Pipe Adapter for PHY implementation

ODSA Project Workshop

September 10, 2019

Robert Wang

Bapi Vinnakota

Consume. Collaborate. Contribute.

Contributors and Collaborators

- Halil Cirit – Facebook
- Ramin Farjad – Aquantia
- Brian Holden – Kandou Bus
- Rita Horner – Synopsys
- David Kehlet – Intel
- Mark Kuemerle – Global Foundries
- Paul Mattos – Global Foundries
- Bapi Vinnakota – Netronome
- Robert Wang – AnalogX
- Jerrold Wheeler – Synopsys

Meeting Summary and Change Notes

Revision	Date	Comment
1.0	8/20/2019	• Added PIPE adapter definition to be more specific on where it is • Added compatibility to VIP support • Suggested number of PIPE lanes that can be supported
1.1	8/27/2019	• Changed PIPE interface requirement from compatible to compliant • Introduced implementation matrix for anyPHY adoption • Discussed each of the PHY implementation at high level on what needs to be done
1.2	9/3/2019	• Added the last slide, may need to offer two requirements based on PIPE4.4.1 or PIPE5.2, or just choose 1 of them.
1.3	9/10/2019	• Concluded each case and their respective IO connectivity technology • Choosing PIPE5.2, user elects backward compatibility as necessary

References

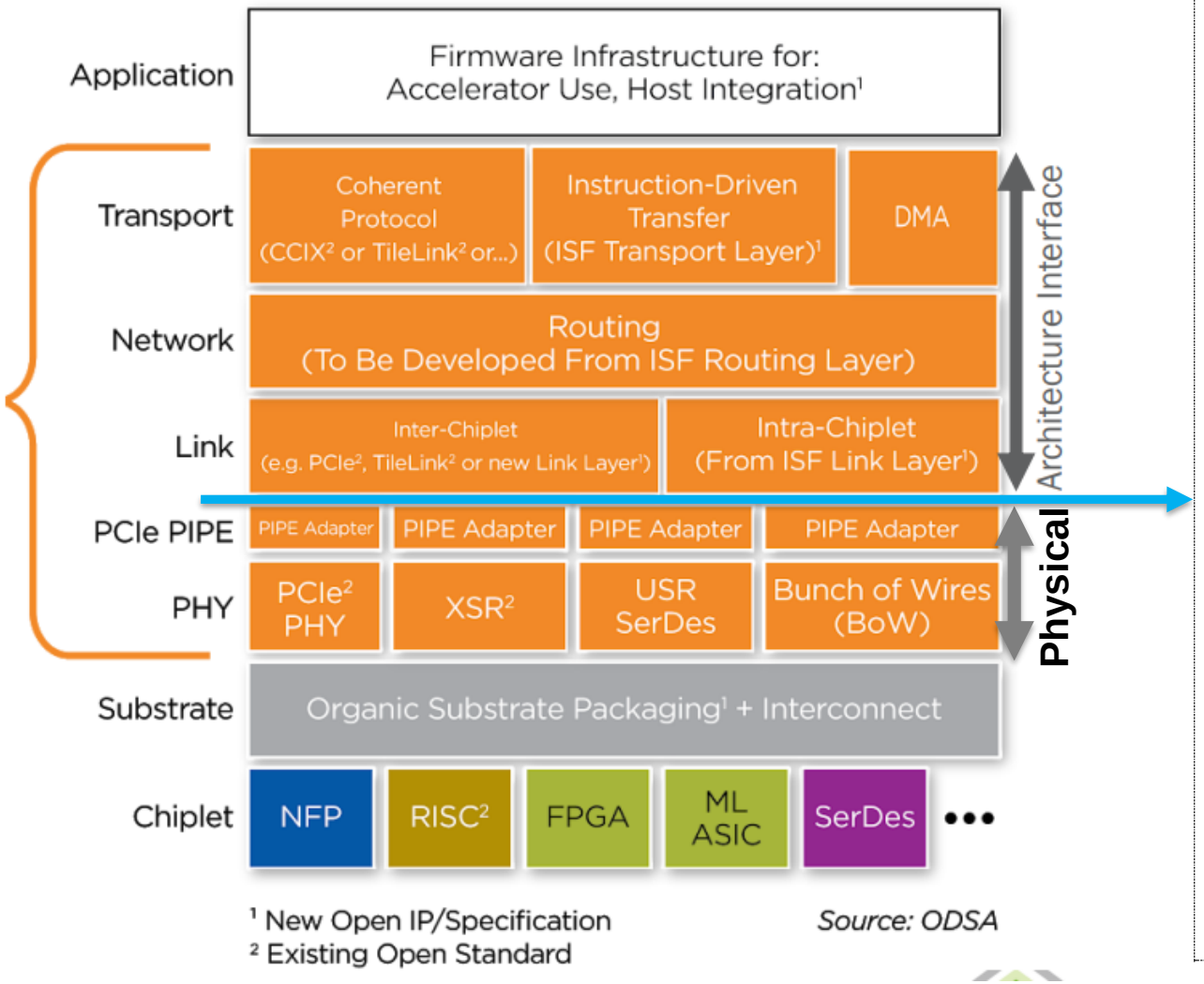
- ODSA Technical Intro, Bapi Vinnakota, https://146a55aca6f00848c565-a7635525d40ac1c70300198708936b4e.ssl.cf1.rackcdn.com/images/a00b210eaf5d10ee932d1b97c960d4c24f549f2a.pdf
- PIPE Adapters, Brian Holden, https://146a55aca6f00848c565-a7635525d40ac1c70300198708936b4e.ssl.cf1.rackcdn.com/images/abfb1d3954d95426e5c4485771ae16120e963251.pdf
- Outline of specification of a PIPE adapter for BoW v2.docx, Brian Holden

To Address What Questions

- How does a die-die SerDes/PHY use a PIPE interface?
- How does a 112G XSR map to PIPE interface?
- How does a HS (//) IO interface map to PIPE interface?
- What would the general control signal guidelines for the PIPE interface?

- Need input from PHY suppliers and MAC Controllers

Pipe Adapter (a.k.a. SerDes PCS)



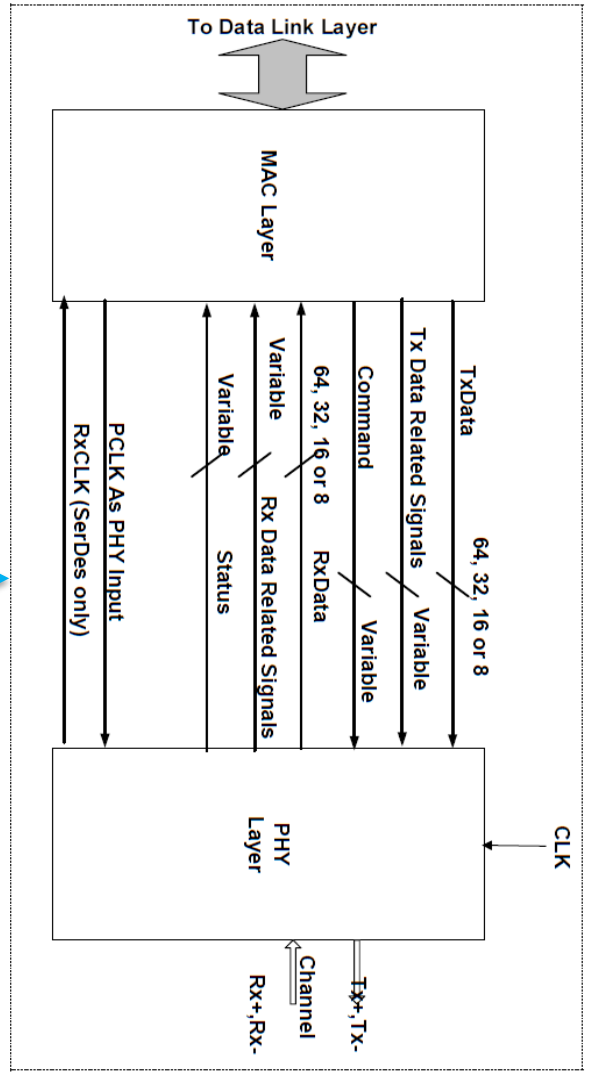


Figure 3-1. PHY/MAC Interface

Pipe Adapter

- Is an added PCS logic layer in serdes PHY to support existing PIPE interface
- Follows PIPE specification (eg. PIPE 5.2 or 4.4.1)
- Implemented by PHY providers to ensure PIPE compliance

Note

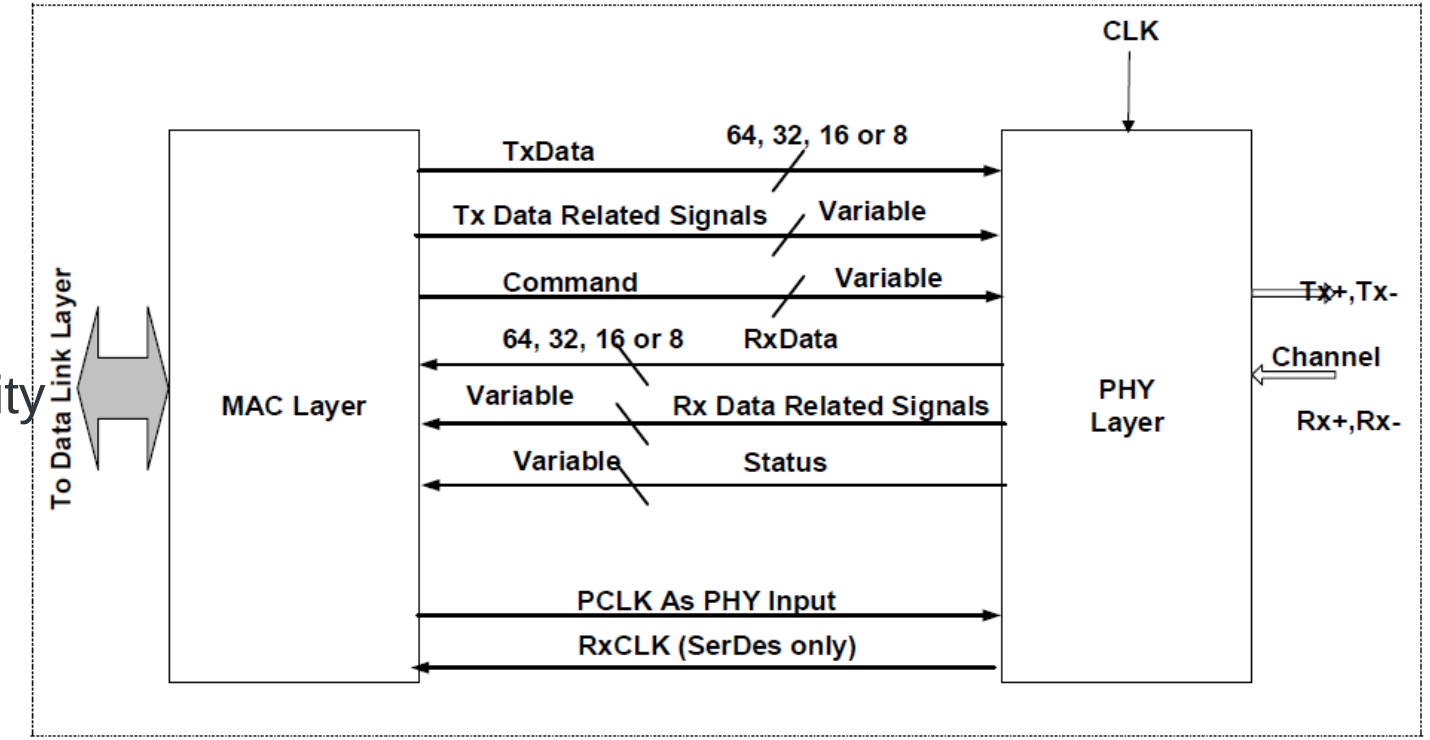
- Each PIPE adaptor is different as it is PHY specific

Idea is not new, but an extension

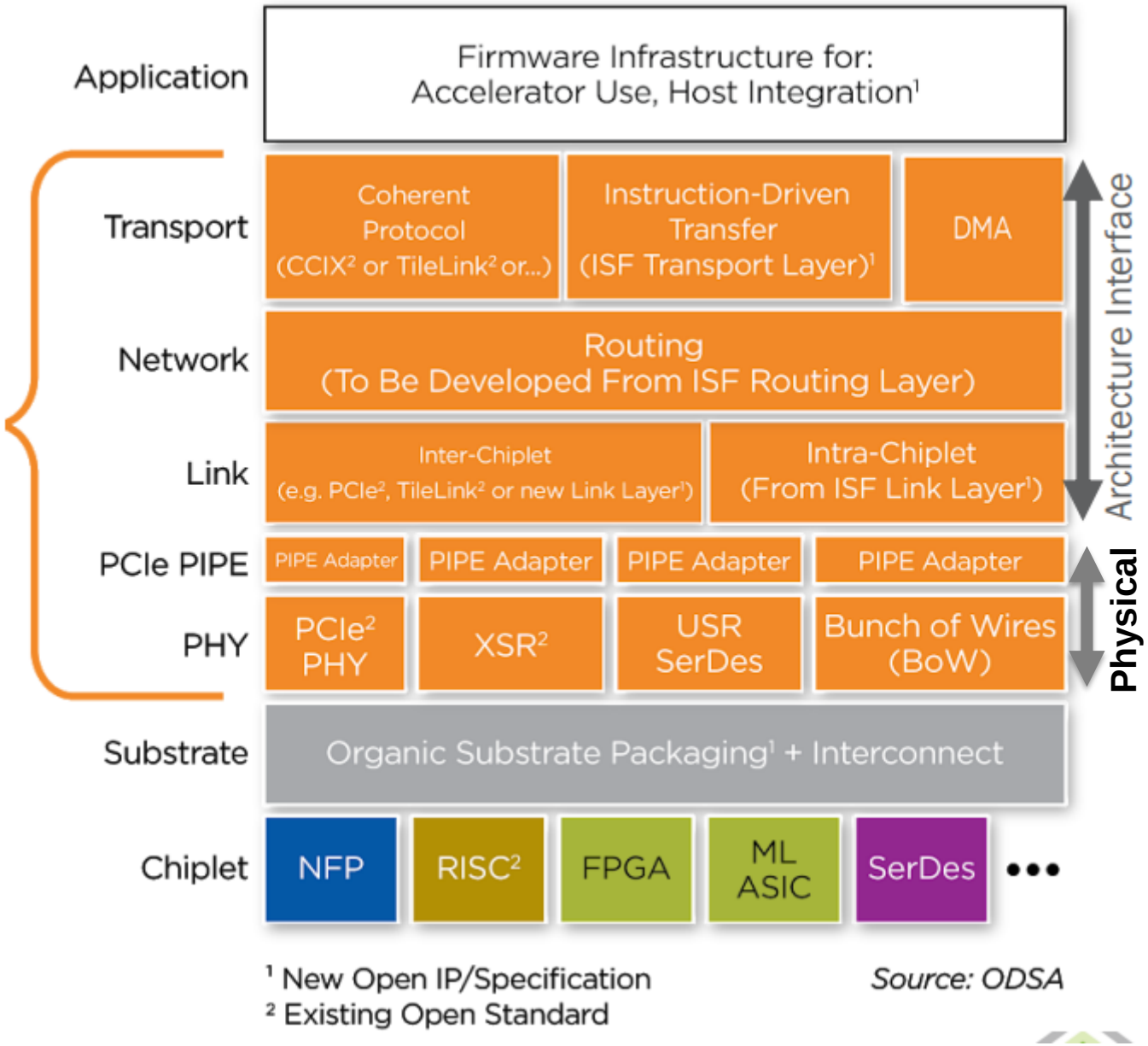
- PIPE supports many different protocol standards with different datarates
  - Namely: PCIe, SATA 3.0, USB 3.1, DP 1.4
  - Also has Converged IO PHY 1.0
- Has support for SR application

- Important to note
  - Not making changes to standards
  - Not proposing changes to PIPE
  - Leverage existing solutions
  - Accelerate adoption and interoperability

Figure 3-1. PHY/MAC Interface



Pipe Adapter Focus



- PIPE interface ~~compatibility~~ Compliance
- Verification IP Compatibility
- Can support 1 to M number of PHY lanes
- Can support 1 to N number of lanes on PIPE side (Typical Nmax = 16, can be different)
- Can support predefined PIPE datarates or user defined datarates
- PHY implementation agnostic (anyPHY)
- Guidelines for PHY vendor to implement

Some Definitions

- PHY Lane (M): refer to the number of lanes on the PHY side
- PHY Lane Rate: specify one lane data rate of the PHY. This can be each lane in a serial PHY, or each lane in a parallel PHY
- PHY Link Rate: specifies the total BW of the PHY with 1 to M number of PHY lanes

- PIPE Lane (N): refer to the number of data path on the PIPE interface, each PIPE lane can consist of data bus width defined in PIPE
- PIPE Lane Rate: specify the data rate of 1 Lane PIPE interface
- PIPE Link Rate: specifies the total BW of the PIPE with 1 to N number of PHY Lanes (eg: x1, x8, x16)

- Data width: refers to the per lane data width as per PIPE specification

Implementation Matrix

- To accommodate different PHYs

		PHY Lanes (M) vs. PIPE Lanes (N)	
		M = N	M != N
PHY Lane Rate vs. PIPE Lane Rate	=	Case A	n/a
	!=	Case B	Case C

- Goal of the PHY Adapter for chiplet/MCM applications
  - Use the existing PIPE specification to interface different PHY layers

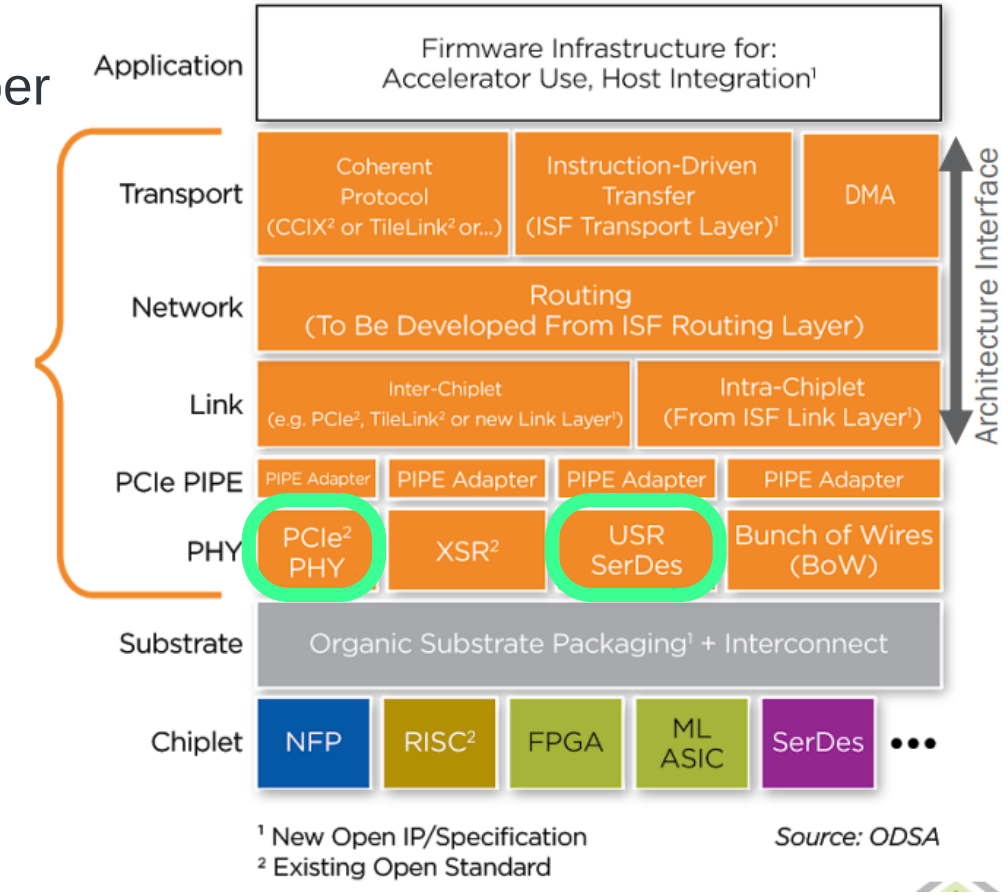
PHY Link Rate	**=**	**PIPE Link Rate**
M * PHY Lane Rate	=	N * PIPE Lane Rate

  - Coding, FEC can be taken into account, and yes, formula can be more extensive

Case A Condition

		Number of Lanes	
		M = N	M != N
Lane Datarate	=	Case A	Case D
	!=	Case B	Case C

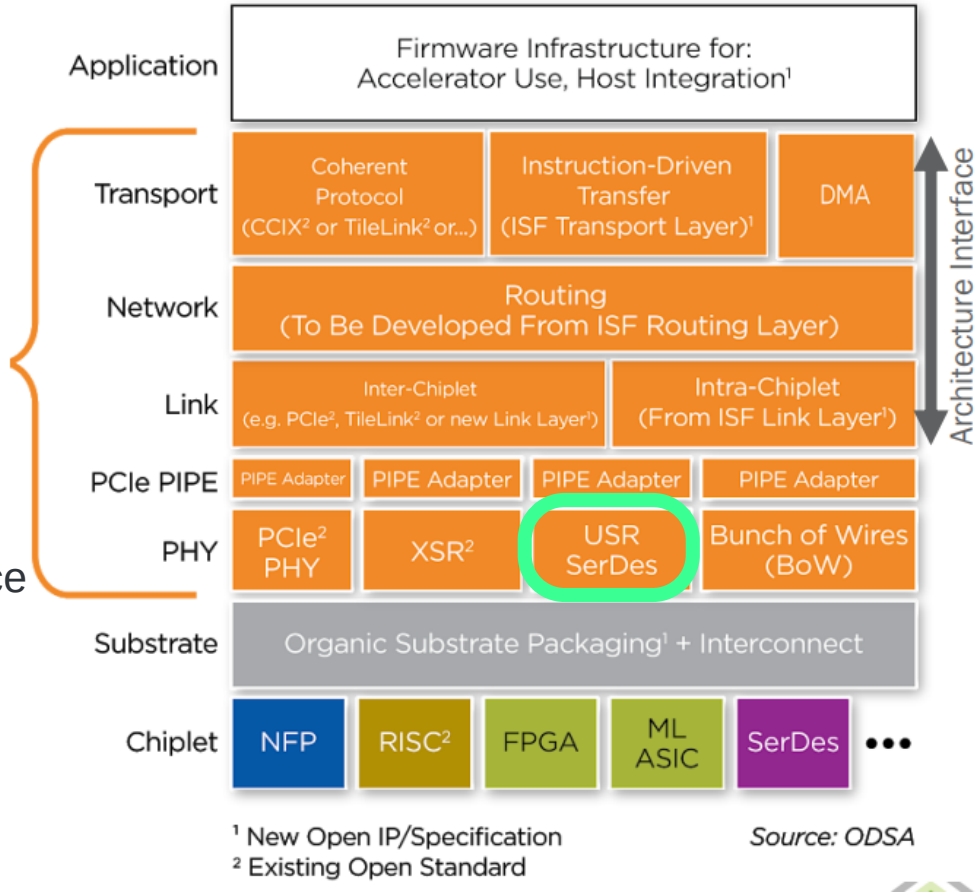
- Example
  - Existing PCIe PHY operating at defined data rate and per lane data width (bus width)
  - A die to die power optimized SerDes link where lane rates and lanes match PIPE Link rates
- Solution Proposal
  - Use existing PIPE interface solution
    - PHY to use existing PIPE datarates for all implementation
  - No changes are required, completely compliant
  - Suggested pin mapping can be provided in a requirements document



Case B Condition

		Number of Lanes	
		M = N	M != N
Lane Datarate	=	Case A	Case D
	!=	Case B	Case C

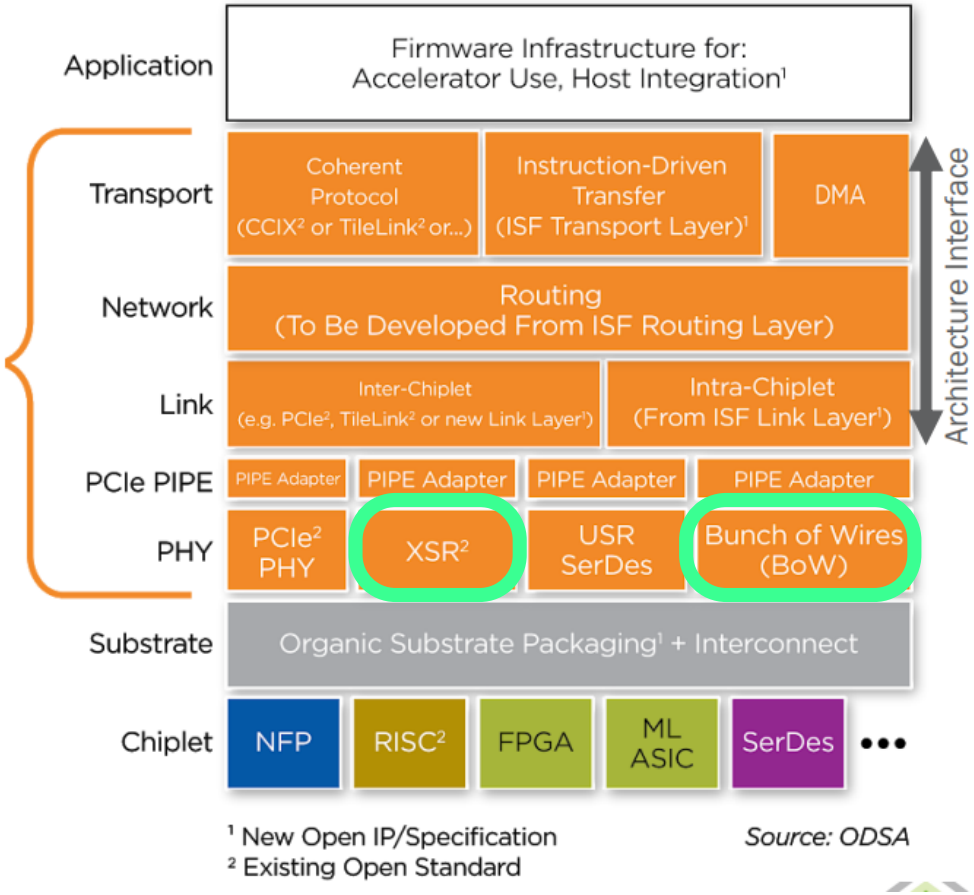
- Condition: PHY lane datarate != PIPE lane datarate
- Example
  - SATA operating at less than or equal to PIPE Lane Rate but same number of lanes (using TXDataValid/RXDataValid)
  - AnalogX AXDieIO Serdes to use PIPE interface
- Solution Proposal
  - Use existing PIPE interface solution to match PIPE Lane Rate and PHY Lane Rate (two methods)
    - To run at PHY rate, lower/change suggested PCLK at PIPE interface
    - Can also change PCLK, data width, and TXDataValid/RXDataValid to ensure overall same data rates between MAC and PHY
    - Controller to adopt compatibility with greater set of datarates in addition to outlined in PIPE specification
  - No PIPE interface or MAC changes are required
  - Can also use SerDes Mode in PIPE5.2



Case C Condition

		Number of Lanes	
		M = N	M != N
Lane Datarate	=	Case A	Case D
	!=	Case B	Case C

- Example
  - 112G XSR operating at single or quad configuration (M = 1 or 4) but need to map to PIPE lane configuration (N = 4 or 16)
  - BoW IO connectivity solutions
- Solution Proposal
  - Should be driven by those PHY implementors
  - Use SerDes Mode defined in PIPE 5.2
  - Require customization due to mapping different lanes and rates, but the customization reside in the adapter layer.



Implementation Proposal (WIP)

- Based on existing PCS and MAC definitions, outline implementation details

PIPE 5.2	Case A	Case B	Case C
MAC/Controller	Follow PIPE 5.2 • Maybe backwards compatible to PIPE4.4.1	Follow PIPE 5.2 • Implement SERDES mode as per PIPE spec 5.2 • Accommodate greater number of PCLK rates	Follow PIPE 5.2 • Implement SERDES mode as per PIPE spec 5.2
	Example MAC/Controller • Any PIPE5.2 or PIPE4.4.1 compliant controller	Example MAC/Controller • Any PIPE5.2 or PIPE4.4.1 compliant controller	Example MAC/Controller • PIPE5.2 compliant controller
Pipe Interface			
Pipe Adaptor (SerDes PCS)	Follow PIPE 5.2 • Maybe backwards compatible to PIPE4.4.1 and previous PCIe Rates	Follow PIPE 5.2 • Recommend to use SerDes Mode in PIPE 5.2 • Define additional PCLK for each datarates	Follow PIPE 5.2 • Use SerDes Mode in PIPE 5.2 only
	Example PHYs • PCIe SerDes or PCIe SR SerDes	Example PHYs • AnalogX AXDieIO SerDes	Example PHYs • BoW IOs • 112G XSR SerDes

Next Step

- Each PHY provider to outline on how their PHY can be connected to publicly available MAC/Controllers via the PIPE interface.
- This can be an application note, a requirements document, or a generic white paper