Concurrent Multipoint-to-Multipoint Communication on Interposer Channels

Lejie Lu, Richard Afoakwa, Michael Huang, and Hui Wu
Dept. of Electrical and Computer Engineering, University of Rochester, Rochester, NY 14627 USA
{lejie.lu, richard.afoakwa, michael.huang, hui.wu}@rochester.edu

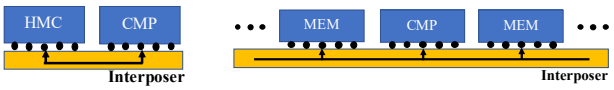
***Abstract*—Chip-to-chip communication for next generation computing will require larger bandwidth density to support ever increasing data traffic between processors, memories and I/O. 3-D integration enables a large number of processor and memory chips to be densely packed on an interposer with fine-pitch interconnect lanes. Advanced signaling techniques such as pulse amplitude modulation (PAM) can be employed to improve bandwidth per lane. Most recent work on interposer-based chip-to-chip interconnects focus primarily on point-to-point serial links. Without adding costly routers, these designs will severely limit the overall system level concurrency. In this paper, we propose an ultrahigh-speed multipoint-to-multipoint link design for interposer channels, which supports PAM signaling. Each node on the link can send, receive, drop, or relay data at line rate without complex routing. This design enables splitting the physical link into segments, and allows multicast/broadcast. A proof-of-concept system prototype with up to 16 nodes integrated on a silicon interposer with up to 22-mm node spacing is designed and evaluated using circuit and system simulations. The PAM-4 transceiver and link interface circuits at each node are implemented using a standard 130-nm SiGe BiCMOS technology. The transceiver can achieve a data rate of 40-Gb/s/lane, with channel loss of -3.5 dB per segment at Nyquist frequency, and energy efficiency between 1.29-pJ/b between two neighboring nodes or 0.21-pJ/b more per additional nodes. Using a cycle-level system simulation, such a high-concurrency communication fabric can improve overall performance between 2% to 18% over baseline.**

***Index Terms*—Interposer, High-speed links, Pulse-Amplitude Modulation**

I. Introduction

Next generation high performance computers (HPCs) will continue to incorporate system-in-a-package (SiP) technologies to increase bandwidth density, reduce latency, and save power [1]. A high-bandwidth inter-chip communication infrastructure is therefore critical. To support such high-bandwidth interconnects, designers can either increase the data rate per lane, or the number of lanes per link [2]. Data rates of 56 Gb/s and above have been demonstrated for chip-to-chip connections on PCB or board-to-board with cable interconnects [3], [4] These techniques, however, only help to improve the signaling speed in a single lane. Some recent work focus on designing high density I/O channels on interposers [2], [5], [6]. Fine-pitch interconnection channels, typically implemented as transmission lines, can be fabricated on these silicon or organic substrates. Fig. 1a shows such an example: a point-to-point inter-chip link connects a hybrid memory cube (HMC) and a chip multi-processor (CMP) integrated on an interposer. Still, in a system with many chips, relying solely on point-to-point links for all communication is simply impractical. While network-on-chip (NoC) is the default system-level solution, NoC can bring undesirable latency, energy, and complexity overheads. The previously proposed multi-drop link can be a powerful component in reducing these overheads [7], [8]. However, in earlier designs (Fig. 1b), such a bus can only support one single transmission at a time even when the transmission is only utilizing a small subset or segment of the channels. In this work, we propose a multipoint-to-multipoint interconnect that permits multiple concurrent transmissions, which can significantly increasing the system's effective throughput without adding too much cost.

978-1-7281-2954-9/19/$31.00 ©2019 IEEE



(a) Point-to-point (b) Multi-drop

Fig. 1: Chip-to-chip interconnects on interposers.

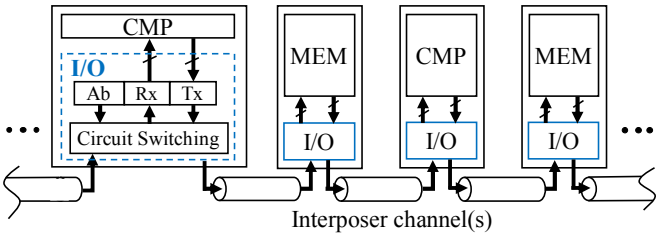


Fig. 2: The proposed multipoint-to-multipoint link.

This paper is organized as follows: in Section II, we will introduce the new interconnect, including the link, channel, and circuit designs and performance. We perform the system level evaluation in Section III to demonstrate its benefit for overall system concurrency.

II. Link, Channel, and Circuit Design

A. Overview

In this work, we propose a multipoint-to-multipoint interconnect for concurrent communication on interposer channels. As shown in Fig. 2, all interconnect nodes (processors and memory chips) communicate with each other through a series

of high-density interposer channels. Each node incorporates an I/O interface with simple networking capabilities: it can transmit (add), receive (drop), or relay (pass) data to/from the channel. Note that the I/O interface does not require full routing support as in a conventional NoC. There are multiple parallel lanes in a channel (Fig. 2 shows only one lane for clarity). Each lane connects to separate I/O circuitry at each node, which is independently controlled. There are two high-level link operations: ① a *direct* link is one connecting two adjacent neighbors; and ② a *relay* link connects two far apart nodes, and needs relay services by intermediate nodes. A link is unidirectional: each node receives data from its neighbor to the left, transmits data to its neighbor to the right, and relays between those two from left to right; or the other way around. A pair of these unidirectional links in opposite directions form a bidirectional one.

The channel characteristics depend transmission distance on the interposer. Here we assume that the interposer is up to 10 cm long, and the chip size varies from 1 to 3 cm. With 2-mm spacing between chips, the channel length is 12 to 32 mm, assuming the I/O ports are located on the same chip edge. Although the interconnect is shown here with the chips horizontally packed on the interposer, this design can be potentially applied to 3-D stacked systems.

For signaling in the link, we envision that highly spectral efficient pulse amplitude modulation (PAM) is employed to increase bit rate per lane as compared to conventional non-return-to-zero (NRZ) signaling. By keeping the baud rate relatively constant, channel loss and dispersion can remain manageable, and hence the interconnect will be more scalable. Transmission line based channels and analog-like high-speed I/O circuits ensure that the link can achieve tens of GBaud per lane. In the following discussion, we assume PAM-4 signaling at 20-GBaud (40 Gbs) per lane based on the prototype circuit design (see Section II).

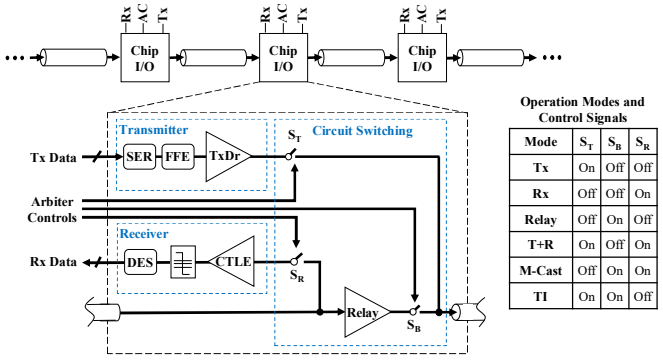


Fig. 3: I/O component interface and operation.

Fig. 3 shows the block diagram of the I/O interface (for one lane only). It consists of a serial link transceiver, a relay amplifier, and three switches controlled by an arbiter (not shown). The transmitter consists of a serializer (SER), feed-forward equalizer (FFE) and driver. The receiver consists of a continuous-time linear equalizer (CTLE), decision circuit, and deserializer (DES). The relay amplifier has built-in equalization to compensate dispersion, similar to CTLE in the receiver. The I/O interface has three basic operation mode: transmit, receive, and relay. There are also combined operation modes: (a) Full duplex operation with simultaneous transmit and receive; (b) Simultaneous receive and relay, which is useful for multicast/broadcast; (c) Time-interleaved transmit when the transmit data from this node is inserted into the relayed data stream. The switch controls for these modes are shown in the inset table in Fig. 3. In the following discussion, we focus on the basic operations of the I/O interface. The combined operations will be explored in our future work.

B. System Operation

Based on the link architecture and basic operation of the I/O interface, we can analyze the high-level latency characteristics of packet transmission using our proposed interconnect, and compare to point-to-point links. Our analysis covers a 4-node system. We assume a packet size of 72 B (64 B cache-line plus 8 B header), and link propagation delay of 1 compute cycle. One link consists of 4 lanes, each operating at 40 Gb/s. Given the processor clock frequency of 2.5 GHz, the link bandwidth is $4 * \frac{40\,Gb/s}{2.5\,GHz} = 64\,bits/cycle = 8\,B/cycle$. All values are base on design criteria discussed later. Based on payload size and link bandwidth, serialization is 9-compute cycles. Additionally, all links in this analysis are unidirectional.

1) Point-To-Point System: The timing diagram in Fig. 4 shows a simple contention-free payload transmission to a destination 3 nodes away. Setup (S) processing happens in cycle 1. The head flit (f_0) traverses the first link in cycle 2, and arrives at the destination in cycle 7. Remaining flits flow the in a pipeline order until the tail flit (f_8) arrives at the destination. There are total 15 compute cycles. Additionally, it must be noted that transmitting payloads to a neighbor (1 node away) would require 11 compute cycles. Essentially, the unloaded latency bounds in this example point-to-point transmission is 11 to 15 cycles.

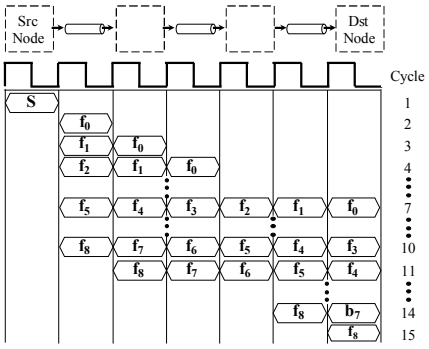


Fig. 4: Timing diagram of the point-to-point system.

2) Proposed Multipoint-to-Multipoint System: Fig. 5 shows the schematic and timing diagram of the 4-node system using our proposed interconnect. Our design utilizes an arbiter for

scheduling events between nodes. Compared to the point-to-point design, this design does need receive the packet and re-transmit it at each intermediate node. Data can be relayed *in analog* through all middle nodes and reach destination within a cycle. Our arbiter scheduling can be performed under two configurations, which we discuss below.

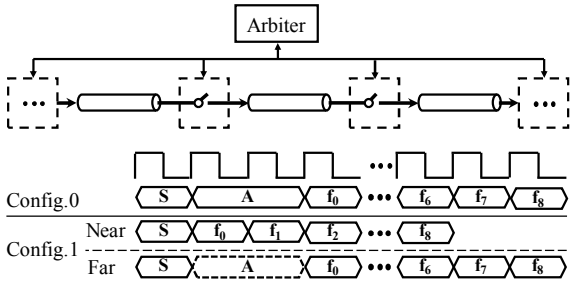


Fig. 5: Timing diagram of proposed multipoint-to-multipoint system.

Baseline: In this configuration, the arbiter is passive. It waits for request from nodes and performs scheduling. Each node needing to transmit first processes the payload (1-cycle), and subsequently signals a request to the arbiter. The arbiter schedules the request, and signals a grant to the requester, as well as a wakeup to the receiver. We assume a fixed 2-cycle arbitration overhead; half a cycle (each direction) for signal propagation to/from arbiter, and 1-cycle arbiter processing. The payload serializes in 9-cycles, and the unloaded node-to-node latency is 12-cycles.

Optimized: We optimize the baseline configuration to eliminate the fixed 2-cycle overhead as mush as possible. We achieve this by making the arbiter more active. Instead of waiting for nodes to place requests, the arbiter, by default, grants a standing permission to all nodes needing to send payloads to a neighbour (1-hop away). If a node has a neighbour-transmit grant, it simply processes the payload, signals the arbiter about the action, and transmits. Receiver circuitry are put into active listening mode in this configuration. The arbiter guarantees the section of the line for the transmitting node. The latency overhead is therefore a minimum of 10-cycles (1-cycle setup, 9-cycle serialization), and a maximum of 12-cycles for neighbor or non-neighbour transmit respectively.

Summary: Compared to point-to-point channels, our proposed multipoint design has the potential to lower overall communication latency, specifically with increasing node count, i.e. concurrency. A fully system level analysis can be found in Section III.

C. Interposer Channel Design

In this paper, we assume the channel is designed as a transmission line in the coplanar waveguide (CPW) structure, and fabricated on a silicon interposer using the 4 μm thick top metal layer. To suppress noise and interference, differential signaling is adopted. To isolate the transmission line from the substrate and improve crosstalk, ground shields are added below. The differential CPW transmission line is shown in Fig. 6a.

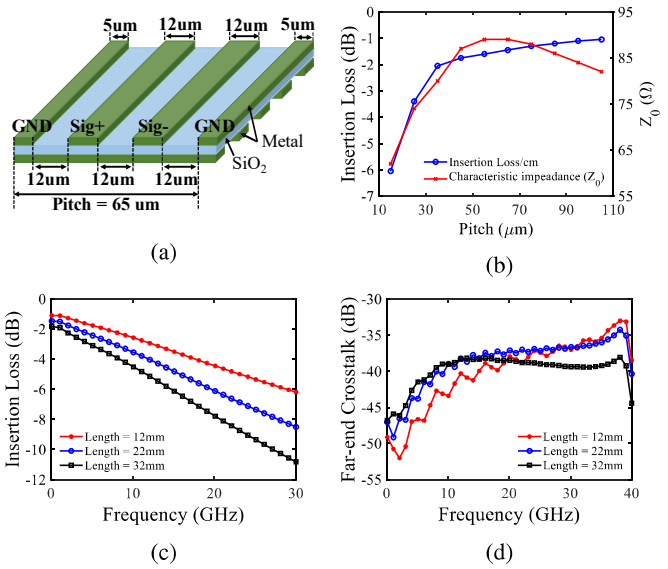


Fig. 6: Channel design based on EM simulations: (a) A differential coplanar waveguide with ground shields on a silicon interposer. (b) Simulated loss and characteristic impedance with different pitch size; (c) Channel loss at different distances; (d) Far-end crosstalk.

To minimize the channel loss and keep the characteristic impedance (Z_0) relatively high, we optimized the dimension of the CPW channel using electromagnetic (EM) simulations. As shown in Fig. 6b, the loss decreases fast with larger pitch size (wider signal lines) initially and then the effect slows down after 30-μm pitch size. To improve the bandwidth-density and energy efficiency, a relatively small pitch size (65 μm) is chosen, which results in a relatively high characteristic impedance (90 Ω) The channel loss at different distances is shown in Fig. 6c. At Nyquist rate of 10 GHz, the channel loss is -2.57, -3.54, and -4.5 dB at 12, 22, and 32 mm. Fig. 6d shows far-end crosstalk (FEXT) between two adjacent lanes are below -35 dB. Therefore, coupling between two adjacent lanes is not a major concern compared to the dominant channel loss.

D. Circuit Design

As a proof-of-concept demonstration, we design a prototype I/O interface using a 130-nm SiGe BiCMOS technology, which will be used for system evaluation later. The transceiver is designed to support up to 40-Gb/s data rate per lane with PAM-4 signaling.

At the transmitter, as shown in Fig. 8, the 2.5-Gb/s input digital data is serialized to four 10-Gb/s signals. After FFE, two 20-Gb/s LSB and MSB signals (and their corresponding delayed versions) are generated and fed into the PAM-4 driver

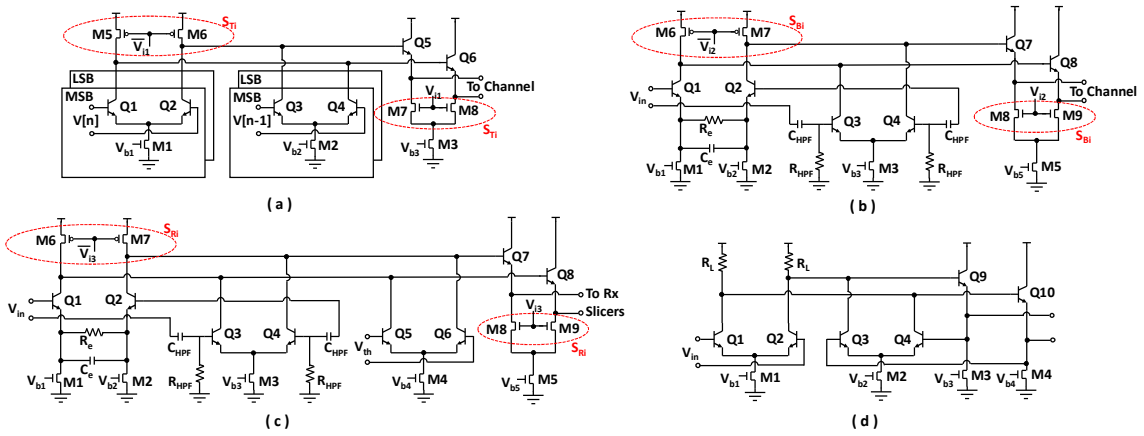


Fig. 7: Schematics of (a) transmitter driver; (b) relay CTLE; (c) receiver CTLE; (d) one of latch-based slicers.

for additional 1-tap FFE equalization. Then, the PAM-4 signal is modulated by last stage transmitter driver. At the receiver, the distorted signal from lossy channel is compensated by CTLE, then sampled by slicers with three different thresholds. Finally, digital data are recovered after decoding and deserialization.

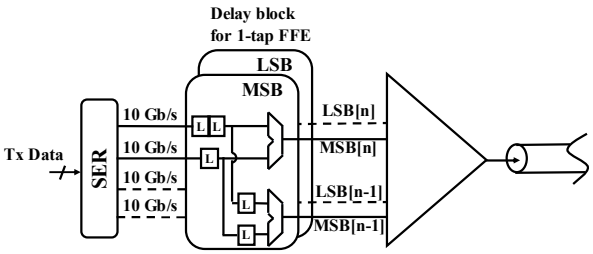


Fig. 8: Transmitter component of transceiver.

Fig. 7 shows circuits of some key building blocks. In [9]–[11], high-swing voltage mode drivers are used to support a single long distance channel with higher loss. Instead of connecting two nodes with one long distance channel, our segmented channels design for multipoint-to-multipoint application allows us to compensate several shorter channels with lower loss. The equalization is easier with multiple distributed CTLEs compared to equalizing high loss channel with a single CTLE.

In our design, we remove nonlinearity effect in upcoming stages by using a low-swing current-mode logic (CML) driver adopted as shown in Fig. 7a. It is split into LSB and MSB segments for PAM-4 modulation, dumping 1X and 2X tail currents, respectively. And additional branches (Q3,Q4) generate the 1-tap FFE to cancel the first post-cursor ISI. This provides 1-4 dB equalization capability based on channel characteristic here. The transmitter isolation switch S_{Ti} is implemented by transistors (M5,M6) and (M7,M8). Moreover, the relay switch S_{Bi} is merged into the relay CTLE as shown in Fig. 7b. Similarly, the receiver switch S_{Ri} is merged into Rx CTLE as shown in Fig. 7c. The CTLE equalization strength is controlled by R_e and C_e, which is set based on the channel characteristics. Fig. 7d shows the latch-based slicer. Three of them are necessary to sample the PAM-4 signal.

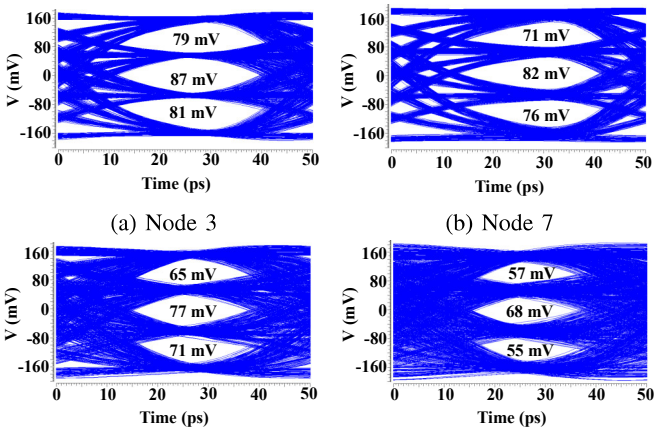


Fig. 9: Eye diagram of the received data (after Rx CTLE) when node 0 transmits.

In our proposed system, nonlinearity is mainly caused by gain compression at each relay stage. And this effect will accumulate with increasing number of stages. Let's define the gain compression factor α as the ratio of the compressed and uncompressed gain. To calculate the different eye amplitude in PAM-4 eye diagram, we assume the differential peak-to-peak amplitude of relay input signal is A_{in}, and the gain of the relay, G, is set to compensate the attenuation of the channel. Also, we assume the middle eye amplitude after the relay, V_m, is not distorted due to the relatively small input signal. So, V_m can be expressed as $V_m = \frac{1}{3}A_{in}G$, After N relay stages, due to the accumulated gain compression, the last stage minimum eye differential peak-to-peak amplitude V_l is changed to:

$$V_l = \frac{1}{2}A_{in}G\alpha^N - \frac{1}{6}A_{in}G \tag{1}$$

Then, the Ratio-of-Level-Mismatch (RLM) can be estimated

based on the above eye amplitude:

$$RLM = \frac{3}{2} - \frac{1}{2\alpha^N} \quad (2)$$

In our proposed interconnect, we assume up to 16 nodes. In order to keep the last stage RLM value above 90%, according to Eqn. 2, the gain compression factor, α, should be at least 98%. Based on the simulation, the relay CTLE output level falls below its ideal level by 98% at the 50 mV input point. This indicates that the maximum amplitude of CTLE input signal should be 50 mV to meet the required linearity specification in this design.

The overall system simulation has been performed in Cadence Custom IC Design tools. Fig. 9-(a) to (d) shows the receiver side equalized eye diagrams when the data is transmitted from node-0 to up to node-15. After the last stage CTLE, signal is amplified and compensated, so that we get 55mV minimum eye height and 91.6% RLM value in the worst case, which ensures 10^{-15} BER criterion. Table. I shows the break down power consumption and energy efficiency for different configurations. Our design is based on SiGe BiCMOS technology. If CMOS technoloy is applied, the energy efficiency will be further improved.

TABLE I: Component power (in mW) and channel energy efficiency (in pJ/b). Power of relay CTLE is 8.4 mW.

Tx	Power	Rx	Power	Hops	Energy
Mux	9	Rx CTLE	6.6	1	1.29
FFE	8	Slicer	6.4	2	1.5
Driver	12.7	De-Mux	9	3	1.71
Total	**29.7**	**Total**	**22**	**+1**	**+0.21**

III. System Level Evaluation

A. Experimental Methodology

We evaluate our multipoint system architecture using a modified version of the cycle accurate GEM5 simulator [12]. We assume 28 nm technology for our design, and target the ARM Cortex-A75 core for our evaluation. We base our hardware configurations on this core family, and use a total of 64 such cores. Our supply is 1 V at 2.5 GHz frequency.

We target the 3D Hybrid Memory Cube (HMC) [1] for our memory sub-system. We model this using HMCSim which we port into GEM5. Memory hardware configuration and timing parameters are derived from HMC spec. 2.1. We use 16 GB per cube, with 32 vaults, 8 partitions, and 32 MB DRAM devices. The timings are as follows; t_{CK} = 0.8 ns, t_{RAS} = 21.6 ns, $t_{RCD/CAS}$ = 10.2 ns, t_{WR} = 8 ns, t_{RP} = 7.7 ns

We use applications from diverse multi-threaded suites, to cover thread-level parallelism and memory intensity; graph [13], map-reduce [14], and parsec [15]. For graph and map-reduced, we use the Stanford large network dataset collection. And for parsec, we use large-sized input set.

In our experiments, we test for performance dependence on concurrency. We use varying node sizes, from 4 to 64, in powers of 2. A *node* is a single memory or processor chip. The processor chip has varying core count. But me maintain a consistent total count of 64. Each system under test is composed of multiple interleaved processor and memory nodes. We developed detailed interconnection model for our proposed multipoint system, as well as conventional point-to-point system. Additionally, we add an *Ideal*, contention-free, 1-cycle latency topology to evaluate performance upper bound.

B. Performance and Energy

Fig. 10 shows a summary of performance breakdown. For brevity, we only show the geometric means across all applications. The first observation is that point-to-point system performance drops by at least 20% as concurrency increases with node count, up to 64 nodes. This is due to payload drop and re-transmit at each node. As concurrency increases, the average hops per-packet also increases, from 1.2 to 21 in 4-node and 64-node systems respectively, in our experiments. The hop distance is therefore a radius of about 30% ($1.2/4$ or $21/64$) the total number of nodes.

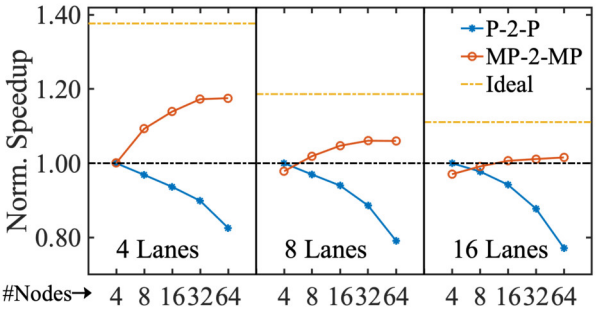


Fig. 10: Each data point is a geometric mean of all applications. Performance is grouped by link bandwidth. Each group is normalized to the lowest node count. *Ideal*, *P-2-P*, and *MP-2-MP* respectively represents performance upper-bound, point-to-point, and multipoint-to-multipoint systems.

Overall, our proposed multipoint system shows performance gains with concurrency, compared to an equivalent point-to-point system. But performance overhead depends on link bandwidth density, which translates to lane count. With increasing interposer lane count (4, 8, and 16), we observe diminishing performance gains; from 18% to 2% respectively for 4- to 16-lanes.

Our observed performance characteristics is due to packet sizing. In our experiments, the ratio of 8 B meta- to 72 B data-packets is 64% to 36% respectively. As link bandwidth density is improved (by adding more lanes), packet serialization reduces. In point-to-point systems, packets still need to traverse multiple nodes to get to the observed average hop count. Therefore, increased concurrency results in more delays and drastic performance loss. This is observed in all plots in Fig. 10. On the other hand, though multipoint system performance degrades with link density, the overall effect is relatively fixed performance.

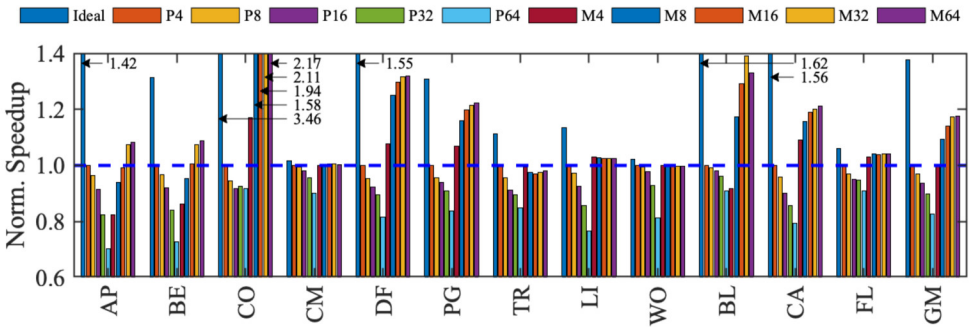


Fig. 11: Overall application performance for 4-Lane configuration. Groups *Px* and *Mx* respectively represent Point-to-Point and Multipoint-to-Multipoint, while the *x* designation represents node count. *Ideal* represents performance upper-bound.

Fig. 11 shows in-depth detail to put the performance gains of multipoint architecture into perspective. We show the application-specific performance for the 4-lane configuration of Fig. 10. For each application, there are 3 groups: *Ideal*, point-to-pint (*Px*), and multipoint-to-multipoint (*Mx*). The *x* label shows increasing node count, and therefore increasing concurrency. We notice that less than half of the application set actually contribute to the overall performance gain with concurrency. These applications generate much higher traffic, and exhibit relatively higher miss rates of over 5% (with *CO* having the highest). On the other hand, another group of applications show no performance gains with concurrency. These applications do not generate much traffic. Finally, we notice that *AP* and *BE* show quite difference performance characteristics, compared to the other two discussed above. Specifically, even though performance increases with concurrency in general, compared to point-to-point, the benefits are marginal. This is because both of these *graph* applications exhibit limited inter-node communication to none-neighbour nodes. And therefore do not benefit from the concurrency of multipoint system.

Finally, we notice that there is still room for improvement with multipoint channel systems. This is because performance is consistently 50% away from an idealized system as shown in Fig. 10.

We can determine the high-level energy characteristic of the interposer interconnection components of each system using a simple heuristic. In our experiments, we observe that 28 Gb of data is transmitted on average, for a high concurrent (64-node) configuration. We also observe an average hop count of 21. Using our energy profile from Table I, we can estimate the energy of interconnection of both systems. For point-to-point, this translates to roughly $766\,pJ$ [i.e., $28\,Gb * 1.29\,\frac{pJ}{b} * 21$]. For multipoint, the energy is $154\,pJ$ [i.e., $28\,Gb * (1.29\,\frac{pJ}{b} + (20 * 0.21\,\frac{pJ}{b}))$].

IV. Conclusion

In this paper, we propose a new multipoint-to-multipoint interconnect design to achieve high-concurrency communication between many chips on an interposer. It is more scalable than point-to-point links currently in use without the overheads of NoC designs. We show that when applied to a 1-D interconnected system with 4 to 64 nodes, the proposed interconnect exhibits almost constant performance at high link bandwidth density, and over 20% gain compared to conventional point-to-point interconnection.

References

[1] J. Jeddeloh et al. Hybrid Memory Cube New DRAM Architecture Increases Density and Performance. In *2012 Symposium on VLSI Technology (VLSIT)*, pages 87–88, 2012.

[2] T. O. Dickson et al. An 8x 10-Gb/s Source-Synchronous I/O System Based on High-Density Silicon Carrier Interconnects. *IEEE Journal of Solid-State Circuits*, 47(4):884–896, 2012.

[3] Y. Frans et al. A 56Gb/s PAM4 Wireline Transceiver Using a 32-Way Time-Interleaved SAR ADC in 16nm FinFET. In *2016 IEEE Symposium on VLSI Circuits (VLSI-Circuits)*, pages 1–2, 2016.

[4] J. Han et al. Design Techniques for a 60-Gb/s 288-mW NRZ Transceiver With Adaptive Equalization and Baud-Rate Clock and Data Recovery in 65-nm CMOS Technology. *IEEE Journal of Solid-State Circuits*, 52(12):3474–3485, 2017.

[5] S. Lee et al. Current-Mode Transceiver for Silicon Interposer Channel. *IEEE Journal of Solid-State Circuits*, 49(9):2044–2053, 2014.

[6] H. Lee et al. A 16.8 Gbps/Channel Single-Ended Transceiver in 65 nm CMOS for SiP-Based DRAM Interface on Si-Carrier Channel. *IEEE Journal of Solid-State Circuits*, 50(11):2613–2624, 2015.

[7] H. Ito et al. A Bidirectional- and Multi-Drop-Transmission-Line Interconnect for Multipoint-to-Multipoint On-Chip Communications. *IEEE Journal of Solid-State Circuits*, 43(4):1020–1029, 2008.

[8] J. Hu et al. A 25-Gbps 8-ps/mm Transmission Line Based Interconnect for On-Chip Communications in Multi-Core Chips. In *2013 IEEE MTT-S International Microwave Symposium Digest (MTT)*, pages 1–4, 2013.

[9] Y. Wang et al. Design High Bandwidth-Density, Low Latency and Energy Efficient On-Chip Interconnect. In *2017 IEEE/ACM International Symposium on Low Power Electronics and Design*, pages 1–6, 2017.

[10] R. Afoakwa et al. High swing pulse-amplitude modulation of transmission line links for on-chip communication. In *2018 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1–5, 2018.

[11] Lejie Lu et al. An energy-efficient high-swing pam-4 voltage-mode transmitter. In *Proceedings of the International Symposium on Low Power Electronics and Design*, ISLPED '18, pages 9:1–9:6. ACM, 2018.

[12] Binkert et al. The gem5 simulator. *can*, 39(2):1–7, 2011.

[13] M. Ahmad et al. Crono: A benchmark suite for multithreaded graph algorithms executing on futuristic multicores. In *Proceedings of IEEE International Symposium on Workload Characterization*, pages 44–55, 2015.

[14] T. Liu et al. Dthreads: Efficient deterministic multithreading. In *Proceedings of the 23rd ACM Symposium on Operating Systems Principles*, SOSP '11. ACM, 2011.

[15] C. Bienia et al. The PARSEC Benchmark Suite: Characterization and Architectural Implications. In *Proceedings of the International Conference on Parallel Architecture and Compilation Techniques*, sep 2008.