Interconnect-Memory Challenges for Multi-chip, Silicon Interposer Systems

Gabriel H. Loh[†] Natalie Enright Jerger[‡†] Ajaykumar Kannan[‡] Yasuko Eckert[†]

[†]AMD Research
Advanced Micro Devices, Inc.
{gabriel.loh,yasuko.eckert}@amd.com

[‡]Dept. of Electrical and Computer Engineering
University of Toronto
{enright,kannanaj}@ece.utoronto.ca

ABSTRACT

Silicon interposer technology is promising for large-scale integration of memory within a processor package. While past work on vertical, 3D-stacked memory allows a stack of memory to be placed directly on top of a processor, the total amount of memory that could be integrated is limited by the size of the processor die. With silicon interposers, multiple memory stacks can be integrated inside the processor package, thereby increasing both the capacity and the bandwidth provided by the 3D memory. However, the full potential of all of this integrated memory may be squandered if the in-package interconnect architecture cannot keep up with the data rates provided by the multiple memory stacks. This position paper describes key issues in providing the interconnect support for aggressive interposer-based memory integration, and argues for additional research efforts to address these challenges to enable integrated memory to deliver its full value.

CCS Concepts

•Hardware → Dynamic memory; 3D integrated circuits; Package-level interconnect;

Keywords

Silicon interposer; die stacking; memory; interconnect.

1. INTRODUCTION

Die-stacking technologies have now been researched for many years as a promising approach for (at least partially) dealing with the Memory Wall problem [30]. In recent years, 3D-stacked memories [11, 14, 22] are finally coming to market and making their way into consumer products [1, 3, 10]. While much early academic work on die-stacking focused on vertical 3D stacking of memory directly on top of processors, the current trends show that 2.5D or silicon interposer-based stacking [7] is gaining more traction.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

MemSys '15 Oct 6–8, 2015, Washington, DC, USA

© 2015 ACM. ISBN 978-1-4503-2138-9.

DOI: 10.1145/1235

Interposer-based integration provides several advantages over pure 3D stacking, which we will explore in greater detail in Section 2. In particular, in a 3D-stacked approach, the total amount of in-package memory that can be integrated is bounded by the size of the processor die upon which the memory is to be stacked. If the processor only has enough area for one or two stacks of memory, then that places a tighter limit on the total in-package memory capacity. An interposer-based arrangement, however, decouples the size of the processor chip from the memory stacks, thereby allowing as much memory to be integrated up to the limit of the interposer size and packaging costs. Furthermore, with a fixed bandwidth per memory stack, enabling the integration of more stacks directly translates into more bandwidth.

So while silicon interposers can be used to provide larger capacities and significantly higher bandwidths for in-package memory, this paper argues that a critical outstanding research problem is how to architect the interconnect substrate that ties the processor together with all of these memory resources. In this paper, we walk through some of the key challenges that need to be addressed so that large-scale in-package memory integration does not end up being hamstrung by the underlying interconnection architecture.

2. INTERCONNECT CHALLENGES FOR INTERPOSER-STACKED MEMORIES

2.1 3D vs. Interposer Memory Stacking

3D stacking enables the placement of memory (which itself will likely be 3D stacked) directly on top of a processor die. Fig. 1(a) shows an example arrangement with a processor die and a stack of 3D-integrated memory. Fig. 1(b) shows a system with similar capabilities, but a silicon interposer is used to integrate the memory horizontally next to the processor die. Both approaches have pros and cons.

Memory Capacity and Bandwidth:

For 3D stacking, the primary advantage is that the maximum potential bandwidth between two layers is limited by the common surface area. With a through-silicon via (TSV) pitch of 50μm, this could theoretically provide eight million TSVs on a 200cm^2 chip (and more aggressive TSV pitches have been demonstrated [28]). In contrast, the connectivity between the processor chip and the memory stack(s) across a silicon interposer is bounded by the perimeter of the chip. Assuming an interposer wire pitch of 50μm, the same 200cm^2 chip could only support about 11 thousand connec-

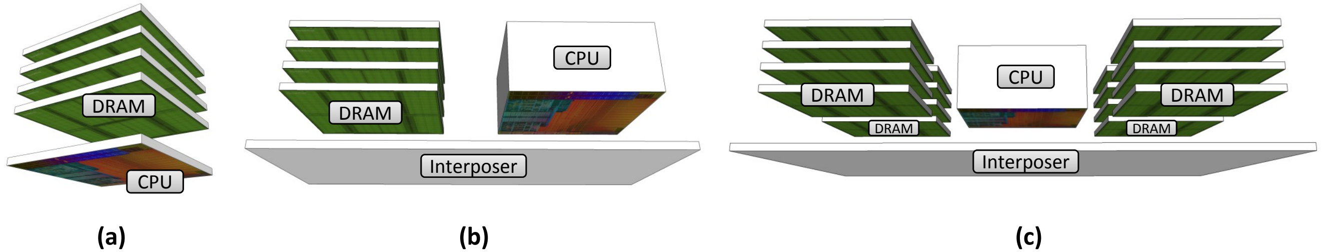

Figure 1: Processor-and-memory systems using (a) vertical 3D stacking, (b) 2.5D stacking on a silicon interposer, and (c) 2.5D stacking incorporating multiple memory stacks.

tions assuming the entire ∼56cm perimeter of the chip was used for routing to the memory stack(s).[1]

While there is an orders of magnitude difference between the 3D and interposer processor-memory bandwidths *in the limit*, the difference has a much smaller practical impact for memory stacking.[2] Four channels of conventional DDR3 interfaces require 960 package pins; the many *thousands* of connections available across an interposer side-steps this bottleneck to provide much higher bandwidths at lower energy-per-bit costs. The big win is in getting the memory inside the package in the first place. Any further benefits of 3D over 2.5D are relatively small in comparison to the gain from the off-package to in-package transition.

A key advantage of silicon interposers is that the amount of memory that can be integrated is not limited by the size of the processor die. Fig. 1(c) shows an example where the total area consumed by the multiple memory stacks is significantly greater than the processor die's footprint. With 3D stacking, one would not be able to place all of this memory on top of the processor. As such, the interposer-based approach can provide more total in-package memory capacity. In a similar manner, current 3D memory standards provide a fixed amount of bandwidth per stack, and so the total available bandwidth is directly related to the total number of stacks integrated in the package. For example, a JEDEC high-bandwidth-memory (HBM) stack [11] may have 1024 bits of data signals operating at 1 Gbps, which provides 128 GB/s for a single stack. With four stacks as shown in the figure, the total available bandwidth is 512 GB/s.

Thermal Behaviors:

Another benefit of silicon interposer organizations is easier thermal management due to the fact that the hot processor die does not have any additional layers stacked above it, and therefore can maintain direct physical contact with the thermal interface material and heat spreader. While past works have demonstrated that 3D-stacked structures can be effectively cooled [8, 21], it is still desirable to maintain lower temperatures as that can either result in less-expensive cooling solutions or provide more headroom for voltage-boosting/sprinting techniques [19, 24, 25]. Fig. 2(a) shows the peak temperature (across all dies) for each of the configurations illustrated earlier (Fig. 1).[3] The experiments vary the power consumed by the processor die while maintaining a constant power consumption of 4W per memory stack. The silicon-interposer approach consistently achieves a lower peak temperature. The four-stack version is slightly hotter, as the total power is 12W higher (due to the three additional memory stacks) while the overall cooling/package assumptions remain the same as the other cases.

In a similar fashion, Fig. 2(b) shows the peak temperature among only the memory stacks for the same configurations. Not only is the processor die cooler in the interposer case than in the 3D case, but the same holds true for the memory. The processor's heat path is not blocked by memory stacks, and the memory stacks are not being directly heated by the processor die (there is still some indirect thermal coupling through the interposer, but the memory stays about ∼10°C cooler than the processor die). Maintaining lower memory temperatures are also important to keep refresh rates down and improve the reliability of the memory.

Fabrication and Costs:

The different stacking approaches require different manufacturing steps and incur different costs along the way. One of the most immediate differences is that 2.5D integration requires the fabrication (and cost) of the silicon interposer whereas 3D stacking needs no additional silicon. The interposer is potentially quite large (must be large enough to fit all of the stacked components), and it must be thinned to provide TSVs to connect through to the package-level C4 bumps, both of which add to the cost. To (partially) offset this, the interposer could potentially be implemented in an older (and cheaper) process technology generation.

The 3D-stacked structure is not without its costs as well. First, the processor chip must be redesigned to deal with having a multitude of TSVs driven through it (which has implications in terms of engineering effort, tools support, physical design, etc.). In contrast, a relatively conventional chip design can be used for 2.5D stacking as the processor die has no TSVs and does not need to undergo die thinning (risking cracking and other mechanically-related yield issues). For the 3D version, the additional TSVs also in-

[1] These are simply estimates, as they assume that the *entire* processor chip area/perimeter will be used for connections to the memory, but nevertheless this illustrates the orders of magnitude differences between the approaches.

[2] For logic-on-logic stacking, the differences may matter more, but the focus of this paper is on memory.

[3] The thermal simulations were conducted with a version of the HotSpot [27] tool that was extended to better support 3D modeling including the air gaps between the different die/stacks. Full methodology details are omitted here as this is a position paper and the data are provided only for motivational purposes.

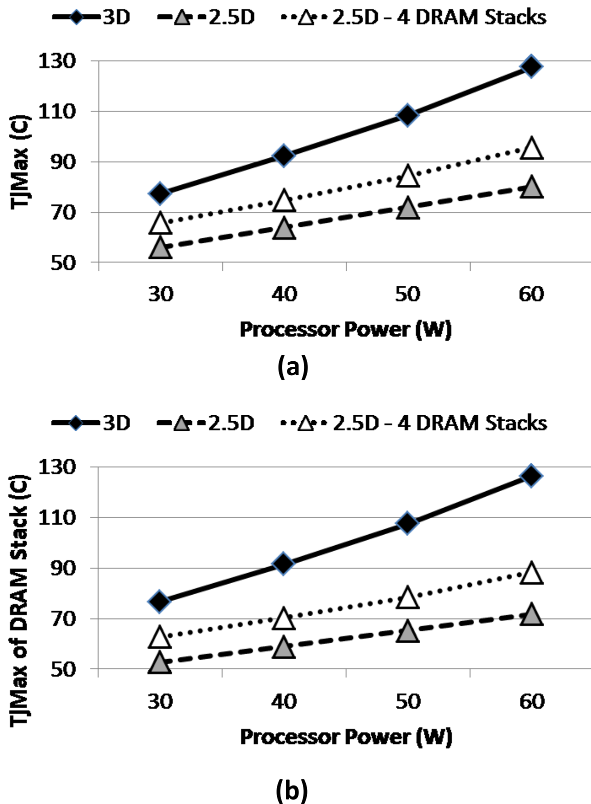

Figure 2: Maximum observed temperature in a processor-and-memory integrated system (a) across all die, and (b) only within the memory stack(s).

crease the size of the processor chip, not only for the area taken up by the TSVs, but also due to additional "keep out zones" around each TSV [2, 20]. This increases the chip size of the processor die. ■

There are other pros and cons between 2.5D and 3D stacking solutions, but given the main points above, we argue that silicon interposers are a very desirable technology for memory integration due to the combination of higher capacities, higher bandwidths, and more manageable thermals. From a fabrication and cost perspective, it is still too early to say whether there is a clear-cut winner between the two approaches as there are many different trade-offs. Furthermore, 2.5D and 3D integration are not mutually exclusive (e.g., additional memory stacks could be placed on top of a processor that was already 2.5D-stacked on an interposer).

2.2 Getting to the Memory

At this point, we have motivated and argued for interposer-based integration of in-package memory. This brings us to the crux of this paper: how does the processor (including individual computing units such as CPU cores and GPU pipelines) get to all of these in-package memory resources?

To make the problem more concrete, let us consider the system depicted in Fig. 3(a) with the four stacks of memory and a 64-core processor. We will also use the previous memory bandwidth estimate of 128GB/s per stack (512 GB/s total). Furthermore, the system would likely also support multiple channels of external memory (e.g., DDR3, DDR4) as even with multiple stacks of memory, it is unlikely that the entirety of a system's main memory can be provided by stacked DRAM [16, 18]. We will assume four channels of DDR4 operating at 2666 MHz, providing ∼21 GB/s per channel, or ∼83 GB/s total.

In our first scenario, we consider a well-balanced application where each processor core's memory traffic is uniformly sourced from each stack of memory and each external memory channel. More specifically, each stack's 128 GB/s bandwidth would be evenly divided among each of the cores (i.e., 2 GB/s per core per stack). Likewise, each external memory channel's 21 GB/s is divided evenly 64 ways for 0.32 GB/s per core per channel. Fig. 3(b) illustrates how the 32 cores on the right would consume 64 GB/s of bandwidth from each of the two memory stacks on the left, plus another ∼10 GB/s from each memory channel. This implies that the underlying network on chip (NoC) must support 148 GB/s across this bisection line to not become a bottleneck. Furthermore, the traffic from the cores on the left will be symmetric, so the interconnect must independently support ∼148 GB/s *in each direction* across the bisection. Making matters worse, the estimated bandwidth numbers here are only for raw memory traffic rates; additional bandwidth is needed for cache coherence traffic among the cores.

Real applications are generally not going to demonstrate a perfectly uniform distribution of memory accesses. Fig. 3(c) provides an example at the other extreme. In this case, all of the right-side cores receive *all* of their traffic from the left-side memory resources. As such, the memory traffic across the bisection doubles to ∼298 GB/s (and again, more bandwidth may need to be supported for cache coherence traffic). In practice, a real application's needs will probably fall somewhere between these cases, but the examples serve to illustrate the likely range of bandwidths that the interconnect must support to avoid becoming the bottleneck to memory.

To put these numbers into context, let us assume that the 64 cores in Fig. 3 are connected with a simple 2D mesh network. Assuming 128-bit links operating at 1Gbps, the NoC bisection bandwidth would only be 128 GB/s: not even enough to support the peak bandwidth needs of the ideal uniformly-distributed traffic. Note that going forward, the bandwidth that could be provided by the stacked memory will likely only increase from a combination of wider stacked-DRAM buses, an increase in the number of channels per stack, faster data transfer speeds, and the integration of more stacks per interposer. All of these place increasing pressure on the NoC, which leads to the research problem of designing new NoC architectures that are suitably scalable to support such high bandwidths, that operate effectively in a multi-chip interposer organization, while not being prohibitively expensive (e.g., power, area) to implement.

3. OPEN RESEARCH CHALLENGES

A scalable interconnect is a necessary component for an interposer-based system to fully take advantage of the high-bandwidth of multiple in-package memory stacks. The challenges span both technology-related and architecture issues, of which both types are discussed in this section.

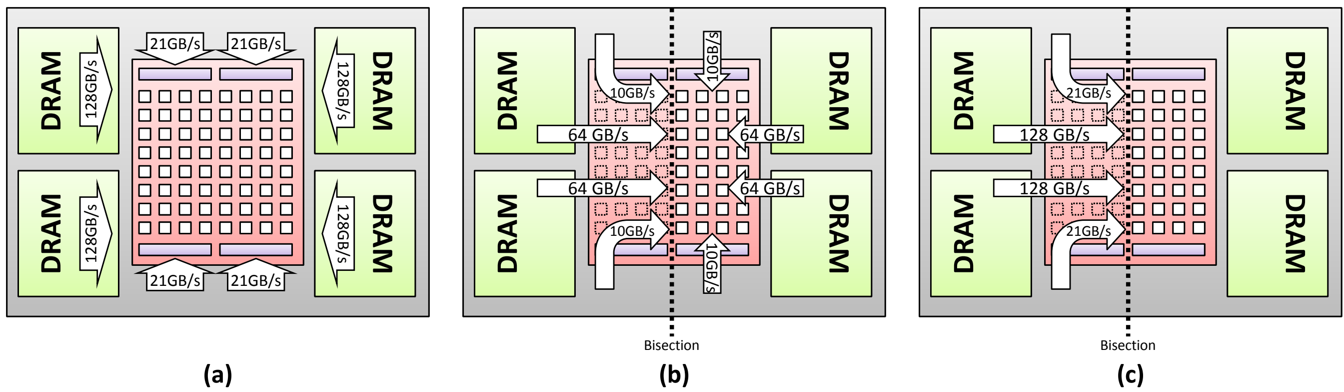

Figure 3: (a) An example processor-and-memory system with four 128 GB/s memory stacks, and four channels of 21 GB/s external memory. (b) Memory usage scenario where all cores access all memory sources uniformly. (c) Memory usage scenario where all cores on the right of the bisection line access only memory from the other side of the bisection.

3.1 Technology Challenges

Recent work has proposed extending the NoC to the interposer itself to provide a more scalable interconnect solution [9]. This can be attractive as it enables the NoC to make use of the routing resources on the interposer layer.

Passive vs. Active Interposers:

In the near term, silicon interposers will likely all be "passive", meaning that they provide metal interconnects (wires) to connect the processor to the memory stacks, but the interposer itself does not provide any "active" devices (i.e., transistors). In the longer term, the interposer layer could conceivably also incorporate active devices[4], especially if the interposer is implemented in an older/cheaper technology and perhaps adequate redundancy is incorporated to address potential yield challenges from building such a relatively large piece of silicon.

Implementing the NoC with a passive interposer enables the NoC to make use of the additional wiring resources of the interposer layer to either relieve routing congestion on the processor die, or to provide more total routing bandwidth without adding more metal layers to the processor. However, active components (i.e., network routers, buffers, repeaters) still must reside on the processor die, and so there are new challenges in organizing the network topology in a meaningful way that takes advantage of these additional wires on the interposer while working with the constraint that all logic remains on the processor die. The additional active components also consume more area and energy on the processor die.

With active interposers, the possibilities are greater in terms of truly 3D topologies with routers and wiring on both processor and interposer layers. For example, a 3D mesh could be implemented across the layers as shown in Fig. 4(a). However, active interposers also introduce new challenges to the NoC implementation. First, as mentioned earlier, the interposer may be implemented in an older technology generation to be cost effective. However, this means that different portions of the overall NoC will be implemented with different device characteristics. This creates new circuit-level design challenges in terms of making all of these disparate components operate at the same speed (otherwise in the worst case, the entire network must operate at the speed of the slowest component). With the passive interposer, all devices are on the same layer and therefore all have the same performance characteristics (modulo within-die parametric variations). Operating devices from different technology generations may also induce other power-related challenges, as the older devices may need to be run at a higher, less-efficient voltage to meet a speed target, which in turn requires power distribution of two different voltage levels. There could also be additional power and latency overheads due to voltage step-up/step-down converters when crossing voltage domains.

[4]With an active interposer, the system is effectively using true 3D stacking. The distinction of an "active interposer" is still useful as conventional 3D stacking typically invokes a vision of a single vertical chip stack, whereas an active interposer plays the structural role of a common integration substrate that happens to also support active devices.

Multiple Processor Chips:

While the previous examples have illustrated a single monolithic multi-core chip stacked on an interposer, another approach is to stack multiple processing chips on the interposer. For example, instead of one 64-core chip, four smaller 16-core chips could be mounted to provide the same compute capability as shown in Fig. 4(b). The smaller chips would likely be cheaper due to better yield (assuming adequate known-good-die testing prior to stacking on the interposer). This, however, further complicates the design of the NoC, as now even some core-to-core communications must navigate their way across the interposer. This can lead to asymmetric NoC topologies, such as that shown in Fig. 4(b), with more complex routing and greater susceptibility to network hotspots.

While all four processing die in Fig. 4(b) may be implemented in the same process technology, die-to-die parametric variations may necessitate additional timing and/or voltage margins leading to less efficient or lower-performance operation.

Clocking/Timing:

Even for a large monolithic CPU chip, trying to run the entire NoC on the same high-speed clock can be very challeng-

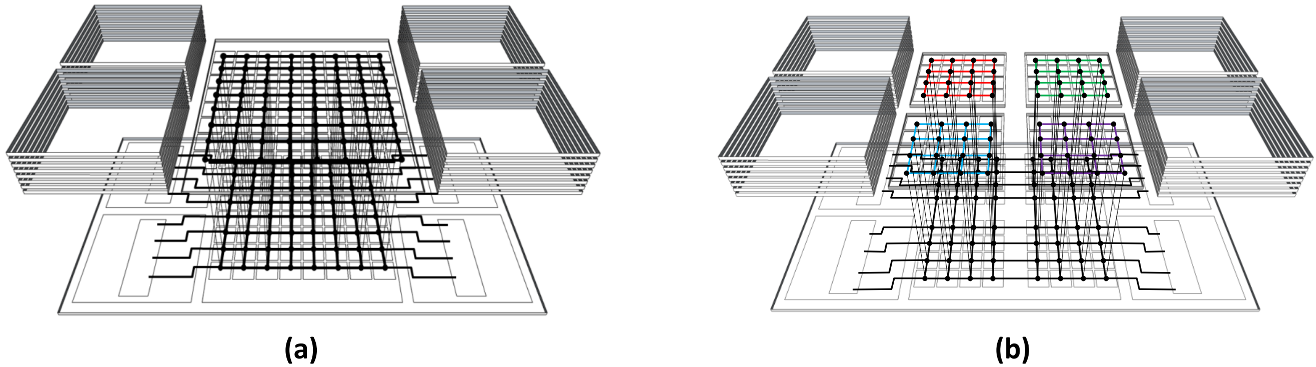

Figure 4: A system with four memory stacks on an interposer along with 64 CPU cores implemented as (a) a monolithic 64-core chip, and (b) four separate 16-core chips. The figure includes example NoC topologies.

ing. The entire clock network must be carefully designed and balanced to minimize clock skew and jitter, and the power consumption is likely non-trivial. Doing this across one or more processing die, as well as possibly an active interposer, would be even harder. The clock network must also deal with the die-to-die-to-interposer parametric variations, along with possibly different process technology generations between the processor layers and the interposer.

One possibility is to decompose the NoC into smaller, independently clocked domains, so that each subnet can have more efficient and easier-to-control local timing. The decomposition allows applying dynamic voltage and frequency scaling (DVFS) to the individual subnets (rather than adjusting the entire NoC at once). However, this would also need to be accompanied by sophisticated power management that, for example, ensures that a subnet operates at a sufficiently high DVFS operating point to prevent it from becoming a bottleneck.

Another downside is that any requests crossing between subnets will now face additional latencies due to the need for synchronizing buffers to safely cross between the different voltage and timing domains. Even if the latency is tolerable, the synchronizers themselves must be designed to keep up with the very high bandwidths required of the NoC.

Complexity, Engineering, Tools:

Implementing a NoC across multiple distinct pieces of silicon introduces new engineering and design complexities that would not be present with a monolithic, single-die NoC. However, this is fundamentally no different than previously proposed 3D (non-interposer) NoC and interconnect designs [12, 13, 15], or for that matter any 3D-partitioned circuits and processors [5, 23, 29]. Design tools are needed to enable engineers to implement and verify NoCs that span multiple chips, possibly using multiple process technologies. This introduces new challenges for circuit timing analysis, area and floorplanning tools, automated place and route, power modeling, etc. Again, these are fundamental challenges that must be overcome for any truly 3D design, but industry and academia are making progress on them, and in specific domains (in particular memory) the tools have gotten far enough along to enable a variety of 3D circuits to be successfully built. For the specific function of NoCs, the problem may be simpler as the number of distinct circuits (e.g., routers, buffers, arbitrators) is relatively limited with well-defined interfaces, and so complexity, tool requirements, and testing/verification procedures to only support NoCs may be more manageable.

3.2 Architecture Challenges

Basic Performance Issues:

An interposer-based interconnect architecture that services the high levels of bandwidth from the in-package memories must provide high bandwidth and low latency. In terms of bandwidth, the use of the interposer should be strictly better than a conventional 2D NoC; in the worst case, if no traffic makes use of the interposer then performance converges back to the 2D case. In the typical case, the additional routing resources on the interposer layer provide additional bandwidth above and beyond the conventional NoC, and so overall bandwidth to memory is increased. Latency could potentially be a challenge as naively adding more routers/nodes on the interposer layer could increase message hop counts (i.e., hops in the Z direction that were unnecessary with a conventional 2D NoC). Different NoC topologies, routing strategies, etc. (discussed more below) can potentially help.

While one may argue that providing high levels of bandwidth on chip is a much simpler problem them the conventional off-chip memory bandwidth problem, conventional off-chip memories have not provided the same levels of bandwidth that we will be seeing from their 3D-stacked counterparts. The in-package memory may provide one or two orders of magnitude more memory, which is a significantly different design point compared to that targeted by conventional on-chip networks. That said, the NoC, especially across the interposer and tying together multiple chips, starts to look a bit more like conventional cluster-level/supercomputer networking, and revisiting and adapting past techniques (topologies, routing algorithms, etc.) from the networking literature is a promising avenue to find solutions for interposer-based NoCs.

NoC Topologies:

Architectural challenges lie in the specific design of the NoC. Ensuring adequate bandwidth to memory such that the in-

terposer network does not become a bottleneck is a primary concern for the topology. Once bandwidth has been satisfied, latency should also be considered to improve memory access times. A conventional multi-core NoC already exists to facilitate cache coherence and on-chip data sharing on the processor die. The interposer can be used to supplement the processor NoC, for example by extending the NoC topology to the interposer as shown in Fig. 4.

Distributing the NoC across the interposer and processor die(s) provides new opportunities to better utilize the collective NoC resources. For example, traffic can be functionally partitioned between the interposer and processor networks, with the interposer portion routing traffic only between cores and memory. With traffic-based partitioning, this then leads to specialization of the sub-networks. A suitable topology to facilitate center-to-edge traffic between cores and memory will likely be different from topologies designed for common any-to-any cache coherence communication patterns in multi-core chips. The core-to-memory traffic patterns routed through the interposer also lend themselves to indirect NoC topologies.

The design of the overall NoC topology must also deal with different physical boundaries and their respective limitations. Vertical communications between the processor die and the interposer may need to exploit concentration to mitigate the reduction in the available bandwidth per core due to μbump overheads. Concentration can also have the positive side-effect of reducing network diameter which improves average latency.

The interposer NoC topology should be scalable in a straightforward manner to a system with more memory stacks and external memory channels, as well as a processor die with more cores or an increasing number of individual processor chips as shown previously in Fig. 4(b). It is a wide-open research problem to explore and devise new topologies that can meet the NoC bandwidth and latency needs while coping with the physical constraints of a multichip die-stacked organization.

NoC Routing Algorithms:

Depending on the selected topology for the interposer and processing chip(s), there may be new challenges (or opportunities) with regard to NoC routing algorithms. While basic routing algorithms can follow naturally from the chosen topologies (e.g., deterministic routing algorithms in many indirect topologies are generally simple to implement), if the routing algorithm does not fully exploit the path diversity of the topology, bisection bandwidth of the NoC may become a bottleneck. There are open problems in designing routing algorithms across the collection of sub-networks with different topologies that simultaneously take advantage of the available path diversity while still ensuring deadlock freedom. The traffic patterns of the memory traffic may offer some opportunities to simplify routing approaches to preventing deadlock. Alternatively, a strict separation of memory versus coherence traffic between dies may result in lost opportunity. Dynamically routing and load balancing across the interposer and core networks can increase resource utilization leading to higher throughput and better overall performance, but then dynamic routing increases complexity in terms of the routing algorithm, congestion detection, and deadlock avoidance.

Chip Topologies:

The examples used so far in this paper have assumed an organization that places the memory stacks on the left- and right-hand sides of the interposer. However, system designs that surround the processor die(s) with memory stacks on, for example, all four sides may also be feasible. The topology for such a system will need to balance east-west and north-south resources to avoid bottlenecks; however, the general approach of using different per-die NoC topologies should still apply to alternative chip/stacking organizations.

The NoC topology must consider the overall floorplanning of the chips on the interposer. Size and placement of the processing dies relative to the memory stacks must be considered. For example, memory stacks could be interleaved with CPU chips across the interposer; this organization would dramatically alter the traffic patterns observed by the NoC and lead to novel topology designs. The placement of external memory ports and memory controllers will impact the design of both the core and interposer NoC subnetworks. Interaction between memory stacks and external memories should also factor into topological and routing decisions. There are many interesting research problems that need to be explored in co-designing the overall chip-level organization with the supporting multi-chip NoC.

Heterogeneous Computing:

Heterogeneity in the processing die(s) through the incorporation of CPUs, GPUs, or other accelerators will place additional demands on the interposer interconnect. Fundamentally, many of these issues are the same as in a conventional 2D chip. These heterogeneous systems require an interconnect with quality of service (QoS) mechanisms that are capable of supplying data efficiently to both the latency-sensitive CPU and the bandwidth-intensive GPU. With increasing heterogeneity, increasing levels of service and variations in traffic behavior will be seen. However, compared to conventional 2D systems, in systems with multiple chips and an interposer, along with multiple timing domains, the coordination of QoS guarantees may become significantly more challenging. QoS mechanisms should also be coordinated between the interconnect and the memory scheduler. How to best design the NoC to support the diverse needs of integrated heterogeneous computing systems, while coping with the multi-chip and interposer organization, is a wide open research topic.

Power and Thermals:

The power consumption challenge for supporting multiple DRAM stacks has two primary components. First, the absolute size of the NoC in terms of total number of nodes may be greater than in an equivalent baseline 2D NoC. For example, the processor chip(s) in Fig. 3 has 64 routers, whereas when one considers both the processor chip(s) *and* the interposer, the total is now about 128 routers. Even with alternative, lower-diameter topologies for the interposer layer [9], the total router count still exceeds that of the original processor chip(s). The immediate impact is that the idle power consumption of the NoC increases due to static/leakage power as well as all of the clocking overheads in a larger NoC. The second aspect is that a NoC with bandwidth matched to support multiple DRAM stacks provides a much higher level of total throughput than today's NoCs, which would result

in commensurately greater power consumption. While the NoC is not typically the dominant power consumer in the overall system, this move to significantly higher bandwidths would increase the importance of power optimizations for future NoCs [4, 6, 17, 26], especially as current and future systems operate under a power cap, such that every Watt spent on the NoC is a Watt taken away from compute (performance).

The thermal concerns of stacking one or more processor chips on top of an interposer are not likely to be as serious. Even with an active interposer, the additional NoC power stacked with the processor will likely still be thermally manageable. Past studies suggest that even with true 3D-stacking of chips, thermals can be managed [8,21]. That is not to say that the effective thermal management for die-stacked systems does not introduce additional engineering challenges, but that solutions exist (albeit perhaps at a higher cost). Additional research targeted at reducing the power in the NoC, and in particular in ways that are thermally friendly, can still be beneficial in terms of reducing the aggressiveness of the required cooling solution for a system.

4. SUMMARY

In this position paper, we have described different aspects of integrated processor-memory systems using silicon interposers. While integrated memory may offer the promise of conquering the Memory Wall, the full potential of in-package memory may be left unrealized if adequate NoC capabilities cannot be provided. Near-term systems may not yet integrate enough memory to fully stress the NoC substrate, but as systems continue to scale to include more stacks of in-package memory, as the per-stack bandwidth continues to increase, and as the integration of the individual computation components increases in complexity (e.g., through multiple smaller processing chips and/or through a heterogeneous mix of processing chips), architecting an effective NoC solution will very quickly become an incredibly challenging problem.

We have described and discussed some of the most pertinent challenges of designing an interposer-based NoC in support of aggressive memory integration. We hope that this position paper has provided a compelling case for the architecture research community to further investigate novel system designs based on memory-on-interposer organizations. While we have focused on the challenge of supplying adequate interconnect bandwidth to support the in-package memory, the broader approach of interposer-based integration leaves many rich research directions to be explored.

Acknowledgments

This work was supported in part by the Natural Sciences and Engineering Research Council of Canada.

AMD, the AMD Arrow logo, and combinations thereof are trademarks of Advanced Micro Devices, Inc. Other product names used in this publication are for identification purposes only and may be trademarks of their respective companies.

5. REFERENCES

[1] Advanced Micro Devices, Inc. AMD Ushers in a New Era of PC Gaming with Radeon^{TM} R9 and R7 300 Series Graphics Line-Up including World's First Graphics Family with Revolutionary HBM Technology. Press Release from http://www.amd.com, June 16, 2015.

[2] K. Athikulwongse, A. Chakraborty, J.-S. Yang, D. Z. Pan, and S. K. Lim. Stress-Driven 3D-IC Placement with TSV Keep-Out Zone and Regularity Study. In *Intl. Conf. on Computer-Aided Design*, pages 669–674, San Jose, CA, November 2010.

[3] B. Black. Die Stacking is Happening. In *Intl. Symp. on Microarchitecture*, Davis, CA, December 2013.

[4] L. Chen and T. M. Pinkston. Nord: Node-router Decouple for Effective Power-gating of On-chip Routers. In *Intl. Symp. on Microarchitecture*, pages 270–281, Vancouver, BC, December 2012.

[5] J. Cong, A. Jagannathan, Y. Ma, G. Reinman, J. Wei, and Y. Zhang. An Automated Design Flow for 3D Microarchitecture Evaluation. In *11th Asia South Pacific Design Automation Conference*, pages 384–389, Yokohama, Japan, January 2006.

[6] R. Das, S. Narayanasamy, S. K. Satpathy, and R. Dreslinski. Catnap: Energy proportional multiple network-on-chip. In *Intl. Symp. on Computer Architecture*, pages 320–331, Tel Aviv, IL, June 2013.

[7] Y. Deng and W. Maly. Interconnect Characteristics of 2.5-D System Integration Scheme. In *Intl. Symp. on Physical Design*, pages 171–175, Sonoma County, CA, April 2001.

[8] Y. Eckert, N. Jayasena, and G. Loh. Thermal feasibility of die-stacked processing in memory. In *Proceedings of the 2nd Workshop on Near-Data Processing*, December 2014.

[9] N. Enright Jerger, A. Kannan, Z. Li, and G. H. Loh. NoC architectures for silicon interposer systems. In *47th Intl. Symp. on Microarchitecture*, pages 458–470, Cambridge, UK, December 2014.

[10] J.-H. Huang. NVidia GPU Technology Conference: Keynote. Technical report, March 2013.

[11] JEDEC. High Bandwidth Memory (HBM) DRAM. http://www.jedec.org/standards-documents/docs/jesd235.

[12] S. Jeloka, R. Das, R. G. Dreslinski, T. Mudge, and D. Blaauw. Hi-Rise: A high-radix switch for 3D integration with single-cycle arbitration. In *47th Intl. Symp. on Microarchitecture*, pages 471–483, Cambridge, UK, December 2014.

[13] J. Kim, C. Nicopoulos, D. Park, R. Das, Y. Xie, N. Vijaykrishnan, M. S. Yousif, and C. R. Das. A Novel Dimensionally-decomposed Router for On-chip Communication in 3D Architectures. In *34th Intl. Symp. on Computer Architecture*, pages 138–149, San Diego, CA, June 2007.

[14] J.-S. Kim, C. Oh, H. Lee, D. Lee, H.-R. Hwang, S. Hwang, B. Na, J. Moon, J.-G. Kim, H. Park, J.-W. Ryu, K. Park, S.-K. Kang, S.-Y. Kim, H. Kim, J.-M. Bang, H. Cho, M. Jang, C. Han, J.-B. Lee, K. Kyung, J.-S. Choi, and Y.-H. Jun. A 1.2V 12.8GB/s 2Gb Mobile Wide-I/O DRAM with 4x128 I/Os Using TSV-Based Stacking. In *ISSCC*, 2011.

[15] F. Li, C. Nicopoulos, T. Richardson, Y. Xie, V. Narayanan, and M. Kandemir. Design and Management of 3D Chip Multiprocessors Using Network-in-Memory. In *33rd Intl. Symp. on Computer Architecture*, pages 130–141, Boston, MA, June 2006.

[16] G. H. Loh, N. Jayasena, K. McGrath, M. O'Connor, S. Reinhardt, and J. Chung. Challenges in Heterogeneous Die-Stacked and Off-Chip Memory Systems. In *SHAW-3*, 2012.

[17] H. Matsutani, M. Koibuchi, D. Ikebuchi, K. Usami, H. Nakamura, and H. Amano. Performance, Area, and Power Evalutations of Ultrafine-grained Run-time Power-gating Routers for CMPs. *IEEE Transactions on Computer Aided Design*, 30(4):520–533, April 2011.

[18] M. Meswani, S. Balgodurov, D. Roberts, J. Slice, M. Ignatowski, and G. Loh. Heterogeneous memory architectures: A HW/SW approach for mixing die-stacked and off-package memories. In *21st Intl. Symp. on High*

Performance Computer Architecture, pages 126–136, San Francisco, CA, February 2015.

[19] S. Nussbaum. AMD ”Trinity” APU. In *24th Hot Chips*, Stanford, CA, August 2012.

[20] C. Okoro, M. Gonzalez, B. Vandevelde, B. Swinnen, G. Eneman, S. Stoukatch, E. Beyne, and D. Vandepitte. Analysis of the Induced Stresses in Silicon During Thermocompression Cu-Cu Bonding of Cu-Through-Vias in 3D-SIC Architecture. In *the Electronic Components and Technology Conference*, pages 249–255, Reno, NV, May 2007.

[21] R. Patti. How 3D is changing the machine. In *Proceedings of the 2nd Workshop on Near Data Processing*, December 2014.

[22] J. T. Pawlowski. Hybrid Memory Cube: Breakthrough DRAM Performance with a Fundamentally Re-Architected DRAM Subsystem. In *Hot Chips 23*, 2011.

[23] K. Puttaswamy and G. H. Loh. Scalability of 3D-Integrated Arithmetic Units in High-Performance Microprocessors. In *44tth Design Automation Conference*, pages 622–625, 2007.

[24] A. Raghavan, Y. Luo, A. Chandawalla, M. Papefthymiou, K. P. Pipe, T. F. Wenisch, and M. M. K. Martin. Computational sprinting. In *18th Intl. Symp. on High Performance Computer Architecture*, 2012.

[25] E. Rotem, A. Naveh, A. Ananthakrishnan, D. Rajwan, and E. Weissmann. Power-Management Architecture of the Intel Microarchitecture Code-Named Sandy Bridge. *IEEE Micro Magazine*, 32(2):20–27, March–April 2012.

[26] A. Samih, R. Wang, A. Krishna, C. Maciocco, and Y. Solihin. Energy-efficient interconnect via router parking. In *19th Intl. Symp. on High Performance Computer Architecture*, pages 508–519, 2013.

[27] K. Skadron, M. R. Stan, W. Huang, S. Velusamy, K. Sankaranarayanan, and D. Tarjan. Temperature-Aware Microarchitecture. In *30th Intl. Symp. on Computer Architecture*, pages 2–13, San Diego, CA, May 2003.

[28] Tezzaron Semiconductor. WWW Site. http://www.tezzaron.com.

[29] Y.-F. Tsai, Y. Xie, N. Vijaykrishnan, and M. J. Irwin. Three-Dimensional Cache Design Using 3DCacti. In *Intl. Conf. on Computer Design*, San Jose, CA, October 2005.

[30] W. A. Wulf and S. A. McKee. Hitting the Memory Wall: Implications of the Obvious. *Computer Architecture News*, 23(1):20–24, March 1995.