Theme Article: Hot Interconnects 26

A Bunch-of-Wires (BoW) Interface for Interchiplet Communication

Ramin Farjadrad
Marvell Technology Group

Mark Kuemerle
Marvell Technology Group

Bapi Vinnakota
Talumbra Services

***Abstract*—Multichiplet system-in-package designs have recently received a lot of attention as a mechanism to combat high SoC design costs and to economically manufacture large ASICs. These designs require low-power area-efficient off-die on-package die-to-die communication. Current technologies either extend on-die high-wire count buses using silicon interposers or off-package serial buses. The former approach leads to expensive packaging. The latter leads to complex and high-power designs. We propose a simple bunch-of-wires interface that combines ease of development with low-cost packaging techniques. We develop the interface and show how it can be used in multichiplet systems.**

Chiplet-based designs, based on the integration of multiple die in a single package using system in package technologies, have recently received attention as a mechanism to extend Moore's law.[1–4] AMD,[5] Intel,[6] Marvell,[7] and Xilinx[8] have announced chiplet-based products. SoC development costs in newer process nodes are rising exponentially,[9] resulting in limited design starts and innovations. To reduce design costs, designers purchase IC components as third party IP. Even with IP purchase, analog, photonic, or RF IC developments in new process nodes, like FinFet, consume more time and effort, require more verification, and carry more risk.[2–4]

Chiplet-based designs can lower development cost and time[1] by decoupling development cycles of complex SoCs through heterogeneous integration. In chiplet designs, RF, analog, photonic, logic, and memory can be developed in a process node optimized for that specific function.

Digital Object Identifier 10.1109/MM.2019.2950352
Date of publication 30 October 2019; date of current version 14 January 2020.

Chiplets from multiple process nodes are integrated into a single package to form a product.[1–4] A single chiplet can be used in designs across several process nodes providing economies of scale. Breaking large chips into multiple chiplets increases product yield and lowers total cost of final product.[5,10]

Chiplet-based designs incur higher packaging costs than do monolithic devices. They also require interchiplet links to transport data between chiplets. These links carry data that would be transported on-die in a monolithic design. Off-die data transfers may consume more energy than on-die data transfers.

A single chiplet can be used in designs across several process nodes providing economies of scale. Breaking large chips into multiple chiplets increases product yield and lowers total cost of final product.

Individual chiplets will need new interfaces for die-to-die communication. Two classes of interfaces have been developed:

- Interfaces, such as the Intel advanced interface bus[11] and the (application-specific) high-bandwidth memory interface (HBM)[12] are derived from highly parallel on-chip buses and use many slow wires, each operating at 1–2 Gb/s, to transport data between chiplets. While they offer design simplicity, these interfaces incur higher packaging costs.
- Serial interfaces derived from board-level SerDes links, e.g., PCI express, use a few serial high-speed wires, operating at 10 s of Gb/s, to transport data.[13,14] While suitable for traditional packages, these interfaces are more expensive to design and potentially experience higher latency and incur higher power.[15]

We propose a new interface, the bunch-of-wires (BoW) that transfers data at up to 4 Gb/s over limited trace length up to 10 mm. The authors have learned (informally) that many companies have similar internal interfaces. We propose a standard interoperable definition. The basic interface can be enhanced for higher data rates by: 1) using terminated impedance matched traces to increase the data rate per trace without trace length limitation; and 2) using bidirectional data transfer to again the double data rate. The interface is described in detail in the "BoW Interface" section.

The BoW interface can combine the best attributes of parallel and serial interfaces. The interface reduces wire count, is easy to design, and can also be used with inexpensive package manufacturing techniques. The designer can choose to increase interface implementation complexity in line with performance requirements. We discuss the trade-offs in the "BOW Design and Reuse" section.

The open-domain specific architecture (ODSA) is a new workgroup in the Open Compute Project.[16] The ODSA workgroup aims to reduce accelerator development costs by creating an open interface for interchiplet communication.[17] This will allow product designers to create best-in-class accelerators by assembling best-in-class chiplets from multiple vendors. In the "System Integration" section, we demonstrate how the BoW interface can be integrated into the ODSA reference accelerator architecture. We start with a review of connectivity and packaging for chiplets.

CHIPLET CONNECTIVITY AND PACKAGING

Multichip packaging technologies are of two types:[1–4] 1) traditional multichip module (MCM) packaging; and 2) newer packaging techniques such as wafer-level fanout (WLFO),[18,19] silicon bridges,[6] and silicon interposers.[10,20]

Multichip Modules for Regular Bumps

A traditional approach is the MCM packaging, where the chiplet dice all sit on an organic (e.g., FR4) package substrate and are connected using the PCB traces on the package substrate. MCMs have been in volume production at low costs for decades. The pad pitch of the chiplets on an MCM substrate typically are 100 μm or higher. Such Chiplets can be screened for known-good-die (KGD) at the wafer level during the production with standard test equipment, a major advantage in improving the yield of the packaged part, and a major cost saving.

The low pad and trace density of the MCM package substrate can limit the interchip throughput. One can use SerDes cores to multiplex lower rate parallel data into higher speed data pipe over each package trace. Conventional multi-Gbps

SerDes incur higher power, area, latency, and design complexity to achieve this throughput.

Advanced Packaging for Microbumps

Silicon Interposers (e.g., TSMC CoWoS)[20] or silicon bridges (e.g., Intel EMIB)[6] provide higher density routing between the chiplets than a simple organic substrate. This allows chiplets to use microbumps for IO (with a bump spacing of 50-80u) to greatly increase the interchiplet bandwidth density. In the interposer solution, the chiplet dice are assembled on top of a large silicon chip that acts like the package substrate with high density routing between the chiplets. In the case of an embedded silicon bridge, a small slice of silicon is embedded in the organic package substrate to provide the same high-density routing as an interposer, but with a smaller size and thus lower cost.[6] Silicon-based interconnect solutions are much more expensive solution than the traditional MCM packaging.

WLFO,[18,19] a more recent but relatively simpler packaging technology, uses a redistribution layer to connect fine pitch pads for dense interchiplet connectivity. The redistribution layer is also used to fanout to regular-size bumps for lower density connectivity on a regular laminate.

One challenge with fine pitch interconnect is that screening for KGD before packaging will require denser probe cards or newer test techniques. However, adequate coverage has been reported for the HBM volume production.[1,21]

Interconnect Requirements

Based on a survey of current technologies,[15] the requirements for interchiplet interconnect are as follows:

1. Throughput efficiency: 0.1 Tbps/mm to 1 Tbps/mm.
2. Energy efficiency $<$ 0.5–1.0 pJ/bit.
3. Small silicon area/port for dense integration. To be pad limited, not silicon limited, for pitch $<$120 μm.
4. Trace length range: 1–50 mm for arrangement flexibility and heat dissipation.
5. Total Latency $<$ 5–10 ns.
6. Minimal complex circuitry to enable easy and fast port into wide range of process nodes.
7. Single supply compatible with logic Vdd in existing SoCs/ASICs in popular process nodes.
8. Minimal technology licensing requirements.

BOW INTERFACE

The BoW specification is a simple, open, and interoperable interchiplet interface technology that meets the requirements listed above. The content of this section expands by Farjad and Vinnakota[22] and is complementary to the content of Kuemerle *et al.*[23]

BoW Interface

The BoW interface uses the simplest form of CMOS IOs. A BoW implementation is expected to be easy to port to multiple process nodes. At the transmitter, a CMOS inverter is used to send full levels of Gnd and Vdd, for 0 or 1 logic values respectively, to generate single-ended NRZ (non-return-to-zero) signaling. At the receiver, a CMOS Latch is used to latch in the received signal at the source-synchronous clock edge. The latching of the received data can be done on both edges of the clock (DDR) and as a result the clock runs at half the data baud rate.

The BoW interface can optionally be enhanced for the better performance. Any enhanced mode is required to be bump compatible with the basic interface. We envision two enhancements suitable for regular bump IO.

Because no line termination is used in the BoW Base, the signaling baud rate (Gbaud) and trace roundtrip delay (ns) product is relatively fixed, which is a function of the signal baud and slew rate. In most CMOS IOs, this "Gbaud x nsec" product is 0.20–0.4 depending on the signal slew rate. For example, a BoW Base with 2 GBd signaling, the practical trace delay will be 0.10–0.2 ns (0.20/2–0.40/2 GBd). This roundtrip timing delay is equivalent to 10–20 mm over a typical FR4 substrate. If we can limit the die-to-die distance to 10 mm, the BoW Base data rate can grow to 4 GBd at a clock rate of 2-GHz DDR. At very short trace lengths, $\sim$ 1 mm, the BoW base data rate can go as high as 8 GBd, but more complex clock-data alignment circuitry may be required.

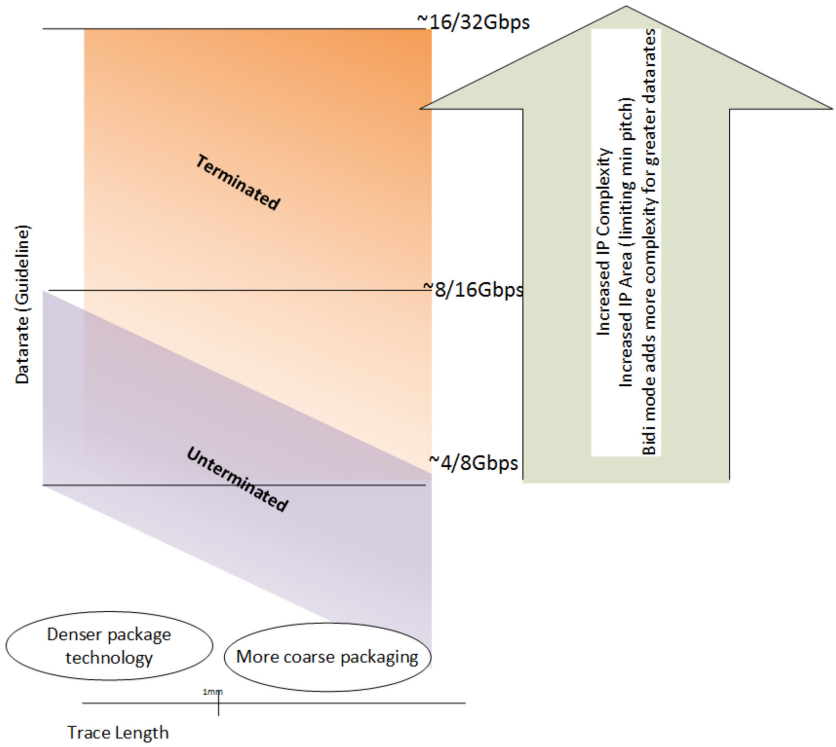

Figure 1. BoW interface operating modes.

Enhanced Modes

The BoW interface can optionally be enhanced for the better performance. Any enhanced mode is required to be bump compatible with the basic interface. We envision two enhancements suitable for regular bump IO. When used with regular bumps, the circuitry per bump may be larger than the area of the pad for a microbump. Figure 1 captures the relationship between the basic and enhanced modes.

BoW Terminated (BoW TD) By designing the traces to have a fixed characteristic impedance (typically ~50 Ω), and terminating them, we can suppress most of the signal reflections, thus, removing the constraint of being able to drive the fixed baud rate (GBd) and trace timing delay (ns) product. As a result, a terminated link can push the data rate higher for longer trace lengths. This enhancement is called BoW TD.

BoW Bidirectional (BoW BiDi) Most physical interfaces instance offer symmetric bandwidth in both directions. BoW BiDi leverages this by using simultaneous bidirectional signaling. Data are transmitted in a physical channel in both directions simultaneously, every port is both an input and output port. A hybrid block, placed between the pad and BoW Tx/Rx ports, creates a Bow BiDi port that separates the receive and transmit signals. BoW BiDi provides a maximum aggregate (i.e., receive + transmit) throughput twice that of BoW TD over one set of wires. BoW BiDi can provide up to 32 Gb/s per trace with a DDR clock of 8 GHz.

BOW DESIGN AND USE

BoW can offer a graceful tradeoff of packaging versus circuit design, as shown in Figure 1 below. BiDi mode adds IP design complexity, but doubles the data rates. We show that BoW Basic with microbumps and BoW BiDi with regular bumps offer very similar performance at similar total costs. Our discussion on the BoW design and use focuses on these two modes.

Bump Maps

A BoW Slice has two clock ports per 16 data ports. The BoW specification itself does not specify a bump map. Figure 2 shows a bump map for a BoW slice. We expect a bump map to be suitable for all BoW modes. Multiple slices can be stacked vertically or linearly, to achieve higher throughput per mm at the die edge. All other modes of BoW are expected to be compatible with BoW Basic.

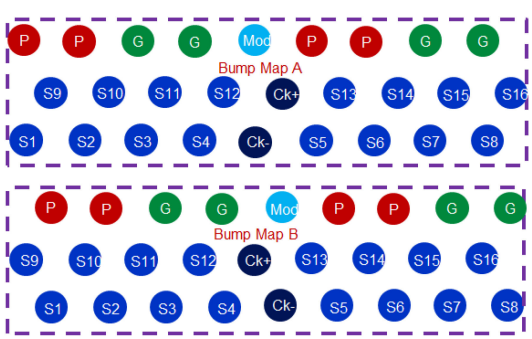

Figure 2. BoW bump map.

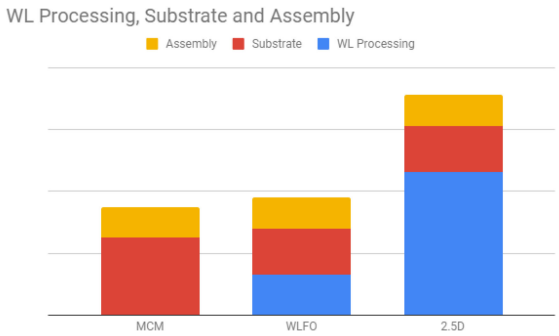

Figure 3. MCM versus 2.5D Packaging costs.

Packaging

BoW can be used with both traditional laminate and advanced packaging technologies. Figure 3 compares the relative costs of: 1) organic laminate with 6-2-6 substrate; 2) a simpler substrate where WLFO is used for dense interconnect; and 3) silicon interposer for an example package with four chiplets. The models use confidential manufacturing data available to the authors. These models are used to estimate the performance and package costs of combinations of BoW modes and packaging technology.

Figure 4 examines the tradeoffs of the BoW technology with various types of packaging. Bandwidth calculated for 5-, 16-, 32-Gb/s operation using BoW standard footprint at 130- and 55-μm pitch. Package costs estimated using proprietary cost models available to the authors, based upon expected build-up requirements, are shown in a relative value to each other. The cost models assigned more cost to more complex laminate wire counts, assuming they lead to more layers. For example, using the WLFO technology adds additional wafer-level processing cost, but reduces substrate layer counts, reducing overall cost in some cases.

Expected Performance

We believe reasonable estimates of performance can be extrapolated from interfaces with similar attributes that have been implemented and lab tested.

BoW basic mode is a simplified version of many DDR interfaces in production today, for example, DDR, LPDDR, and GDDR memory interfaces operate with very similar clocking and clock/data relationships at high baud rates (e.g., 16 GBd) from module to module. BoW has the advantage of lower skew signal routing on substrate and no discontinuities caused by additional package and board components, along with being implemented wholly in technologies better suited for IO and clocking than DRAM.

AQlink a die-2-die interface by Aquantia uses both terminated and bidirectional transmission lines as proposed in BoW BiDi. AQlink was implemented on 14-nm silicon and the measured performance data serves as a reference point[15] for BoW BiDi. Based on the simulation and silicon measurements on AQlink, BoW BiDi can comfortably operate at 16 GBd over a trace length of ~50 mm. The trace length limitation is caused by high-frequency signal attenuation, which can be addressed by equalizer circuits. Because the maximum BoW baud rate is significantly lower than other solutions (e.g., XSR at 56 GBd), it can use very simple and low-power equalizers relatively.

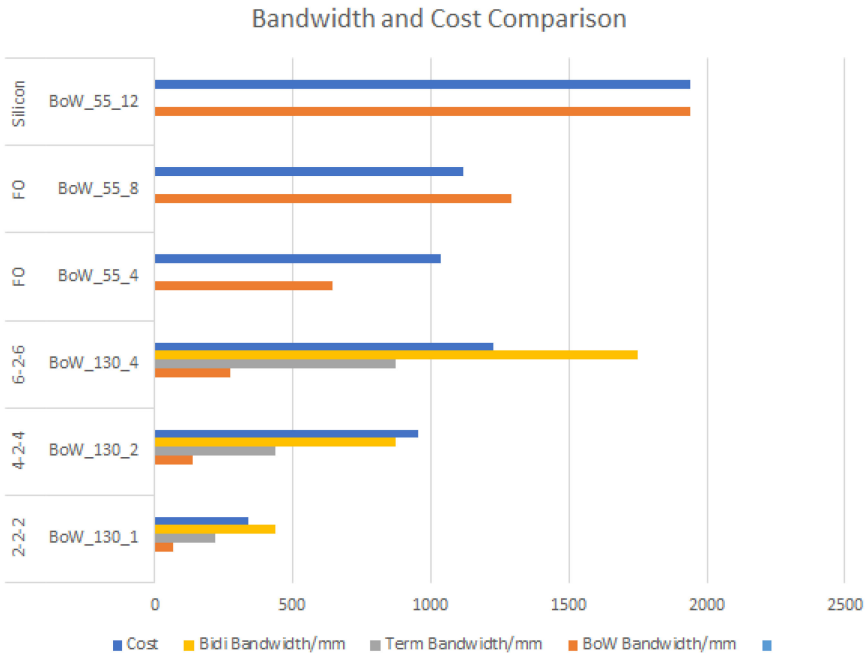

Figure 4. Packaging tradeoff with BoW.

Table 1. BoW parameters.

BoW Mode	Basic	BiDi
Data rate/bump	5 Gb/s	32 Gb/s
Bump type	micro bump	regular bump
Minimum pad pitch	55 μm	100 μm
Single supply voltage	0.7–0.9 V (±5%)	0.7–0.9V (±5%)
Power efficiency	0.4–0.7 pJ/bit	<0.70 pJ/bit
Substrate	Organic/WLFO/Silicon	Organic
Max Trace Length	10 mm	50 mm
Max Throughput/ Chip Edge	1.9 Tbps/mm (55 μm pad pitch interposer)	1.76 Tbps/mm (130 μm pad pitch, organic)
Power & Area/ Tbps (14 nm/ 0.7 V)	<600 mW, 1.01mm^2	<600 mW, 0.73 mm^2
BER (No FEC)	<1E–15	<1E–15
Latency (no FEC)	<3 ns	< 3 ns
ESD/CSM Requirement	250 V/50 V	250 V/50 V
Silicon Proof Point	multiple	GF 14 nm

Table 1 captures the parameters for BoW Basic with microbumps and BoW BiDi with regular bumps.

BoW BiDi can provide over 1.76-Tbps/mm bidirectional throughput per chip edge with a standard bump pitch of 130 μm. The same IP Core provides a maximum throughput of 0.88 Tbps/mm with BoW TD, and 0.22 Tbps/mm with BoW base.

Technology Discussion

Relative to parallel interfaces, BoW can achieve high bandwidth and power efficiency without requiring expensive silicon interposers or bridges. The different modes of BoW allow implementers to trade off IP complexity and package complexity options to find the right fit for their chiplet. Specifically, each version of BoW technology has the effect of reducing the bump density requirements, simplifying test support. This is evident when we compare four rows of BoW on organic with four or eight rows of BoW using fine pitch interconnect with WLFO technology, where fanout can provide a similar cost/bandwidth solution to four row implementations of BoW TD.

Relative to SerDes interfaces, BoW achieves bandwidth efficiency without the complexity and latency of multilevel PAM that needs FEC. By being easier to design, we believe BoW is more easily ported across multiple process nodes, a key requirement for the heterogeneous integration.

PAM4 signaling typically leads to undesired error rates (e.g. >1E-9) that mandates the use of forward error correction (FEC).[13,14] FEC not only increases the link power, but also increases the link latency. Based on silicon results using similar bidirectional interface (i.e., AQlink), BoW BiDi, using NRZ signaling, can operate at BER <1E–15, acceptable in most use cases. If better error rates are required, FEC codes can be used. A proposed Reed–Solomon[24] FEC code of RS (34,32,8) can correct one error within its RS frame of 272 bits (34 × 8 bits). In this case, an input BER = 1E–15 is equal to an input frame error rate (FER) of 2.72E–13. The proposed RS frame remains uncorrected after RS decoding if there are two or more random errors across the frame. The frame error probability of such event is $(2.72E–13)^2$ or FER = ~7.4E–26, which is equal to BER = ~2.7E–28. Such low BER is acceptable for all practical purposes, but the FEC incurs power and latency overhead.

In summary, BoW is area, power, and bandwidth efficient, offers a graceful tradeoff of design versus packaging costs, and combines the best attributes of parallel and serial interfaces.

SYSTEM INTEGRATION

Multi-chiplet products are usually motivated by one of the two following requirements:[25]

- Board-to-Chiplets: A need to reduce the footprint, power, and cost of a board product.
- Die-to-Chiplets: A large and/or complex design that needs to be partitioned to reduce manufacturing and/or design costs.

Multichiplet products require both physical connectivity and logical data transactions between the chiplets in a package.

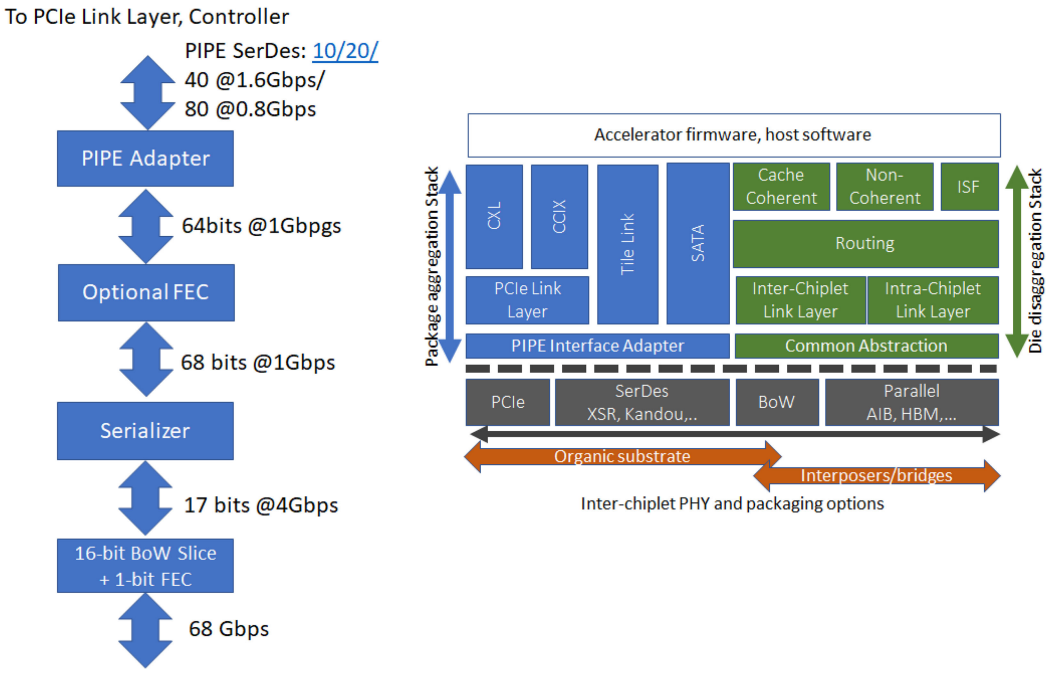

Figure 5. Open domain-specific architecture stack and PIPE adapter.

Domain-specific architectures have recently received renewed attention.[26] The ODSA aims to create a chiplet marketplace to enable domain-specific architectures to be created by integrating best-in-class chiplets from multiple vendors. The marketplace will be enabled by developing an open interface so chiplets from multiple vendors can interoperate easily. The ODSA stack aims to support open physical and logical data transactions between the chiplets in a package. Figure 5 shows the interchiplet networking stack under development.

We demonstrate the use of the BoW interface for an example design for the first case, board-to-chiplets. In this approach, the PCIe transactions between chiplets are executed over the interchiplet BoW PHY, rather than the long range PCIe PHY.

Pipe Interface Adapter

The PHY interface for PCI Express (PIPE)[27] is an open standard interface defined between PHY physical coding sublayer and media access layer (MAC). The PIPE interface serves as an abstraction layer between the PHY implementation and higher layers of the interface.

If an interchiplet PHY supports the PIPE interface through an adapter, the MAC and transaction layers of the PCIe protocol can be run over the interchiplet link. (The adapter will match the bit-width the data rates of the PIPE interface to the bit-width and data rates of the PHY.) This implies two chiplets connected by PHYs with PIPE adapters can use the PCIe protocol for data transactions, as used in board-to-chiplet designs, as well as any protocols that use the PCIe for data transport.

The use of a PIPE adapter for interchiplet links was first proposed by Kandou corporation for its USR SerDes.[28] More recently, Intel announced support for a low-power mode for interchiplet (and intrapackage) PCIe links on PCIe PHYs.[27]

Figure 5 shows the functionality required by an adapter. The adapter maps the interface data to a 16-bit BoW Turbo slice. The PIPE specification defines two types of interfaces, parallel and SerDes. Figure 5 shows the clock rates for PIPE adapter that maps PCIe Gen 4 lanes to a 16-bit Turbo BoW slice through a 40-bit SerDes interface. At the data rates shown, a single BoW Basic/Turbo slice can transport 4/32 PCIe Gen 4 Lanes. With this adapter, a design can potentially use a commercial controller[29] to execute PCIe transactions over a BoW interface.

System-Level Impact

The ODSA is building a proof-of-concept (PoC) multichiplet product with chiplets from

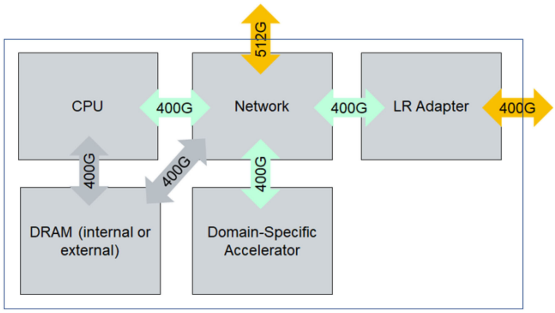

Figure 6. ODSA PoC prototype.

multiple vendors. The first prototype will use die produced to be standalone product as chiplets. A block diagram of the reference design used in the PoC is shown in Figure 6. The PCIe Gen 3 protocol is the logical interface between the components, with the exception of the network I/O. The reference design targets SmartNIC and network storage use cases.

The second-generation implementation of the ODSA reference architecture will use a low-power interchiplet PHY. Figure 6 shows the estimated internal bandwidth requirements for the reference design. The 1:1 ratio for bandwidth in links between significant components and 1:2 ratio for bandwidth to memory is consistent with designs for two use cases:

- In the Google TPU accelerator, the host-accelerator bandwidth is 14 GB/s. Correspondingly, the link between the unified buffer and the host interface is 10 GB/s, the memory bandwidth is 30 GB/s.[30]
- In networking applications, a minimum size 40-Byte packet (which occupies 64 Bytes of wire bandwidth) will result in accessing a 20-Byte 5-tuple from memory in IPv4 networks.

The BoW interfaces can be used to support a board-to-chiplets use case. A BoW interface with a PIPE adapter can support PCIe Gen 4 for interchiplet communication. This change will require new die that implement the BoW interface, but is transparent to the application software. Table 2 estimates the total power savings from using the BoW interface for PCIe Gen 4 transactions instead of a traditional PCIe PHY[31] in the ODSA reference architecture.

- Arrows in green are interchiplet PCI links, rounded up to 512 Gb/s to estimate power costs.
- Arrows in gray are links to memory, not included in the estimate, though they can also be BoW interfaces.
- Arrows in yellow are off-package interfaces.

Table 2. BoW system benefits.

Interface parameters				
PHY	PCIe	BoW Basic	BoW TD	BoW BiDi
Trans. Protocol	PCIe4	PCIe4	PCIe4	PCIe4
pJ/bit	7.5[31]	0.6(est)	0.7(est)	0.6(est)
Cost of 512 Gb/s interface (2 × 16 PCIe Gen4 lanes, Tx = Rx = 512 Gb/s)				
Power	3.84 W	0.3 W	0.35 W	0.3 W
Bump count		416	104	54
Area sq mm.	3.9	5.1	1.8	0.9
Impact on PoC (3 × 512 Gb/s interfaces)				
Total power	11.5 W	0.9 W	1.05 W	0.9 W

CONCLUSION

We proposed a new open BoW interface for inter-Chiplet communication. The basic interface is derived from the HBM specification. The Terminated and BiDirectional modes increase the speed of the basic interface by 4× and 8×. The BoW combines the process portability of parallel interfaces with the easy packaging attributes of serial interfaces. The definition offers a tradeoff—BoW allows designers to start with a basic interface and add either more complex IP development and/or more complex packaging technology to maximize edge and substrate data bandwidth. Our next step with this interface is to build a test chip, potentially with a PIPE adapter and a commercial PCIe controller.

ACKNOWLEDGMENT

The authors would like to thank B. Bahador from Eliyanpro, Inc. This work was performed while Bapi Vinnakota was with Netronome.

REFERENCES

1. IEEE Electronics Packaging Society Heterogeneous Integration Roadmap 2019 Edition, [Online]. Available: https://eps.ieee.org/technology/heterogeneous-integration-roadmap/2019-edition.html

2. B. Bayraktaroglu, "Heterogeneous integration technology," AFRL-RY-WP-TR-2017-0168, Aug. 2017.
3. A. Olofsson *et al.*, "Enabling high-performance heterogeneous integration via interface standards," in *Proc. IP Reuse, Modular Des. Int. Symp. Microelectron.*, vol. 2018, pp. 000246–000251, 2018.
4. J. H. Lau, "Recent advances and trends in heterogeneous integrations," *J. Microelectron. Electron. Packag.*, vol. 16, pp. 45–77, 2019.
5. L. T. Su *et al.*, "Multi-chip technologies to unleash computing performance gains over the next decade," in *Proc. IEEE Int. Electron Devices Meeting*, Dec. 2017, pp. 1.1.1–1.1.8.
6. R. Viswanath *et al.*, "Heterogeneous SoC integration with EMIB," in *Proc. IEEE Elect. Design Adv. Packag. Syst. Symp.*, Dec. 2018.
7. Marvell Corporation, "MoChi architecture," Tech. Rep., [Online]. Available: http://www.marvell.com/architecture/mochi/
8. G. Singh *et al.*, "Xilinx 16 nm datacenter device family with in-package HBM and CCIX interconnect," *HotChips*, 2017.
9. B. Bailey, "The impact of Moore's law ending," Oct. 18, 2018. [Online]. Available: https://semiengineering.com/the-impact-of-moores-law-ending/
10. D. Stow *et al.*, "Cost-effective design of scalable high-performance systems using active and passive interposers," in *Proc. 36th IEEE/ACM Int. Conf. Comput.-Aided Design*, Nov. 2017, pp. 728–735.
11. Intel AIB Bus, 2018. [Online]. Available: https://github.com/intel/aib-phy-hardware
12. JEDEC, "High bandwidth memory (HBM) DRAM," JESD235B, 2018.
13. Kandou Bus, 2018. [Online]. Available: https://kandou.com/technology/
14. OIF, Common Electrical Interface—112G—XSR, 2018.
15. G. Taylor, R. Farjad, and B. Vinnakota, "High capacity on-package physical link considerations," *Hot Interconnect*, 2019.
16. Open Domain-Specific Architecture, 2019. [Online]. Available: https://www.opencompute.org/wiki/Server/ODSA
17. ODSA Workgroup, "The open domain-specific architecture: A chiplet-based open architecture," [Online]. Available: https://www.netronome.com/m/documents/WP_ODSA_Open_Accelerator_Architecture.pdf
18. J. A. Lim *et al.*, "Fan-out wafer level eWLB technology as an advanced system-inpackage solution," in *Proc. IWLP*, San Jose, CA, USA, Oct. 2017.
19. K.-L.-Suk *et al.*, "Low cost Si-less RDL interposer package for high performance computing applications," in *Proc. IEEE 68th Electron. Compon. Technol. Conf.*, 2018, pp. 64–69.
20. Y.-L.- Chuang *et al.*, "Unified methodology for heterogeneous integration with CoWoS technology," in *Proc. IEEE 63rd Electron. Compon. Technol. Conf.*, May 2013, pp. 852–859.
21. D.U. Lee *et al.*, "A 1.2V 8 Gb 8-channel 128 GB/s high-bandwidth memory (HBM) stacked DRAM with effective microbump I/O test methods using 29nm process and TSV," in *Proc. IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers*, 2014, pp. 432–433.
22. R. Farjad and B. Vinnakota, "A bunch of wires (BoW) interface for inter-chiplet communication," in *Proc. Hot Interconnect*, 2019.
23. M. Kuemerle, R. Farjad, and B. Vinnakota, "Bunch of Wires Interface Proposal, v0.7," 2019. [Online]. Available: https://www.opencompute.org/wiki/Server/ODSA
24. Reed Solomon Error Correction Intro, [Online]. Available: https://en.wikipedia.org/wiki/Reed%E2%80%93Solomon_error_correction
25. R. Nagisetty, "Intel ODSA Workshop," Jun. 2019. [Online]. Available: https://146a55aca6f00848c565-a7635525d40ac1c70300198708936b4e.ssl.cf1.rackcdn.com/images/be20ea9409cc558936fa2623c5222792e8118c69.pdf
26. J. L. Hennessey and D. A. Patterson, "A new golden age for computer architecture," *Commun. ACM*, vol. 62, no. 2, pp. 48–60, Feb. 2019.
27. Intel, "PHY interface for the PCI express, SATA, USB 3.1, DisplayPort and converged IO architectures, Version 5.2," 2019. [Online]. Available: https://www.intel.com/content/dam/www/public/us/en/documents/white-papers/phy-interface-pci-express-sata-usb30-architectures-3-1.pdf
28. Kandou Corp, "Kandou bus document: 16-lane femtoserdes PIPE Adapter," 2019.
29. PCIe PIPE 4.4.1: Enabler for PCIe Gen4, 2018. [Online]. Available: https://blogs.synopsys.com/vip-central/2018/01/17/pcie-pipe-4-4-1-enabler-for-pcie-gen4
30. "An in-depth look at Google's First Tensor Processing Unit," 2017. [Online]. Available: https://cloud.google.com/blog/products/gcp/an-in-depth-look-at-googles-first-tensor-processing-unit-tpu
31. S. Li *et al.*, "A power and area efficient 2.5-16 Gbps gen 4 PCIe PHY in 10nm FinFET CMOS," in *Proc. IEEE Asian Solid-State Circuits Conf.*, Nov. 2018, pp. 5–8.

Ramin Farjadrad is currently a CTO & VP of Networking/PHYs at Marvell Semiconductor, in charge of developing multi-GHz connectivity solutions for autonomous vehicles, hyperscale data centers, and enterprise networks, and the Chairman of technical committee at Networking for Autonomous Vehicles (NAV) Alliance. He proposed signaling schemes adopted as IEEE Standard for 10 Gbps Automotive Ethernet and Multi-Gbps Enterprise Ethernet. He has domain expertise in high-speed communication circuits and systems, signal processing/coding, optimized mixed-mode architectures. He is author of 100+ granted/pending U.S. patents. He received the MSc/PhD degrees in electrical engineering and computer science from Stanford University. Contact him at farjad@gmail.com.

Mark Kuemerle is a fellow of Integrated Systems with Avera Semiconductor. He has broad experience in a wide range of large complex infrastructure ASICs, significant IP and expertise in power efficiency methodologies, complex interconnect and packaging, and integrated memory structures. He has ben PHY/Interface workstream leader in the ODSA, member and contributor to various IEEE and other industry publications, other previous industry group leadership includes HMC Consortium protocol committee. He received the Master of Science degree from Case Western Reserve University. Contact him at mark.kuemerle@globalfoundries.com.

Bapi Vinnakota is currently a consultant with Talumbra Services. While at Netronome, he started and is the leader for the Open Domain-Specific Architecture, a chiplet-based open architecture project in the Open Compute Project. After a Ph.D. at Princeton, he taught at the University of Minnesota, where he received an NSF CAREER Award and three IBM Faculty Development Awards. He joined Intel through an acquisition and was an architect of a VoIP flow processor and worked in networking technology, and incubated a networking SaaS product. At Netronome, he created and ran open-nfp.org, a service for research in networking. He is the corresponding author of this article. Contact him at bapi.vinnakota@gmail.com.