Exploiting Interposer Technologies to Disintegrate and Reintegrate Multicore Processors

The authors exploit a silicon interposer to disintegrate a multicore CPU chip into smaller chips that collectively cost less to manufacture than a single large chip. They study the performance-cost tradeoffs of interposer-based, multichip, multicore systems and propose interposer network-on-chip organizations to mitigate the performance challenges while preserving the cost benefits. This article also paves the way for various research problems in interposer-based disintegrated systems.

Ajaykumar Kannan
Natalie Enright Jerger
University of Toronto
Gabriel H. Loh
Advanced Micro Devices

Moore's law has conventionally enabled increasing integration; however, fundamental physical limitations have slowed the rate of transition from one technology node to the next, and the costs of new fabrication facilities are skyrocketing. The maturation of die stacking enables the continued integration of system components in traditionally incompatible processes. A key application of die-stacking is silicon-interposer-based integration of multiple 3D stacks of DRAM, shown in Figure 1a,[1–4] potentially providing several gigabytes of in-package memory with bandwidths already starting at 128 Gbytes per second per stack.

The use of an interposer presents new opportunities; in particular, if one has already paid for the interposer for the purposes of memory integration, any additional benefits from exploiting the interposer could come at a relatively small incremental cost. We study how the interposer can be used to address (at least in part) the increasing costs of manufacturing chips in a leading-edge process technology. We propose to use the interposer to "disintegrate" a large multicore chip into several small chips, such as in Figure 1b. These chips are less expensive to manufacture because of a combination of higher yield and better packing of the rectangular die on a round wafer. Unfortunately, this approach fragments the network on chip (NoC) such that each chip contains only a piece of the overall network, and communications between chips must take additional hops through the interposer. We explore how to disintegrate a multicore processor on an interposer while addressing the problem of a fragmented NoC.

Background and Motivation

To mitigate the rising costs of large multicores, systems can comprise multiple smaller

Published by the IEEE Computer Society

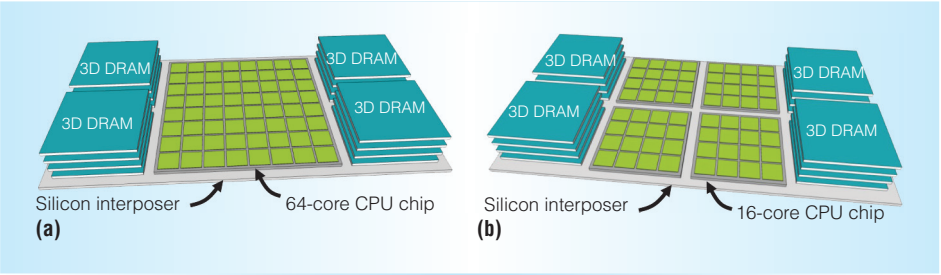

Figure 1. Potential organizations for interposer-based systems. (a) An example interposer-based system integrating a 64-core processor chip with four 3D stacks of DRAM. (b) A 64-core system composed of four 16-core processor chips.

chips. Various options exist for integrating multiple chips, including multisocket symmetric multiprocessors (SMPs), multichip modules (MCMs), 3D die stacking, and silicon interposers. The SMP and MCM approaches are less desirable because they do not provide adequate bandwidth for arbitrary core-to-core cache coherence without exposing significant nonuniform memory access effects. 3D stacking by itself is a less-attractive solution at this time because it is more expensive and complicated, introduces potential thermal issues, and can be an overkill in terms of how much bandwidth it can provide. This leaves us with silicon interposers. Having settled on silicon interposers as our integration technology, the key contributions of this work include a cost and yield analysis to show the benefits of disintegration, the potential for minimally active interposers, novel NoC topologies, and the concept of misaligned NoCs.

Cost and Yield

To reduce the cost of manufacturing a system on a chip (SoC), we can reduce the chip's size. A larger chip's cost comes from two main sources. The first is geometry: fewer larger chips fit on a wafer, and the smaller chips can be packed more tightly (that is, utilizing more of the area around the periphery of the wafer). The second cost of a larger chip is due to manufacturing defects. A defect that renders a large die inoperable wastes more silicon than one that kills a smaller die. Although smaller chips result in lower costs, the downside is that they also provide less functionality (for example, half the area yields half the cores). If we could manufacture several smaller chips and combine them together into a single system, we would be able to have the functionality of a larger chip while maintaining the economic advantages of the smaller chips. Ignoring for the time being exactly how multiple chips could be combined back together, Table 1 summarizes the impact of implementing a 64-core system ranging from a conventional 64-core monolithic chip all the way down to building it from 16 separate quad-core chips. The last column shows that using a collection of quad-core chips to assemble a 64-core SoC yields 29 percent more working parts than the monolithic-die approach.

Additional performance benefits can be had by speed-binning the individual chips to ensure that fast chips are stacked together on the same interposer to maximize the number of high-performance, high-margin parts produced. We used Monte Carlo simulations to consider three scenarios:

- A 300-mm wafer is used to implement 162 monolithic good dies per wafer (as per Table 1).
- The wafer is used to implement 3,353 quad-core chips, which are then assembled without speed-binning into 209 64-core systems.
- The individual dies from the same wafer are sorted so that the 16 fastest chips are assembled together, the next fastest 16 are combined, and so on.

Figure 2 shows the number of 64-core systems per wafer in 100 MHz speed bins, averaged across 100 wafer samples per scenario. The monolithic 64-core chip and 16 quad-core approaches have similar speed distributions. However, with speed-binning, we avoid the situation wherein the overall system

Table 1. Example yield analysis for different-sized multicore chips. A system on a chip (SoC) here is a 64-core system, which might require combining multiple chips for the rows in the table corresponding to chips with fewer than 64 cores each.

Cores per chip	Chips per wafer	Chips per package	Area per chip (mm^2)	Chip yield (%)	Good dies per wafer	Good SoCs per wafer
64	192	1	297.0	84.5	162	162
32	395	2	148.5	91.7	362	181
16	818	4	74.3	95.7	782	195
8	1,664	8	37.1	97.8	1,627	203
4	3,391	16	18.6	98.9	3,353	209

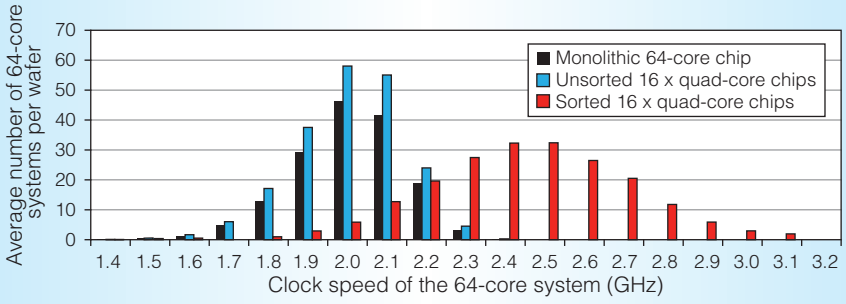

Figure 2. Average number of 64-core SoCs per wafer per 100 MHz bin from Monte Carlo simulations of 100 wafers. Results are shown for a monolithic 64-core chip and systems with 16 quad-core chips. Using multiple chips allows speed binning, leading to higher clock rates as shown in the sorted case.

speed is slowed down by the presence of a single slow chip, resulting in significantly faster average system speeds (the mean shifts by approximately 400 MHz) and more systems in the highest speed-bins (which usually carry the highest profit margins).

Minimally Active Interposers

The interposer is effectively a very large chip. Current approaches use passive interposers[2,5] in which the interposer has no devices, only routing. This greatly reduces the interposer's critical area (A_{crit}) (that is, unless a defect impacts a metal route, there are no transistors that can be affected), resulting in high yields. Based on our previous chip cost analysis, it would seem prohibitively expensive to consider an active interposer, but the flexibility of placing routers on the interposer enables many more interesting NoC organizations. Although the interposer could be implemented in an older process technology[6] that is less expensive and more mature (that is, one with lower defect rates), such a large chip, perhaps near the reticle limit, still would not be expected to be less expensive if the yield rates remained low.

For regular chips, it typically is desirable to maximize functionality by cramming in as many transistors as possible into the chip-area budget. However, making use of every last mm^2 of the interposer would lead to a very high fraction of the area being critical ($Frac_{crit}$) multiplied over a very large area (A_{crit} = chip area × $Frac_{crit}$), thereby leading to low yields and high costs. However, there is no need to use the entire interposer: its size is determined by the geometry of the chips and memory stacked on it, and using more or fewer devices has no impact on its final size. As such, we advocate using a *minimally active interposer,* which implements

the devices required for the system's functionality (in our case, these would primarily be routers and repeaters), but no more. This results in a sparsely populated interposer with a lower $Frac_{crit}$ and therefore a lower cost.

We used the same yield model from our earlier cost analysis to estimate the yields of different interposer options: a passive interposer, a minimally active interposer, and a fully active interposer. The interposer size assumed throughout this work is 24 mm × 36 mm (864 mm^2), and we assume six metal layers in the interposer. For a passive interposer, $Frac_{crit}$ for the logic is zero (it remains nonzero for the metal layers). For a fully active interposer (that is, if one fills the entire interposer with transistors), we use a $Frac_{crit}$ for logic of 0.75 and $Frac_{crit}$ for wires of 0.2625. For a minimally active interposer, we estimate the total interposer area needed to implement our routers (logic) and links (metal) to be only 1 percent of the total interposer area. To be conservative, we also consider a minimally active interposer in which we pessimistically assume the router logic consumes 10 times more area, although the metal utilization is unchanged. Minimizing utilization of the interposer for active devices also minimizes the potential for undesirable thermal interactions resulting from stacking highly active CPU chips on top of the interposer.

Considering an example defect rate of 2,000 defects per m^2 (from Table 1), we find that the passive interposer has a nonperfect yield rate of 94.1 percent because it still uses metal layers that can be rendered faulty by manufacturing defects. At the other extreme is a fully active interposer with very low yields of 61.5 percent. This is not surprising given that a defect almost anywhere on the interposer could render it a loss. This is the primary reason why one would likely be skeptical of active interposers. However, when using only the minimum amount of active area necessary on the interposer, the yield rates are not very different from the passive interposer—93.9 percent for 1 percent active and 92.4 percent for 10 percent active. The vast majority of the interposer is not being used for devices; defects that occur in these white-space regions do not impact the interposer's functionality. As a result, we believe that augmenting an otherwise passive interposer with just enough logic to do what is needed has the potential to be economically viable, and it should be sufficient for NoC-on-interposer applications.

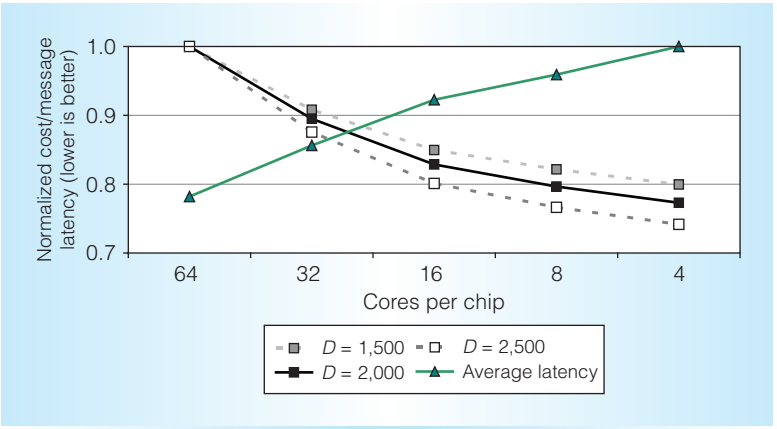

Figure 3. Normalized cost and execution time (lower is better for both) for different multichip configurations. Sixty-four cores per chip corresponds to a single monolithic 64-core die, and four cores per chip corresponds to 16 chips, each with four cores. Cost is shown for different defect densities (in defects/m^2), and the average message latency is normalized to the 16 quad-core configuration.

Taking the cost argument alone to its logical limit would lead one to falsely conclude that a large chip should be disintegrated into an infinite number of infinitesimally small dies. The countervailing force is performance: although breaking a large system into smaller pieces could improve overall yield, going to a larger number of smaller chips increases the amount of chip-to-chip communication that must be routed through the interposer. In an interposer-based multicore system with a NoC distributed across chips and the interposer, smaller chips create a more fragmented NoC, resulting in more core-to-core traffic routing across the interposer, which eventually becomes a performance bottleneck. Figure 3 shows the cost reduction for three example defect rates, all showing the relative cost benefit of disintegration. The figure also shows the relative impact on performance. So, although more aggressive levels of disintegration provide better cost savings, they are directly offset by a reduction in performance.

In the rest of this article, we explore the problem of how one can get the cost benefits

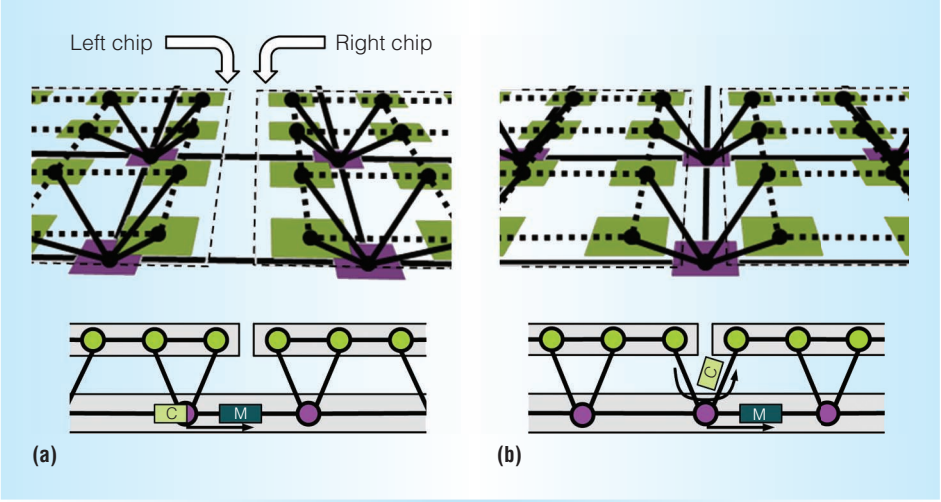

Figure 4. Perspective and side/cross-sectional views of (a) 4-to-1 concentration from cores to interposer routers aligned beneath the CPU chips, and (b) 4-to-1 concentration misaligned such that some interposer routers are placed in between neighboring CPU chips. The cross-sectional view also illustrates the flow of example coherence (C) and memory (M) messages.

of a disintegrated chip organization while providing a NoC architecture that, while physically fragmented across multiple chips, still behaves (performance-wise) at least as well as one implemented on a single monolithic chip.

Architecting the NoC for Multichip Interposers

The analysis in the preceding section shows that there are economic incentives for disintegrating a large multicore chip into smaller dies, but that doing so induces performance challenges with respect to the interconnect. We now discuss how to address these issues.

We propose the concept of *misalignment*. With conventional topologies such as a concentrated Folded Torus, links that cross the bisection between the two halves of the interposer still carry a higher amount of traffic and continue to be a bottleneck for the system. For concentrated topologies in which one router connects to four cores, every four CPU cores in a 2×2 grid share an interposer router that was placed in between them, as shown in both the perspective and side/cross-sectional views in Figure 4a.

Our misaligned interposer network offsets the location of the interposer routers. Cores on the edge of one chip now share a router with cores on the edge of the adjacent chip (see Figure 4b). The change is subtle but important: with an "aligned" interposer NoC, the key resources shared between chip-to-chip coherence and memory traffic are the links crossing the bisection. If both a memory-bound message (M) and a core-to-core coherence message (C) wish to traverse the link, one must wait as it serializes behind the other. With misaligned topologies, the shared resource is now the router. As the bottom of Figure 4b shows, this simple shift allows chip-to-chip and memory traffic to flow through a router simultaneously, thereby reducing queuing delays for messages to traverse the network bisection.

For interposer-based NoCs, providing sufficient bisection bandwidth is critical. One straightforward way to provide more bisection bandwidth is to add more links. However, if this is not done carefully, it can cause the routers to need more ports (higher degree), which increases area and power, and can decrease the router's maximum clock speed. By combining different topological aspects of both the Butterfly and Folded Torus topologies into our novel ButterDonut topology, we can further increase the interposer NoC bisection bandwidth without impacting the router complexity. We maintain the advantages of long express links of

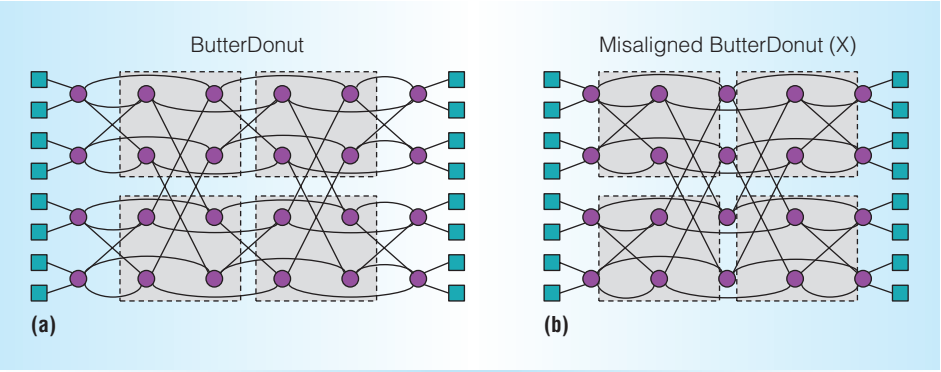

Figure 5. Our ButterDonut topology. The (a) aligned and (b) misaligned ButterDonut topologies combine topological elements from both the Double Butterfly and Folded Torus.

the butterfly and exploit folded ring networks in the X-dimension (from the Folded Torus). This increases the bisection bandwidth and leads to low hop counts for both coherence and memory traffic, as shown in Figure 5a. The ButterDonut can also be misaligned to provide even higher throughput across the bisection, as shown in Figure 5b.

We can compare topologies among several different metrics. Table 2 shows all of the concentrated topologies considered in this article, along with several key network and graph properties. The metrics listed correspond only to the interposer's portion of the NoC (for example, nodes on the CPU chips are not included), and the link counts exclude connections both to the CPU cores as well as to the memory channels (this is constant across all configurations, with 64 links for the CPUs and 16 for the memory channels). Misaligned topologies are annotated with their misalignment dimension in parentheses; for example, the Folded Torus misaligned in the X-dimension is shown as "Folded Torus (X)." Misalignment can change the number of nodes (routers) in the network (see, for example, Figure 5). From the perspective of building minimally active interposers, we favor topologies that minimize the number of nodes and links to keep the interposer's A_{crit} as low as possible. At the same time, we want to keep the network diameter and average hop count low (to minimize expected latencies of requests) while maintaining high bisection bandwidth (for network throughput). Overall, the X-misaligned ButterDonut topology has the best properties out of all of the topologies except for the link count, for which it is a close second behind Double Butterfly (X). ButterDonut (X) combines the best of all of the other non-ButterDonut topologies, while providing 50 percent more east-west bisection bandwidth.

Methodology

For the yield and relative cost figures, we use analytical yield models, a fixed cost-per-wafer assumption, and automated tools for computing die-per-wafer,[7] and we consider a range of defect densities. All analyses assume a 300-mm wafer. Our baseline monolithic 64-core die size is 16.5 mm × 18 mm (the same assumption as used in a recent interposer-NoC paper[3]). Smaller-sized chips are derived by halving the longer of the two dimensions (for example, a 32-core chip is 16.5 mm × 9 mm). The yield rate for individual chips is estimated using a simple classic model.[8]

For the speed binning results given earlier, we simulate a wafer's yield by starting with the good-die-per-wafer based on the desired chip's geometry (see Table 1). For each quad-core chip, we randomly select its speed using a normal distribution (mean: 2,400 MHz; standard deviation: 250 MHz). Our simplified model treats a 64-core chip as the composition of 16 adjacent (4 × 4) quad-core clusters, with the speed of each cluster chosen from the same distribution as the individual quad-core chips. Therefore, the 64-core chip's clock speed is the minimum from among its constituent 16 clusters. For each configuration, we simulate 100 different wafers' worth of parts and take the average over the 100 wafers. Similar to the yield

Table 2. Comparison of the different interposer NoC topologies studied in this article. In the node column, ($n \times m$) indicates the organization of router nodes. Bisection links are the number of links crossing the vertical bisection cut.

Topology	Nodes	Links	Diameter	Average hop	Bisection links
Concentrated Mesh	24 (6 × 4)	38	8	3.33	4
Double Butterfly	24 (6 × 4)	40	5	2.70	8
Folded Torus	24 (6 × 4)	48	5	2.61	8
ButterDonut	24 (6 × 4)	44	4	2.51	12
Folded Torus (X)*	20 (5 × 4)	40	4	2.32	8
Double Butterfly (X)*	20 (5 × 4)	32	4	2.59	8
Folded Torus (XY)*	25 (5 × 5)	50	4	2.50	10
ButterDonut (X)*	20 (5 × 4)	36	4	2.32	12

*Misaligned.

results, the exact distribution of per-chip clock speeds is not so critical: so long as there exists a spread in chip speeds, binning and reintegration via an interposer can be beneficial for the final product speed distribution.

To evaluate the performance of various interposer NoC topologies for our disintegrated systems, we use a cycle-level network simulator.[9] To evaluate the baseline and proposed NoC designs, we use both synthetic traffic patterns and SynFull traffic models[10]; these two evaluation approaches cover a wide range of network utilization scenarios and exercise both cache-coherence and memory traffic. For the SynFull workloads, we run multiprogrammed combinations composed of four 16-way multithreaded applications from Parsec.[11] The application's threads are distributed across the 64 cores, and they share all 16 memory channels. We construct workload combinations based on their memory traffic intensity.

Evaluation

We evaluated the performance, power, and cost of our disintegrated multicore architecture considering different chip sizes and NoC topologies. We compared our proposed misaligned and ButterDonut topologies against more conventional topologies, such as a Mesh, Concentrated Mesh (CMesh), and Folded Torus. To evaluate the baseline and proposed NoC designs, we considered several traffic workloads to cover a wide range of network utilization scenarios. We used synthetic traffic patterns to stress our proposed network and realistic application workloads to evaluate NoC performance. Figure 6 shows average packet latency results when the system executes multiprogrammed SynFull workloads. The results are for a system consisting of four 16-core CPU chips. These workloads exercise a diverse set of routes highlighting differences among the topologies. Across the workloads, the misaligned ButterDonut (X) and misaligned Folded Torus (XY) consistently perform the best.

Our evaluations show that the proposed NoC architecture outperforms the baseline Mesh and CMesh. Misaligned topologies provide the best performance by reducing bisection pressure through isolating chip-to-chip coherence traffic from interposer memory traffic. Having this traffic share a router but not a link provides substantial throughput improvements. The ButterDonut topologies generally provide the best performance by optimizing both east-west memory traffic through long links and chip-to-chip coherence traffic through reduced diameter from the folded rings along the X dimension.

Higher levels of disintegration (smaller chips) can result in lower overall cost based on our analysis, due to better yields and more chips per wafer. The smaller chips could decrease interconnect performance because the NoC is fragmented, but the finer-grained binning also increases average CPU clock speeds. To put it all together, we consider a figure of merit (FOM) based on cost and performance, or "delay" for brevity. Our FOM is delay^{cost}, which provides a greater emphasis on performance (see Figure 7).

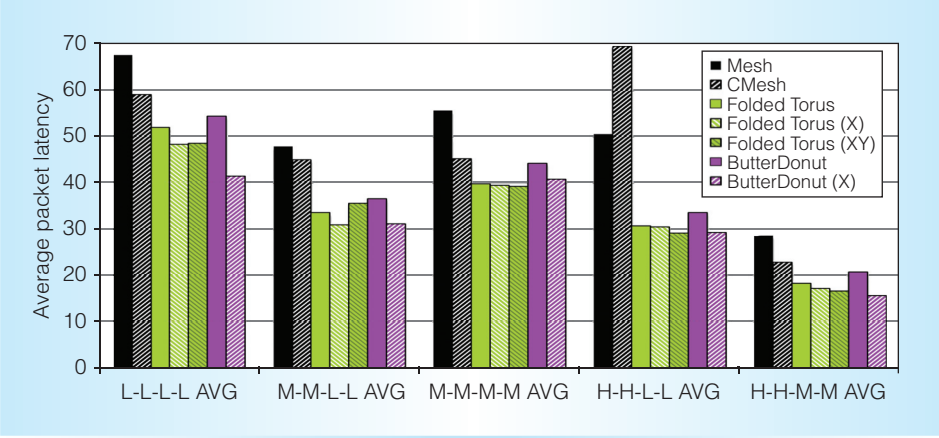

Figure 6. Average packet latency results for different multiprogrammed SynFull workloads with varying network load intensities. Aligned and misaligned topologies are compared against the Mesh and CMesh, with the misaligned ButterDonut having the lowest average latency in most cases.

The rationale for the performance-heavy FOM is that for high-end servers, even relatively smaller performance differentiations at the high end can translate into substantially higher selling prices and margins.

Figure 7 shows our FOM. For a basic Mesh on the interposer, the performance loss due to disintegration actually hurts more than the cost reductions help until one gets down to using eight-core chips or smaller. CMesh provides an initial benefit with two 32-core chips, but its lower bisection bandwidth is too much of a drag on performance, and further disintegration is unhelpful. The FOM results show that the remaining topologies can ward off the performance degradations of the fragmented NoC sufficiently well such that the combination of binning-based improvements and continued cost reductions allow more aggressive levels of disintegration.

With different FOMs, the exact tradeoff points will shift, but our FOM illustrates that simple disintegration (using Mesh or CMesh) alone might not be sufficient to provide a compelling solution for both cost and performance. However, interposer-based disintegration appears promising when coupled with an appropriate redesign of the interposer NoC topology.

The combination of 2.5D and 3D stacking technologies enabled through silicon-interposer-based integration will significantly change the hardware organization of many future systems. We believe that the general framework of silicon-interposer-based integration for the physical organization and assembly of future SoCs will become common, and one of the key goals of this article is to raise awareness of and evangelize for new research efforts in this broad area considering impacts on cost, performance, power, and functionality.

We focused on a homogeneous disintegrated multicore system using 2.5D stacking to connect processors to 3D stacks of memory. Although we focused on interconnect issues to reintegrate homogeneous multicore chips, we anticipate broader interest and new research around silicon-interposer-based systems in general. The interposer simply provides a mechanical and electrical substrate for the integration of multiple disparate chips. Multiple computing chips (such as the CPU and GPU) can be 2.5D stacked on the interposer with the DRAM. Separate chips for logic and analog devices, possibly in different process technology nodes, can be independently manufactured and 2.5D stacked. Having a solid framework for integrating these disparate components enables a wide range of systems and prompts new research questions.

We focused on enabling cost-effective, large multicore designs. However, this is a rich platform for many exciting new systems that could become more economically viable. Consider, for example, new accelerators that

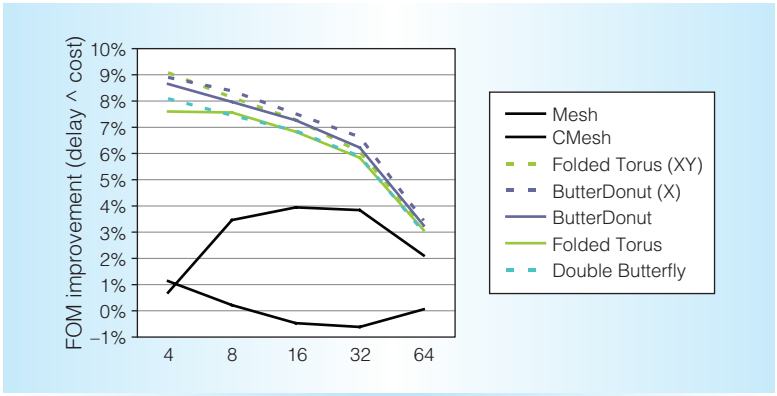

Figure 7. Figure of merit based on delay^{cost}. The *x*-axis specifies cores per chip (64 = monolithic). Proper topology design improves cost and performance at high levels of disintegration (for example, four cores per chip).

might not be applicable across all market segments. Without interposer-based integration, a new accelerator would likely need to be directly incorporated into the same monolithic SoC silicon as the other commodity components (for example, the CPUs and memory controllers). This saddles the overall SoC with the cost of the accelerator, even for products that do not benefit from acceleration, or forces the manufacturer to pay for additional engineering to develop two separate SoC designs (that is, with and without the accelerator). With interposer-based integration, the manufacturer can separately design a stand-alone CPU and accelerator chips, and then individual SoC designs can choose which chips to stack. Although generic, our proposed interconnect designs offer sufficiently rich connectivity to enable efficient routing of chip-to-chip and chip-to-memory traffic regardless of the different SoC components.

Our approach could dramatically accelerate the adoption of accelerator-based technologies. This in turn motivates new research in general system architectures to support a wide range of accelerators in a way that enables plug-and-play at the system level with respect to shared virtual memory, cache coherency between conventional computing and accelerators, work/task scheduling, and more.

We proposed the use of a minimally active interposer. We estimate that only 1 percent of the interposer area must be devoted to interconnect logic. Devoting this minimal amount of area to active logic has a very small impact on interposer yield. Although cramming the interposer full of transistors would be prohibitively expensive, we believe that some additional area can be spent on new interposer-based logic. Indeed, our results suggest that even devoting 10 percent of interposer area to active logic could still be practical. The cost-effective nature of such minimally active interposers opens many interesting research opportunities as we decide what to put on the interposers and how best to use the devices and routing there. Beyond the interconnect, the interposer could house all manner of system monitoring and online profiling ("introspection") mechanisms, auxiliary computing devices, and security features. The opportunities to reimagine computer architectures are wide open.

Given the value of interposer-based systems, a key open question is exactly how to interconnect the individual chips, memory stacks, and interposer. We take a first step in exploring possible topologies to effectively route both coherence and memory traffic across a disintegrated CPU, but we expect much follow-on work from the community. A key contribution is the advocacy of general approaches to interposer-based interconnect design. Future interposer-based systems will likely span a large range of sizes, functionalities, and physical arrangements of chips and memory stacks. Our approach to interposer-based NoCs, while illustrated with a specific example system, extends across a broad range of heterogeneous designs. Common ground must be found across this wide range of systems to design a generic interconnect that can effectively reintegrate a limitless range of future systems. Our focus on low diameter and misalignment to avoid bisection-link bottlenecks are a step in this direction, but new research is needed to understand the impact of traffic patterns in these different systems and re-envision the NoC accordingly. Our work provides a stepping-stone for these designs.

This work assumes that all core dies are identical. This creates new physical design and layout challenges as each die must implement a symmetric interface since each die must correctly interface with the interposer regardless of the mounting position. Conventional SoCs have no such requirements for their layouts. This extends beyond functional

interfaces to issues such as power delivery and thermal management. Many of these challenges fall more in the domains of physical design, CAD, and electronic design automation. However, our work demonstrates the cost and performance potential for these systems, which in turn motivates new research in the physical design realm to support such disintegrated systems. Both the architecture and the larger ecosystem require additional research to make these compelling systems a reality. MICRO

References

1. "AMD Ushers in a New Era of PC Gaming with Radeon R9 and R7 300 Series Graphics Line-Up Including World's First Graphics Family with Revolutionary HBM Technology," Advanced Micro Devices, June 2015.
2. B. Black, "Die Stacking Is Happening," *Proc. Int'l Symp. Microarchitecture*, 2013; www.microarch.org/micro46/files/keynote1.pdf.
3. N. Enright Jerger et al., "NoC Architectures for Silicon Interposer Systems," *Proc. 47th Int'l Symp. Microarchitecture*, 2014, pp. 458–470.
4. M. O'Connor, "Highlights of the High-Bandwidth Memory (HBM) Standard," *Memory Forum Workshop*, 2014; www.cs.utah.edu/thememoryforum/mike.pdf.
5. Y. Deng and W. Maly, "Interconnect Characteristics of 2.5-D System Integration Scheme," *Proc. Int'l Symp. Physical Design*, 2001, pp. 171–175.
6. N. Madan and R. Balasubramonian, "Leveraging 3D Technology for Improved Reliability," *Proc. 40th Int'l Symp. Microarchitecture*, 2007, pp. 223–235.
7. M. Hackerott, "Die Per Wafer Calculator," Informatic Solutions, 2011.
8. C.H. Stapper, "The Effects of Wafer to Wafer Defect Density Variations on Integrated Circuit Defect and Fault Distributions," *IBM J. Research and Development*, Jan. 1985, pp. 87–97.
9. N. Jiang et al., "A Detailed and Flexible Cycle-Accurate Network-on-Chip Simulator," *Proc. Int'l Symp. Performance Analysis of Systems and Software*, 2013, pp. 86–96.
10. M. Badr and N. Enright Jerger, "SynFull: Synthetic Traffic Models Capturing a Full Range of Cache Coherence Behaviour," *Proc. ACM/IEEE 41st Int'l Symp. Computer Architecture*, 2014, pp. 109–120.
11. C. Bienia, "Benchmarking Modern Processors," PhD dissertation, Dept. Computer Science, Princeton Univ., 2011.

Ajaykumar Kannan is an advanced design engineer in the Intel Programmable Solutions Group. His research interests include silicon-interposer-based networks on chip. Kannan received an MASc in computer engineering from the University of Toronto, where he completed the work for this article. Contact him at kannan.ajay@gmail.com.

Natalie Enright Jerger is an associate professor in the Department of Electrical and Computer Engineering at the University of Toronto. Her research interests include multi- and many-core architectures, on-chip networks, cache coherence protocols, memory systems, and approximate computing. Enright Jerger received a PhD in computer architecture from the University of Wisconsin–Madison. She is the recipient of an Alfred P. Sloan Research Fellowship, the Borg Early Career Award, the 2014 Ontario Professional Engineers Young Engineer Medal, and the 2012 Ontario Ministry of Research and Innovation Early Researcher Award. She served as program co-chair of the 7th Network-on-Chip Symposium (NOCS) and as program chair of the 20th International Symposium on High-Performance Computer Architecture (HPCA). Contact her at enright@eecg.toronto.edu.

Gabriel H. Loh is a Fellow at Advanced Micro Devices. His research interests include computer architecture, processor microarchitecture, emerging technologies, and 3D die stacking. Loh received a PhD in computer science from Yale University. He's a senior member of IEEE and an ACM distinguished scientist. Contact him at gabriel.loh@amd.com.