives PwrGood and Reset HIGH at t_0 .
 - All other input signals may be in an undefined state (driven LOW or HIGH, or Hi-Z) at this point.
 - Anchor shall drive PwrGood HIGH during initialization and as long as stable power is maintained.
 - Anchor shall drive Reset HIGH and maintain for a minimum of t_{INIT1} time.
 - During Reset HIGH, each Anchor and Chiplet DWORD must drive TX_Clk_t to LOW and TX_Clk_c to HIGH respectively (TX_Clk = logical LOW state) no later than t_{INIT2} (min) before the Anchor de-asserts Reset (i.e., no later than $(t_{INIT1} - t_{INIT2})$ from Anchor asserting Reset HIGH).
 - Anchor must drive RefClk to a stable frequency no later than t_{INIT2} (min) before de-asserting Reset. RX_Clk_t and RX_Clk_c are input and are driven by another anchor or chiplet.
3. Wait for Chiplet ready to be configured (e.g., PLL or DLL settling time, chiplet specific).
4. Configure each OpenHBI-L DWORD as described in Section 12.3.
5. Perform calibration and training of each DWORD as described in Section 12.4.
6. The OpenHBI-L DWORDS are now ready for Mission mode.

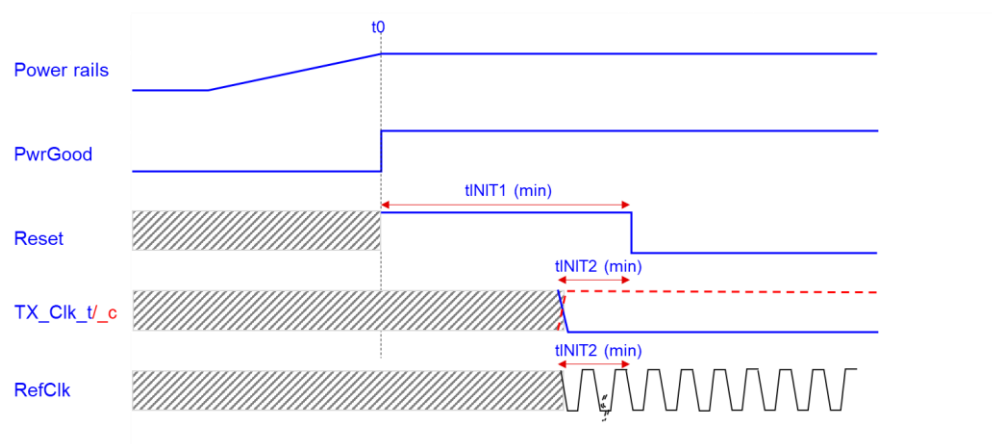


Figure 12-4: Initialization Flow Timing Parameters

Table 12-1: Initialization Timing Parameters

Parameter	Min	Unit
t_{INIT1}	200	usec
t_{INIT2}	10	nsec

12.3 Capability, Control and Status (CCS) Registers

Section 12.3.1 defines a normative set of I3C Bus and Device Characteristics Registers (BCR and DCR).

Section 12.3.2 to 12.3.5 describes an example set of registers of an OpenHBI-L chiplet. All Capability, Control and Status Registers, collectively called CCS Registers, must be accessible via I3C configuration interface by system firmware. The I3C address space is mapped into the device address space. Only the lower eight bits of the address are specified by I3C. The layout of the Chiplet and Anchor CCS Registers are implementation-specific.

Section 12.4 to 12.5 illustrates the typical initialization and the calibration and training sequence.

The registers and bit fields described are from the perspective of Chiplets (as configuration subordinate, with Anchor as configuration manager). All “TX” and “RX” described in this section imply “Chiplet-side TX DWORD” and “Chiplet-side RX DWORD” respectively, unless otherwise stated (e.g., Anchor-side TX, or Anchor-side RX).

For the sake of clarity, the flow described below applies to one DWORD only. All DWORDs in an OpenHBI-L instance must be initialized, configured, calibrated and trained before it can enter Mission mode. It is recommended that all DWORDs can be configured and trained in parallel to reduce initialization time, but it is allowed to perform each DWORD or groups of DWORDs sequentially.

Note: The registers and the bit fields described in the examples below are informative only and are not meant to be exhaustive and do not include all the necessary parameters during the actual sequence.

12.3.1 Required I3C Characteristics Registers

This section describes the normative I3C Characteristics Registers that are required for all I3C compliant devices. This includes the Bus Characteristics Register (BCR) and the Device Characteristics Register (DCR). For interoperability across OpenHBI-L chiplets, the settings of these registers must be as defined in this section. Note that the I3C Legacy Virtual Register (LVR) is not included in this section because OpenHBI-L chiplets are defined as I3C compliant and not as legacy I²C devices.

12.3.1.1 Bus Characteristics Register (BCR)

This is used to describe the I3C compliant device’s role and capabilities for use in dynamic address assignment and common command codes (CCCs).

Table 12-2: Bus Configuration Register (BCR)

Bits	Field Name	Attr	Description	Default
BCR[7:6]	Device role	R	2'b00 = I3C Target. 2'b01 = I3C Controller. 2'b10 to 2'b11 = Reserved for future use by MIPI.	-
BCR[5]	Advanced capabilities	R	0 = Does not support optional advanced capabilities. 1 = Supports optional advanced capabilities. This bit must be set to 1 for OpenHBI chiplets.	1
BCR[4]	Virtual target support	R	0 = Is not a virtual target and does not expose other downstream device(s). 1 = Is a virtual target or exposes other downstream device(s). This must be set to 0 for OpenHBI chiplets.	0
BCR[3]	Offline Capable	R	0 = Device will always respond to I3C bus commands. 1 = Device will not always respond to I3C bus commands. This must be set to 0 for OpenHBI chiplets.	0

BCR[2]	IBI payload	R	0 = No data bytes follow the accepted IBI 1 = One data byte (MDB) shall follow the accepted IBI, and additional data bytes may follow. This must be set to 1 for OpenHBI chiplets.	1
BCR[1]	IBI request capable	R	0 – Not capable. 1 – Capable. This must be set to 1 for OpenHBI chiplets.	1
BCR[0]	Maximum data speed limitation	R	0 = No limitation. 1 = Limitation. This bit must be set to 0 for OpenHBI chiplets.	0

12.3.1.2 Device Characteristics Register (DCR)

This is used to describe the I3C compliant device type for use in dynamic address assignment and common command codes (CCCs).

Table 12-3: Device Characteristics Register (DCR)

Bits	Field Name	Attr	Description	Default
DCR[7:0]	Device ID	R	This must be set to 0x00 for OpenHBI chiplets to specify a Generic I3C Device.	0x00

12.3.2 Recommended (Informative) Control and Status Registers

This section describes a reference set of configuration control and status registers used during initialization, calibration and training sequences as shown in Section 12.4. For the sake of simplicity, the capability registers that the configuration manager can use to discover the chiplet's capabilities before starting the training and calibration sequence are not described in this section.

12.3.2.1 DWORD Configuration Register

This is used to configure the DWORD. There is one copy of this register for each DWORD.

Table 12-4: DWORD Configuration Register

Width (bit)	Field Name	Attr	Description	Default
1	TX/RX	RW	A high level "1" indicates this DWORD is TX A low level "0" indicates this DWORD is RX	0

12.3.2.2 TX - Training and Test Control Register (TX-TTCR)

This is used to control all chiplet-side TX DWORDs of the link during training and testing phase. There is one copy of this register for a set of (one or more) DWORDs that forms the C2C link.

Table 12-5: TX-TTCR (TX - Training and Test Control Register)

Width (bit)	Field Name	Attr	Description	Default
1	Training enable	RW	A high level "1" indicates training mode.	0
1	TX transmit start	RW	Configuration manager writes this bit to "1" to request TX to transmit data. The length of the transmitted data is controlled by the burst length register (BLR).	0
1	Mission mode	RW	A high level "1" indicates Mission mode.	0

Note: "Training enable" and "Mission mode" shall not be set to "1" at the same time.

12.3.2.3 RX - Training and Test Control Register (RX-TTCR)

This is used to control all chiplet-side RX DWORDs of the link during training and testing phase. There is one copy of this register for a set of (one or more) DWORDs that forms the C2C link.

Table 12-6: RX-TTCR (RX - Training and Test Control Register)

Width (bit)	Field Name	Attr	Description	Default
1	Training enable	RW	A high level indicates training mode.	0
1	RX data request	RW	RX will write this bit to indicate it is ready to receive data and request Anchor-side TX to transmit data. Chiplet can generate an I3C IBI (In-Band Interrupt) with MDB to Anchor, or the Anchor can check this bit. The Anchor-side TX burst length is controlled by Anchor's burst length register which is Anchor-specific.	0
1	RX initialization done	RO	RX initialization done. If set, it indicates RX has completed all its trainings. Anchor will check this bit, or chiplet can generate an I3C IBI with MDB when RX initialization is done.	0
1	RX training error	RO	RX will set when there is an error after "RX initialization done" bit is set.	0
1	Mission mode	RW	A high level "1" indicates Mission mode.	0

Note: "Training enable" and "Mission mode" shall not be set to "1" at the same time.

12.3.2.4 LFSR Control Register

The LFSR Control Register is used to configure all OpenHBI-L 40-bit Galois LFSR generators within a DWORD. There is one copy of this register for a set of (one or more) DWORDs that forms the C2C link.

Table 12-7: LFSR Control Register

Width (bit)	Name	Attr	Description	Default
1	LFSR enable	RW	Enables LFSR generator 0 = Disable 1 = Enable	0
1	LFSR preset	RW	LFSR Preset 0 = Normal operation 1 = Preset LFSR to the value specified by LFSR Initial Value Register. Software should set to 1 to preset and then reset the bit to 0 for normal operation.	0
2	Mode	RW	Operating Mode 00 = LFSR mode 01 = LFSR Compare mode 10 = Programmable Pattern mode. In this mode, the TX DWORD will continuously transmit the 40-bit pattern specified by the LFSR Initial Value Register and the LFSR will not advance with TXClk or RXClk. 11 = Reserved	

12.3.2.5 LFSR Initial Value Register

The LFSR Initial Value Register specify the 40-bit initial value (i.e., Seed) for LFSR Transmit, Receive and Compare operating modes. The 40-bit value programmed into this register can also be used for

Programmable Pattern mode to generate a repeating pattern of any 40-bit value to be continuously transmitted which can be useful for some training algorithms.

There is one copy of this register for each LFSR Control Register and is implementation-specific.

Width (bit)	Name	Attr	Description	Default
40	LFSR Initial Value	RW	Specifies a 40-bit value that can be used as LFSR Seed for LFSR mode or LFSR Compare operating mode, or used as a pre-programmed 40-bit pattern to be transmitted repeatedly in Programmable Pattern mode.	0xAAAAAAAA

12.3.2.6 Burst Length Register (BLR)

The burst length register (BLR) specifies the burst length of chiplet-side TX data transmission during calibration and training (sent to the Anchor-side RX). There is one copy of this register for a set of (one or more) DWORDs that forms the C2C link.

Table 12-8: Burst Length Register

Width (bit)	Name	Attr	Description	Default
32	Burst length (BL)	RW	Specifies the number of data beats for each RX data request. Number of data beats transmitted = $8 \times (BL + 1)$.	0

12.3.2.7 DWORD Address Register (DWAR)

The DWORD address register (DWAR) is an index pointer to specify the address of the DWORD from which the I3C register read accesses.

There is one copy of this register for a set of (one or more) DWORDs that forms the C2C link. Please refer to the Anchor or chiplet datasheet on the C2C beachfront organization and how the DWORDs numbering is specified.

Table 12-9: DWORD Address Register

Width (bit)	Name	Attr	Description	Default
8	DWORD address	RW	Specifies the DWORD number accessed during I3C register read access. Valid ranges: 0x00 to 0xFF for DWORD 0 to 255	0x00

12.3.3 Additional Recommended Capability Registers

This section describes a set of additional recommended capability registers to allow a configuration manager to discover the chiplet's capabilities in order to properly configure it. The list below is for reference only and not meant to be exhaustive.

Top level capabilities:

- Number of DWORDs associated with this CCS register set
- Directionality: Fixed TX / Fixed RX / Configurable Bidirectional

PHY layer:

- Gearbox ratios supported

Logical PHY layer:

- Max Logical PHY layer clock frequency (to determine optimal Gearbox ratio)
- Framing / Programmable Extended repeating pattern support
- Parity support
- Bit reordering support
- DBI support
- Logical PHY layer modes supported – one bit per Mode 0-3
- Logical PHY layer Bypass support – Mode 4

Additional Data Integrity support

Error reporting / logging support: examples: Parity, Framing errors

12.3.4 Additional Recommended Control Registers

This section describes a set of additional recommended Control registers to allow a configuration manager to properly configure the chiplet. The list below is for reference only and not meant to be exhaustive.

Top level Control:

- Directionality: TX or RX (if Configurable as TX or RX)

Clocking / PLL related Control (if provided)

Cold Reset (e.g., Anchor-side may control to toggle PwrGood signal)

Software Reset (toggle Reset signal)

PHY layer:

- Gearbox ratio control

Logical PHY layer:

- Logical PHY layer Mode 0-3 or Mode 4 Bypass (which will determine Framing, Parity, DBI enabled or not)
- Framing Programmable Extended repeating pattern control
- Bit reordering

Additional Data Integrity control

Error logging enable / Clear Error log

12.3.5 Additional Recommended Status Registers

This section describes a set of additional recommended Status and Error Logging registers to allow a configuration manager to read the status of the Chiplet DWORD or the OpenHBI-L C2C link. The list below is for reference only and not meant to be exhaustive.

Status Registers:

- TX transmission completion status
- Training completion status of DWORDs or OpenHBI-L C2C link
- Error Log register: example, Parity, Framing errors

12.4 Calibration and Training

Figure 12-5 describes an example OpenHBI-L Anchor TX DWORD to Chiplet RX DWORD calibration and training sequence needed before entering the Mission mode.

Using Mesochronous Clocking mode as an example, there will be a PLL associated with one or more DWORDs that form the OpenHBI-L C2C link. The PLLs on both Anchor and chiplet are programmed to the target frequency and wait for the settling time and locked. The PLL output will serve as the source of DWORD TX_Clk.

The following steps are done for each DWORD either in parallel, in groups or sequentially:

- 1 TX DWORD impedance is calibrated to ensure the driver strength meets the specification.
- 2 Anchor configures TX and RX DWORD to get ready for training and calibration. The high-speed trainings can be divided into two phases.
- 3 “High Speed Phase 1” includes pattern detection, clock position training, VREF leveling, and data deskewing.
 - TX transmits a data pattern for the RX to train the optimal clock sampling edge (Data-to-RX_Clk Position Training) at the center of the data eye. Clock position trainings can be done by on-chip CDR circuitry or controlled via firmware. Refer to Section 9.4 for more details.
 - VREF leveling minimizes the duty cycle distortion of the RX. The leveling is performed by sweeping the VREF levels to look for optimal VREF level to maximize the eye height.
 - Data deskewing aligns all D[41:0] to remove static timing skews among different lanes.
 - Optionally, the clock position training, VREF leveling and data skewing can be performed iteratively to achieve the best settings.
 - At the end of this phase, RX will notify TX that the High Speed Phase 1 has completed.
- 4 “High Speed Phase 2” includes clock positioning training with channel equalization parameters and optional lane margin test.
 - TX transmits an LFSR pattern for the RX to train the optimal clock sampling edge at the center of the data eye. Refer to Section 9.4 for more details.
 - EQ training and search algorithm determines the EQ settings to make the widest eye opening
 - Optionally, the clock position training and EQ parameters search can be performed iteratively to achieve the optimal settings.
 - At the end of this phase, RX will notify TX that High Speed Phase 2 has completed and can provide optional marginal lane information if the lane margin test is performed.
- 5 After all the initialization sequence is complete, both TX and RX enters Mission Mode (normal operation mode).

The Data-to-RX_Clk clock positioning training, VREF leveling, Data deskewing, channel equalization parameters search algorithms and the optional Lane margin test are implementation-specific.

After successful completion of training, OpenHBI-L link shall meet the BER of $\leq 1e-25$ for Standard Profile and BER $\leq 1e-15$ for Enhanced Profile data rates.

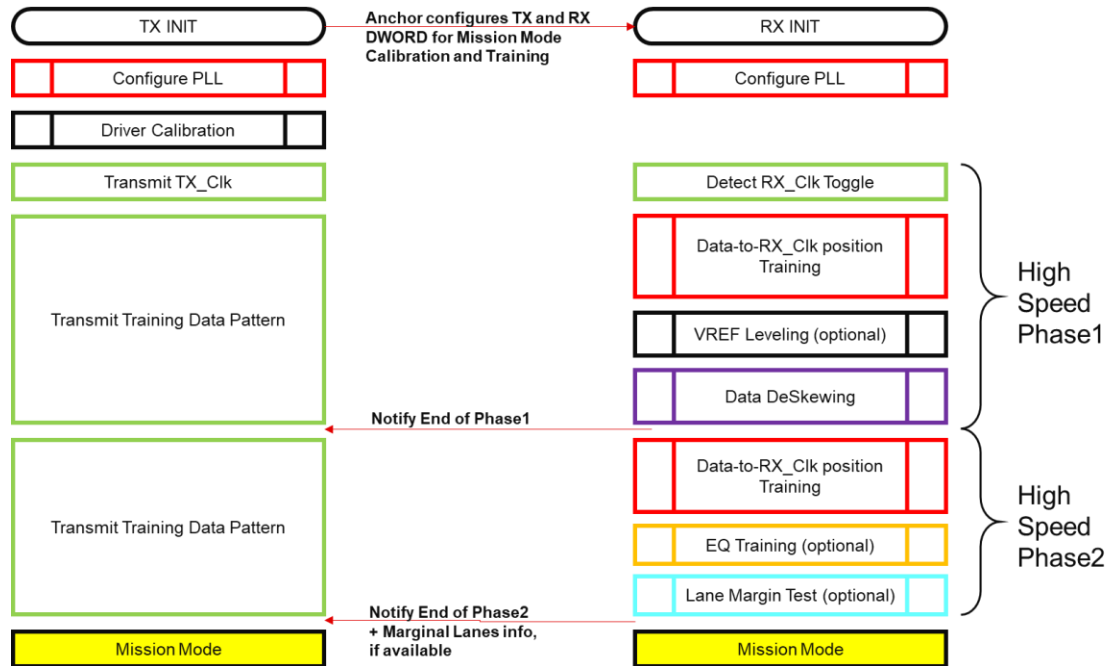


Figure 12-5: Calibration and Training Sequence

12.5 Continuous Training during operation

The OpenHBI-L link is conceived to require continuous uninterrupted operation. Training is performed at initialization of the link for lane deskew and sampling edge placement. Thereafter the link must continue to operate uninterrupted without training interruption. This requirement is put forth to align with the likely usage of the link to transport SerDes backhaul data traffic which also does not allow for interruption. Due to the extremely high bandwidth of the link, if the receive side were to ask for traffic interruption for retraining, an unfeasibly large memory would be required to store the unsent traffic in the transmitter. Lane status and power state information communicated between the transmit and receive ends are low bandwidth and not suitable for synchronously stopping traffic for the interruption.

The design must have sufficient margin to maintain BER with only initial deskew, VREF and sampling edge placement training, or a mechanism must be put in place to track those quantities and update them in response to voltage and temperature drift while traffic is flowing.

There are different ways to implement the continuous training mechanism during operation. Figure 12-6 illustrate an example implementation (informative) by placing a sampling circuit on the RX_Clk to monitor continuously or periodically monitor the eye position and adjust the sampling edge, VREF or skew as needed. There is no requirement on data lines for transition density, scrambling or DC balanced patterns.

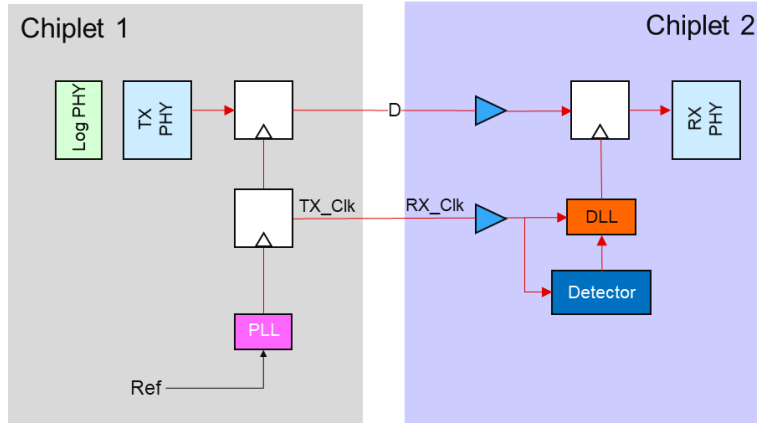


Figure 12-6: Example DLL-Detector loop for Continuous Training During Operation

13. Power Management

OpenHBI-L supports two power management modes:

- (1) The Upper Layer can send static data (e.g., not toggling or all 0s) and indicate “idle” state when mapping the protocol to the C2C data traffic which will reduces ~95% of the IO power.
- (2) The Anchor or external controller can terminate the activities on both chiplets (stopping TX activities to quiesce the bus before disabling RX) and then powering down the OpenHBI-L link. This mode is used when the OpenHBI-L C2C link will not be used for an extended period of time and can be re-initialized and/or re-configured for the next applications. The powering up procedure after powering down is the same as described in Section 12.2.

14. References

[1] JEDEC JESD238 HBM3 specification Jan 2022

[2] MIPI I3C[®] v1.1.1 (or later) Specifications - <https://www.mipi.org/specifications/i3c-sensor-specification>

[3] MIPI I3C BasicSM v1.0 Specifications - <https://resources.mipi.org/mipi-i3c-basic-v1-download>

[4] ODSA OpenHBI Inter-Die Link Layer Specification

Appendix A - Requirements for IC Approval (to be completed Contributor(s) of this Spec)

List all the requirements in one summary table with links from the sections.

Requirements	Details	Link to which Section in Spec
Contribution License Agreement	OWF CLA 1.0 (modified)	Please refer to Section 1
Are All Contributors listed in Sec 1: License?	Yes	
Did All the Contributors sign the appropriate license for this spec? Final Spec Agreement/HW License?	Yes	
Which 3 of the 4 OCP Tenets are supported by this Spec?	Openness Yes Efficiency Yes Impact Yes Scale Yes	Please refer to Section 2
Is there a Supplier(s) that is building a product based on this Spec? (Supplier must be an OCP Solution Provider)		
Will Supplier(s) have the product available for GENERAL AVAILABILITY within 120 days?	Seeking exception to have extended period for silicon availability. Test chips expected in 2023.	

Appendix B-_____ - OCP Supplier Information (to be provided by each Supplier of Product)

Company:

Contact Info:

Product Name:

Product SKU#:

Link to Product Landing Page:

Please complete the following [2021 Supplier Requirements](#). This link will allow you to create a copy for your product-specific requirements.

For OCP Inspired™,

- All Suppliers must be a OCP Solution Provider.
- All Suppliers must run the Hardware Management Conformance Checks and all products must meet the [OCP Hardware Baseline Profile v1.0.0](#).
- All Suppliers must fill out a Security Profile (No Badge Level) for their product.

For OCP Accepted™, Supplier details are required.

- All Suppliers must be a OCP Solution Provider.
- All Suppliers must run the Hardware Management Conformance Checks and all products must meet the [OCP Hardware Baseline Profile v1.0.0](#).
- All Suppliers must fill out a Security Profile (No Badge Level) for their product.
- All Products must meet the Open System Firmware requirements.
- All Products must have source code for BMC, if applicable. This must be in the OCP Github repository.

List all the requirements in one summary table with links from the sections.

Requirements	Details	Links
Which Product recognition?	OCP Accepted™ or OCP Inspired™	Provide Marketplace Link
If OCP Accepted™, who provided the Design Package?		Link
2021 Supplier Requirements for your product(s)		Link