

The Bunch of Wires (BoW) – An Open-Source Physical Interface Enabling Chiplet Architectures

Lei Shan

Ampere Computing, Santa Clara, CA

shanlei@amperecomputing.com

Abstract

As on-chip performance scaling slows down due to semiconductor manufacturing technology and yield limitations, in-package system integration combining with the concept of “chiplet” becomes a widely adopted solution especially with the growth of system functionality and heterogeneous architectures. Accompanied with this emerging trend, various interconnect standards are being developed to meet the demands of high I/O bandwidth with low-power and low-latency, so that system performance shall not be compromised while cost is significantly reduced by “reassembling” chiplets into an equivalent gigantic monolithic silicon chip.

Historically, serial data links have been employed for high-speed data transmissions, such as PCIe®, Gen-Z, Omni-Path, and NVLink™, just to name a few. However, due to serialization/deserialization processes and the overhead on top of the data payload, they tend to introduce more latency and power than parallel links at the same total bandwidth. On the other hand, parallel data buses running at lower speed are more inherently suitable for the chip-to-chip interconnections, but in most cases, they require higher I/O pin count. Fortunately, the latter challenge may be addressed by today’s advanced packaging technologies including silicon/organic interposers, embedded/overlay bridges, and wafer-level fanout, etc.

To help establish a healthy ecosystem, most interconnect standards targeting chiplet applications are initiated as open-source collaborations, such as the Advanced Interface Bus (AIB), the recently released Universal Chiplet Interconnect Express (UCIe), and the one we are going to focus on in this article, the Bunch of Wires (BoW).

Overview

The Bunch of Wires (BoW) is a simple, open, and interoperable physical interface between two chiplets or chip-scale-packages (CSP) in a common package. The standard was initiated by the Open Domain Specific Architecture (ODSA) group of the Open Compute Project (OCP) organization that was originally founded to share designs of data center products and best practices among companies, including Ampere Computing, Alibaba Group, AMD, ARM, Cisco, Dell, Fidelity, Goldman Sachs, Google, Hewlett Packard Enterprise, IBM, Intel, Lenovo, Meta, Microsoft, Nokia, NVIDIA, Rackspace, Seagate Technology, and Wiyynn, etc. Because of the close collaboration among the data center leading companies, the BoW standard covers a broad range of interests and concerns particularly associated with the cloud as well as edge computing applications.

Even though there are “short” and “long” channel versions, the use of BoW is mainly confined to connecting die placed close to one another within the same package. In this environment, signal attenuation is low, and the interconnect can be simple with the least overhead. The definition of the BoW interface aims to meet the design objectives, including low implementation cost, portability across IC process nodes, flexibility to support various packaging technologies and bump pitches, low power, low latency, and high throughput density, etc. According to the ODSA technical work group, the BoW interface provides several key advantages for chiplet-based systems, such as higher data rates per pin than

existing parallel standards, implementability in legacy technologies (either low-cost laminates or higher-density silicon-based interposers), easier design effort than a traditional SerDes link, and compatibility with mixed bump-pitch packaging scenarios. Compared to SerDes, BoW uses a lower data rate per wire, so it requires more wires. However, the lower data rates allow the use of single-ended signaling and denser wire packing. As a result, BoW can take the advantages of multiple wiring layers in laminates as well as the high wire density in advanced packaging solutions.

Physical Layer (PHY)

As shown in Figure 1, the BoW PHY is defined as a single unidirectional slice. Multiple slices are combined to create links of the desired throughput. A link may be symmetric, asymmetric, or unidirectional. The BoW PHYs between two dies are physically connected through wires on a substrate or interposer. However, a BoW PHY is not designed for such applications as off-package interfaces and on-chip data buses.

A BoW PHY slice either transmits or receives 16 bits of data between two die. The BoW is a source-synchronous PHY and each transmitting PHY slice transmits a complementary clock signal CLK+ and CLK- with the data. A BoW PHY optionally has two additional wires designated FEC (Forward Error Correction) and AUX, for other optional functions such as Data Bus Inversion (DBI).

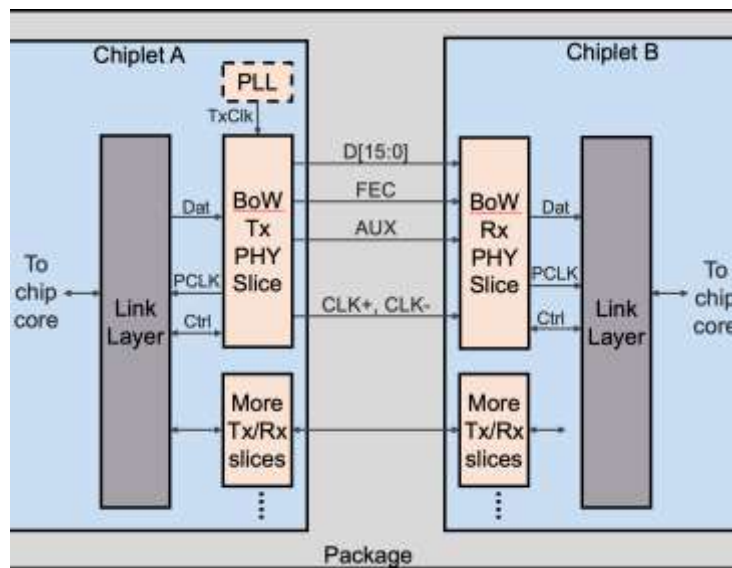


Figure 1. BoW Physical Layer

A BoW PHY must be operable in one of the BoW Modes listed in ascending order in Table 1. A BoW Mode defines the speed of clock and data of the PHY on the die-to-die wires. In all modes, the data must be clocked DDR: the chip-to-chip data wire bit rate is double the clock wire frequency. All BoW interfaces faster than BoW-64 should also be able to support BoW-64. Supporting rates other than the defined four modes is an implementation choice. Figure 2 shows the tradeoff between package, data rate, termination, and reach. Source-terminated BoW on laminate allows a longer reach than advanced packaging, but the wider design rules in laminate means that both cases are barely able to reach 8Gbps/wire. A double-terminated link offers longer distances and higher rates but requires a more complicated receiver design.

Table 1. BoW Modes

BoW Mode	Slice Data Rate Gbps	Wire Bit Rate Gbps/wire	TxCk GHz
BoW-32	32	2	1
BoW-64	64	4	2
BoW-128	128	8	4
BoW-256	256	16	8

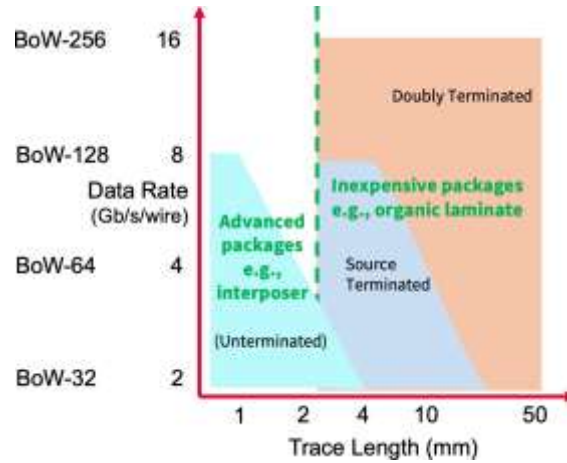


Figure 2. BoW Data Rate vs. Channel Length

Configuration

A BoW link between two chiplets is made up of wires, slices, and stacks as seen in Figure 3(a). A slice is the basic unit of a BoW PHY, which contains 18 or 20 signal bumps, i.e., 2 bumps for the differential clock and 16 single-ended data bumps, plus 2 optional bumps for AUX and FEC. A stack is composed of one or more slices to achieve more interconnections and therefore higher bandwidth. The slice positions are designated A, B, C, D, etc. starting with the slice closest to the edge of the chip. A link from one chiplet to another is composed of one or more stacks placed along the chip edge, which may be configured with equal numbers of RX and TX slices or asymmetric arrangement. An example of laying out a 4-slice bidirectional link is shown in Figure 3(b) in an 8-2-8 laminate substrate. In advanced packaging technologies, the shorter wire length and higher wire resistance suggests the use of non-controlled-impedance wires and unterminated transmitters and receivers. The smaller wire and space dimensions may allow the wires for multiple slices to be interleaved on a single wiring layer. The wire order within each slice must be maintained, even if interleaving with other slices is used.

Specifications

As mentioned earlier in Table 1, BoW interface may operate at DDR data-rate ranging from 2 to 16 Gbps/wire with 1 to 8 GHz synchronized clock. To retain some degree of design flexibility on each of the TX and RX, a bound is always placed on the maximum deterministic error and on the maximum total error budget at the target error rate of $1e-15$. All BoW links must support signaling based on a 0.75V “I/O voltage”. The simplest implementation is to provide a 0.75V supply voltage to the BoW VDD bumps, but the supply voltage may be different from the I/O voltage as long as the signal voltages meet the specification.

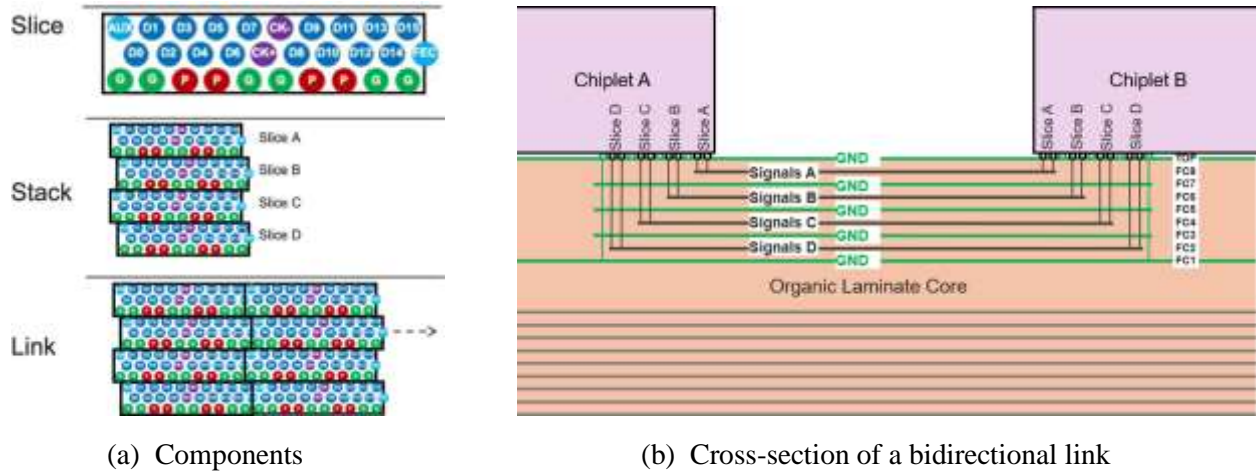


Figure 3. BoW Link Configurations

The BoW does not place any direct requirements on characteristics such as channel loss or crosstalk. Instead, BoW channels are considered conforming if they can achieve the bit error rate of $1e-15$ in conjunction with reference transmitters and receivers that meet all of the requirements mentioned above. Therefore, the insertion loss and crosstalk limits shown in Figure 4 are quoted as recommendations.

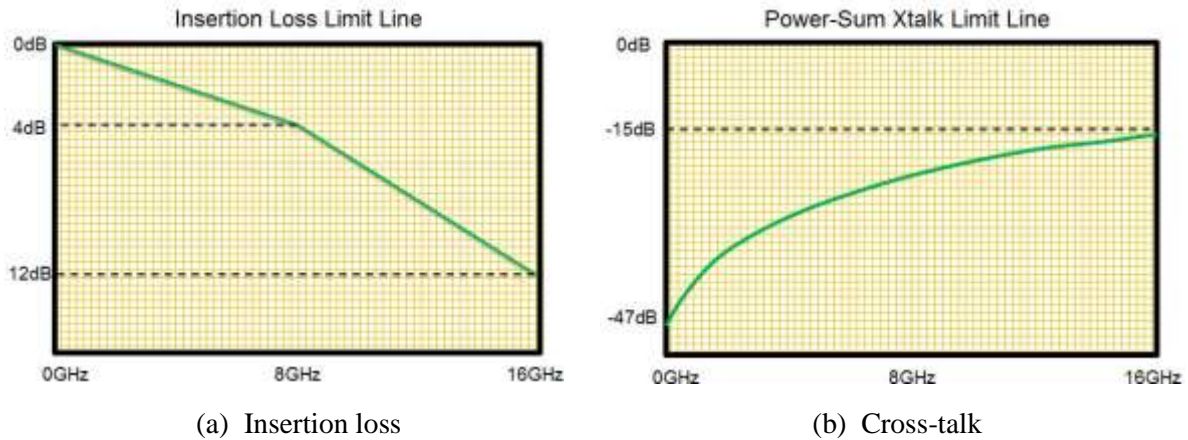


Figure 4. BoW Link Recommendations

Comparisons

Based on public sources through the internet, a brief summary of three die-to-die I/O standards is listed in Table 2. Since the actual channel performance varies with the specific implementations, the cited numbers are for information only, and therefore should be used with caution.

Table 2. Comparisons of Die-to-Die I/O Standards

	BoW	AIB 2.0	UCIe
Data Rate, Gbps	2-16	<6.4	4-32
Bus width	16	16	16 or 64
Channel Reach, mm	1-50	<10	2-25
Power Efficiency, pJ/bit	0.5-1	0.5	0.25-0.5
Latency, nsec	<5	<4	<2
Bandwidth per mm, Gbps	100-1000	<1638	165-1350

