Seizing the Bandwidth Scaling of On-Package Interconnect in a Post-Moore's Law World

Grigory Chirkov
Princeton University
Princeton, USA
gchirkov@princeton.edu

David Wentzlaff
Princeton University
Princeton, USA
wentzlaf@princeton.edu

ABSTRACT

The slowing and forecasted end of Moore's Law have forced designers to look beyond simply adding transistors, encouraging them to employ other unused resources as a manner to increase chip performance. At the same time, in recent years, inter-die interconnect technologies made a huge leap forward, dramatically increasing the available bandwidth. While the end of Moore's Law will inevitably slow down the performance advances of single-die setups, interconnect technologies will likely continue to scale. We envision a future where designers must create ways to exploit interconnect utilization for better system performance.

As an example of a feature that converts interconnect utilization into performance, we present Meduza – a write-update coherence protocol for future chiplet systems. Meduza extends previous write-update protocols to systems with multi-level cache hierarchies. Meduza improves execution speed in our benchmark suite by 19% when compared to the MESIF coherence protocol on a chiplet-based system. Moreover, Meduza promises even more advantages in future systems. This work shows that by exploiting excess interconnect bandwidth, there is significant potential for additional performance in modern and future chiplet systems.

CCS CONCEPTS

• **Computer systems organization → Interconnection architectures**; **Multicore architectures**.

KEYWORDS

multi-chiplet; interconnect; multicore; write-update; coherence; bandwidth; scaling

ACM Reference Format:
Grigory Chirkov and David Wentzlaff. 2023. Seizing the Bandwidth Scaling of On-Package Interconnect in a Post-Moore's Law World. In *2023 International Conference on Supercomputing (ICS '23), June 21–23, 2023, Orlando, FL, USA.* ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3577193.3593702

1 INTRODUCTION

The approaching demise of Moore's Law [43] has forced chip designers to search for performance by leveraging techniques and resources beyond what can be provided solely due to increased transistor count. By exploring and exploiting resources that are underutilized in current designs or that are on an increasing trajectory, computer architects can continue increasing performance even in the face of stagnant transistor scaling. One promising resource that has traditionally been judiciously conserved, which bears reconsideration in a post-Moore's Law setting, is inter-chip, inter-die, and off-chip interconnect bandwidth.

Indeed, the slowdown of Moore's Law foreshadows the limited computational resources achievable in a single-die setting. At the same time, novel packaging technologies including silicon bridges [51, 85], TSVs [84], interposers [12, 48, 49], organic packages [6, 60] together with superior channel coding standards [42, 60, 65, 75] have collectively enabled increasing inter-chip, inter-die, and off-chip bandwidth. The promise of chip-to-chip optical interconnects [13, 27, 74] may play an even bigger role in increasing future chip-to-chip bandwidth.

Fig. 1 shows these two trends graphically. Their combination will likely lead to a future, where **chips with conventional architectures will not be able to utilize the available channel bandwidth to its full potential**. This context poses an essential question to researchers from both academia and industry: **how can we effectively convert the excessive interconnect bandwidth into better performance in our future systems?**

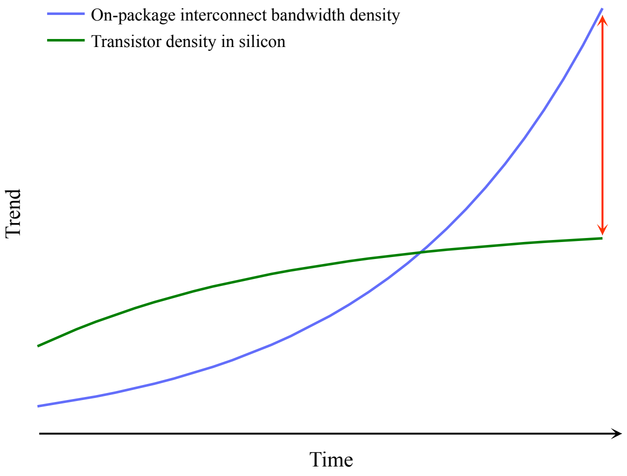


Figure 1: An illustration of the expected future improvements in the interconnect and transistor technologies. The mismatch between inter-chip bandwidth and transistor density advancements is expected to grow as time progresses. Researchers need to start thinking now about how they can convert interconnect bandwidth into systems' performance.

ICS '23, June 21–23, 2023, Orlando, FL, USA
© 2023 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-0056-9/23/06.
https://doi.org/10.1145/3577193.3593702

One of the answers given by the field in the past five years is the extensive usage of accelerators, especially in data centers. However, specialized hardware like TPUs [1], GPUs [45], and FPGAs [70] answer this question only partially. They require wide communication channels for effective data movement to/from the host machine, but they do not help with general-purpose tasks.

Another trend is the rise in the popularity of chiplet designs. Multiple manufacturers have transitioned their top-of-the-line products to chiplets [4, 14, 60]. The concept of chiplets is not new. However, in these new circumstances, this old idea becomes very powerful. New high-bandwidth, low-latency interconnects let these partitioned designs act as one without serious performance degradation. Multi-chiplet systems are a general concept applicable to all sorts of computational mechanisms: CPUs, GPUs, accelerators, etc. In this paradigm, any feature that trades off performance for increased interconnect utilization can be generalized to all hardware designs.

In this work, we present our vision for the future of inter-chip interconnects and chiplet systems. We analyze the current technologies and make predictions about the future of these technologies. We argue that future advances in interconnects are an interesting and important way for engineers to continue the performance scaling of their designs.

As an example of how we can utilize this opportunity, we present Meduza – a write-update coherence protocol for future multi-chiplet systems. Like chiplets, write-update protocols were discarded in the past due to their high bandwidth requirements. And like chiplets, this technology becomes more viable and efficient with modern on-package interconnects. Meduza adapts the existing write-update coherence protocols to work in a multi-level cache hierarchy. It eliminates coherence misses from the last-level caches (LLCs) at the expense of high interconnect usage. The concept of write-update protocols is not new [7]; however, like with chiplet systems, the demise of Moore's Law and the trend of increasing inter-die bandwidth precipitates a re-evaluation of cache coherence protocols.

Indeed, the majority of the modern multicore CPUs choose write-invalidate protocols, and in particular, they employ some variation of the MESI [64] protocol as write-invalidate protocols to save cache-to-cache bandwidth. In this work, we critically re-evaluate the choice of using write-invalidate over write-update protocols in the context of modern and future multi-chiplet systems.

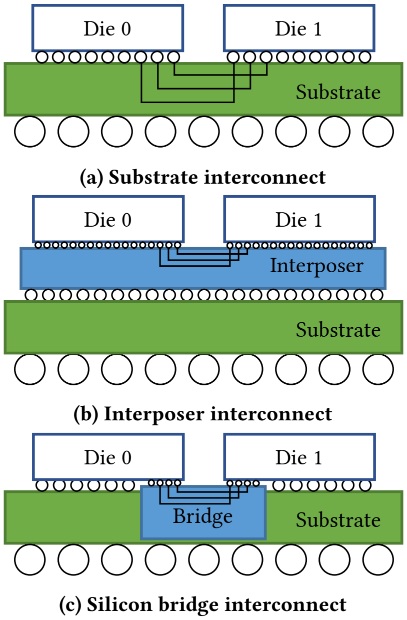
We evaluate Meduza compared to MESIF coherence primarily in the context of the available bandwidth of multi-chiplet systems similar in structure to current AMD EPYC systems. Meduza beats invalidation protocols in our benchmark suite by 19% with current 32 bytes/cycle interconnects and by 22% with future 256 bytes/cycle interconnects.

2 BACKGROUND

2.1 Interconnect Technologies

The end of Moore's Law forces designers to look for performance increases in areas other than transistor density and speed. One such area is inter-chip, inter-die, and off-chip interconnect bandwidth.

In off-chip space, both DRAM and PCIe standards have shown fast scaling in the past, and their roadmaps promise to deliver more in the future [37, 86]. On the one hand, this trend is ensured by the steady development of communication channel characteristics and advanced encoding schemes [75]. On the other hand, interconnect bandwidth is driven by the demands of modern data centers. Accelerators, GPUs, NICs, and NVMe storage are all now common parts of servers [1, 45]. These devices require more and more available data throughput each year, and designers are willing to allocate more system costs to this. For example, Intel used a package with 3647 pins for Skylake servers in 2017, 4189 pins for Ice Lake servers in 2020, and 4677 pins for Sapphire Rapids servers in 2022 [55, 56]. This amount to almost a 30% increase in pin count in five years. Given that these chips have roughly equal power requirements, we can assume that the added pins are used for data transmission and not power delivery. Another indication of this is that Intel has increased the number of PCIe lanes from 48 to 64 and the number of memory channels from 6 to 8. In parallel, during this period, Intel transitioned from PCIe 3.0 to PCIe 5.0 and from DDR4 to DDR5. This illustrates that manufacturers are able and willing to provide more and more data throughput in new chips.



(a) Substrate interconnect

(b) Interposer interconnect

(c) Silicon bridge interconnect

Figure 2: Three categories of on-package interconnects

However, the most significant trend in the domain of chip communication in recent years is the development and proliferation of on-package interconnects. Their biggest difference from off-chip connections is much higher bandwidth and lower communication latency. The academia-proposed on-package silicon photonics can enhance interconnect properties even more radically in the future [28]. These interconnects, however, come with a higher production cost and stricter requirements for the distance between connected chips [59]. They fall broadly into three different categories.

2.1.1 *Substrate Interconnect.*

In chips with substrate interconnect, the dies are connected to each other using a high-density redistribution layer (RDL) inside the substrate to create Multi-Chip Modules (MCMs) [59]. The high-level organization of such chips is shown in

Fig. 2a. Substrate interconnect is the cheapest of the three technologies because it does not require additional layers for connectivity like interposers or bridges. The downside is the relatively low wire density.

AMD uses organic substrate interconnects in their EPYC and Ryzen CPU lineups in combination with Infinity Fabric On-Package (IFOP) that was designed specifically for on-package connections [60]. Together, they provide 6.4 Gb/s per pin, with an energy consumption of 2 pJ per bit.

2.1.2 Interposer Interconnect. Interposers are used in so-called 2.5D stacking [39]. Its high-level idea is shown in Fig. 2b. Multiple dies are put on top of an intermediate high-density RDL interposer, which is in turn located on top of a chip substrate. The interposer can be made from various materials, including silicon, organic material, or glass [44]. Moving data signals to a separate routing level help achieve a lower wire pitch. Other connections, including power and ground, are routed directly to dies through the interposer using through-interposer vias (TIVs), or through-silicon vias (TSVs) in the case of silicon interposers.

Silicon interposers are an important subclass of interposers. In this technology, RDL is formed inside another die during tape-out. This die can be either active (with device layers) or passive. Typically, it is made passive and is manufactured in an older technology node (e.g. 65nm) to increase the yields [39]. Silicon interposers are important because signals routed through silicon have even lower wire pitch and therefore allow for higher densities and bandwidths.

TSMC has two versions of interposer technology: Chip-on-Wafer-on-Substrate (CoWoS) [12] for the "chip-last" assembly flow and Integrated Fan-Out (InFO) [49] for the "chip-first" assembly flow. In conjunction with TSMC's Low-voltage-IN-Package-INterCONnect (LIPINCON) [48], CoWoS can provide data rates up to 8 Gb/s per pin, 1.6 Mb/s per squared micrometer, with energy consumption of 0.56 pJ per bit, and InFO achieves bandwidth up to 2.8 Gb/s per pin, 0.3 Mb/s per micrometer squared, and energy consumption of 0.42 pJ per bit [49].

2.1.3 Silicon Bridge Interconnect. The biggest downside of 2.5D stacking is the fact that all active dies must be located on top of the interposer. This means that the total area of the active dies must be smaller than the interposer's area, limiting the amount of computation achievable with 2.5D stacking. Moreover, this also means that the interposer must have a large area to accommodate all active dies, making it an expensive approach.

One of the ways of dealing with this problem is using localized interposers, called bridges [85]. A high-level diagram of a system with a silicon bridge is shown in Fig. 2c. Instead of using one large monolithic interposer, a couple of smaller interposers are located only in places where they are needed. This approach enables removing the limit on the total active die area of the chip and cuts down manufacturing costs.

Intel's version of silicon bridge is called Embedded Multi-die Interconnect Bridge (EMIB) [32]. A combination of EMIB with Intel's Advanced Interface Bus (AIB) [42] can provide up to 2 Gb/s of bandwidth per wire, 1.5 Mb/s per micrometer squared, with energy consumption of 0.85 pJ per bit.

TSMC's competitors for EMIB are InFO-L for "chip-first" assembly and CoWoS-L for "chip-last" assembly [85]. Currently, there is no information available about their bandwidth and energy characteristics.

All the technologies described in this section enable the throughputs not achievable before and promise to deliver more performance in the future [66, 85].

2.2 Multi-chiplet Design

With Moore's Law approaching its predicted end, it is becoming more and more challenging for chip designers to increase the performance of the circuits and the number of transistors at the same rate as was achieved in the last 40 years. For example, Intel's new advanced 10nm node was delayed multiple times and started large-scale chip fabrication only in 2019, three years later than the company initially promised to its investors [33]. This caused a massive delay in the release of the Ice Lake architecture to the market. Another example of such issues is the decision of GlobalFoundries to stop the development of smaller feature sizes altogether [3].

In these circumstances, chip designers are forced to exploit new, sophisticated designs to allow for higher transistor counts without chip cost increase. One of these popular designs is the multi-chiplet design, where the circuit is implemented across multiple smaller dies, called chiplets. Smaller die size provides higher yields and lower chip costs. Moreover, the multi-chiplet design allows fabricating chips with a total silicon area larger than is possible with monolithic chip architecture due to the reticle size limit. Multi-chiplet architecture is not a new idea, but modern, on-package interconnects with lower latency, energy, and congestion penalties of moving data between different dies make it more feasible and effective now.

Multiple modern designs use a multi-chiplet approach, including:

- Intel's Sapphire Rapids, Ponte Vecchio, and Stratix 10 platforms use EMIB [14, 15, 32]
- AMD Instinct GPUs use AMD's proprietary silicon bridge technology called Elevated Fanout Bridge (EFB) [76], which looks similar to TSMC's InFO-L
- AMD EPYC and Ryzen CPUs use organic substrate interconnects [59]
- Apple M1 Ultra SoCs use their proprietary UltraFusion technology [4]
- IBM Telum CPUs have a dual-die design with undisclosed interconnect technology [36]
- Nvidia proposed the use of organic substrate interconnects for multi-chiplet GPUs [6]

The biggest downside of this approach is the inter-die latency. There still exists some penalty for crossing die boundaries, and the data movement does not happen as fast as inside monolithic silicon [39, 50]. AMD EPYC CPUs are a good demonstration of this. The core-to-core latency between cores on the same chiplet is about 20 ns and about 140 ns for cores on different chiplets [24].

2.3 Write-Update Coherence Protocols

A majority of the modern multicore CPUs employ some modification of the MESI (Illinois) [64] coherence protocol. This is a write-invalidate protocol [78]: on a write to a shared cache line, all other copies are invalidated. The opposite behavior would be to update all the other copies with the new value. Protocols like

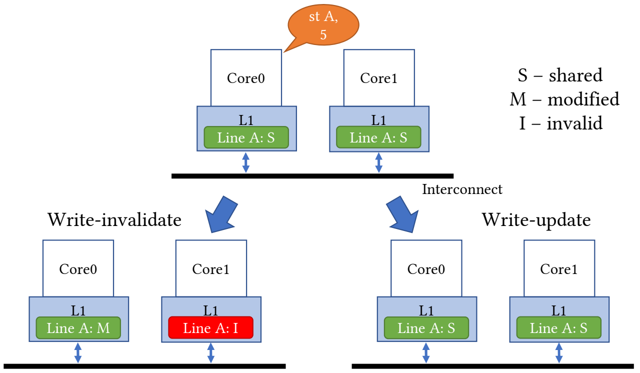


Figure 3: State transitions that happen during a write to a shared line. Write-invalidate protocols invalidate copies in other caches, and write-update protocols broadcast changes to other caches.

this are called write-update. Fig. 3 shows the transitions that happen during memory writes in write-invalidate and write-update systems. Write-update protocols utilize significantly more interconnect bandwidth to eliminate coherence misses from the system. This works especially well in systems with constant data migration, such as applications written in a producer-consumer paradigm [78].

The two original write-update protocols are Firefly [82] and Dragon [7]. Newer write-update protocols make changes to adapt this paradigm to multi-level caches [31] or ccNUMA systems [21, 40, 41]. Most of the new works in the space focus on conserving the used bandwidth by making the protocol either adaptive or hybrid [2, 23, 25, 26, 30, 61, 67]. Ultimately, write-update protocols never achieved wide adoption because of their prohibitively high interconnect bandwidth demands.

3 HIGH-BANDWIDTH INTERCONNECTS FOR PERFORMANCE SCALING

Considering the approaching end of Moore's Law, it becomes clear that researchers and designers must look for ways to increase performance that does not depend on transistor density in future chips. At the same time, the Heterogeneous Integration Roadmap (HIR) [18] from IEEE Electronics Packaging Society (EPS) predicts that the proliferation of high-bandwidth on-package interconnects described above will continue in the future. They argue that the bandwidth density of these interconnects will double or triple every two years, creating an equivalent of Moore's Law for on-package interconnects. Intel and TSMC have similar plans for their interconnect technologies [66, 85]. The promise of on-package photonic interconnects [28] makes us even more confident that the on-package interconnect bandwidth will not be a problem in the future. From a computer architecture perspective, this means that there will be an eventual mismatch between the amount of available inter-chip bandwidth and the ability to utilize it. Therefore, we argue that **researchers should start thinking now about ways to convert the on-package interconnect throughput into performance increase**.

There is ultimately no single answer to this question. For example, a possible solution is the usage of specialized accelerators. In this case, the interconnect bandwidth is used to move data to the more optimal computational engines. Another example is multi-chiplet architecture: interconnects are used to allow a higher total core count per chip. Multi-chiplet architecture is also an example of a previously discarded idea, becoming more feasible and valuable in a modern setting with on-package interconnects.

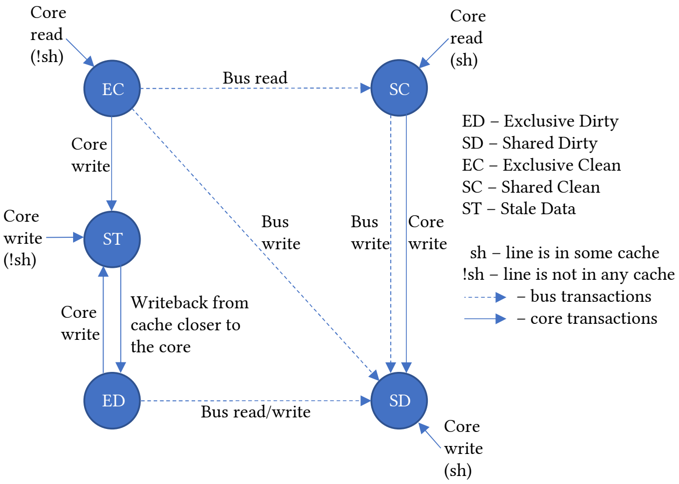


Figure 4: State transition diagram for Meduza coherence protocol. Compared to Dragon, Meduza introduces the Stale (ST) state. The most recent updates to the cache line in this state have not yet reached the LLC. During the bus read, a writeback precedes the request execution.

Likewise, write-update protocols have been discarded before due to their high bandwidth demands and low impact on performance in single-die SoCs. However, they are a perfect candidate for multi-chiplet systems. At the expense of increased inter-die channel utilization (which we previously discussed as growing for future systems), it can eliminate coherence cache misses and remove the data movement between coherent caches from a memory request's critical path. This can potentially have a huge impact in multi-chiplet systems, where there is a relatively high latency penalty for crossing the die boundary.

3.1 Meduza Coherence Protocol

The rest of this section describes Meduza - a write-update coherence protocol for future multi-chiplet systems. This is our example of how engineers can convert interconnect bandwidth into system performance. Meduza is built on top of the Dragon [7] protocol to support coherence in multi-level cache systems. Meduza is neither adaptive nor hybrid for the reason described above: future multi-chiplet systems do not need to preserve bandwidth in inter-die links. Fig. 4 shows the state transition diagram of the Meduza protocol.

Meduza deals with three main challenges for write-update protocols used in the multi-level cache hierarchy. The first challenge is the consistency model of the system. Meduza employs a 2-phase commit (2PC) to support a Total Store Order (TSO) consistency model in the system. During 2PC, the writing cache first obtains exclusive ownership of the cache line to perform a write in the first phase and then propagates the updated data across the system in the second phase. The second challenge arises from cache updates

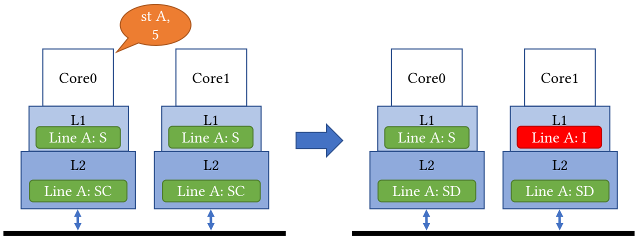


Figure 5: Example of line invalidation happening in the cache closer to the core during an interconnect-side update. Cache line states are defined in Fig. 4.

coming from the interconnect side making data in caches closer to the core stale. Meduza reuses the solution from previous work on multi-level cache coherence [31] and invalidates all caches closer to the core during interconnect side updates. Fig. 5 shows an example of such invalidation.

The third challenge for write-update protocols is the write-back policy. This policy is usually employed in modern CPUs in at least one of the cache levels to save cache bandwidth. The write-back policy is incompatible with write-update protocols because, with write-back, the memory writes do not reach LLC immediately. Without additional modifications to the write policy, new writes are not observed in the other cores, and coherence is not maintained in the system. Previous works assume that all caches except for the LLC employ the write-through policy. However, this design excessively pollutes the write ports of LLCs with unnecessary writes.

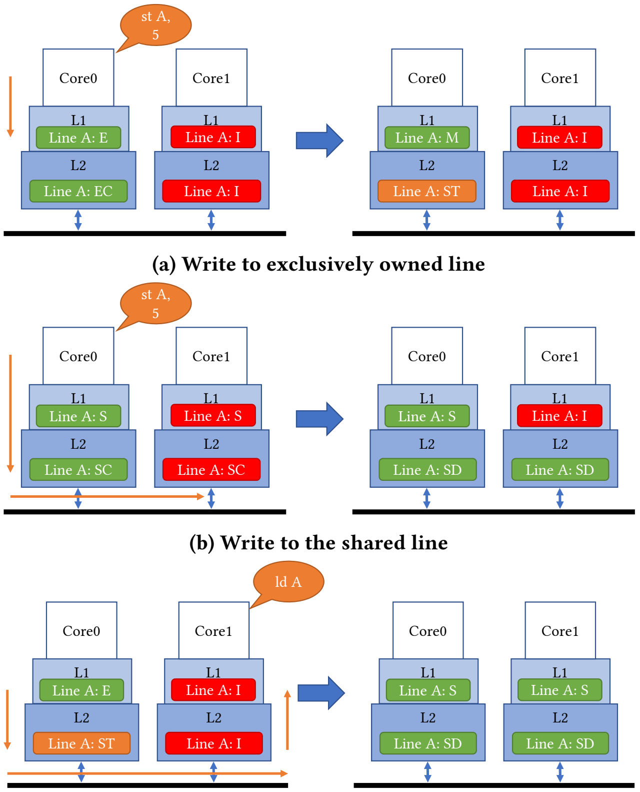
The solution we came up with is to apply a write-through policy **only** to the shared lines. We call this policy *write-select*. Fig. 6 shows examples of this write-select functionality. This approach is similar to the Dynamic Write-Policy mechanism used in VIPS [69]. However, in the case of VIPS, it is used to simplify coherence in clustered cache hierarchies and is not used in the context of write-update protocols.

Write-select allows us to use the resources more efficiently while delivering only needed lines to the LLC. This happens at the expense of an additional Stale (ST) state, which denotes a cache line with stale data. Fig. 4 shows that when another core tries to fetch the line in the Stale state from some other cache, that line is first written back to its LLC and only then propagated further into the requesting core.

The performance of write-through and write-select options is shown in Fig. 7. Write-select policy delivers almost identical performance to the write-through policy. Small performance degradation in particular benchmarks is caused by fetching lines in the ST state (shown in Fig. 6c). At the same time, the number of writes is an order of magnitude lower, and in this regard, write-select behaves much like write-back and allows the system to save large amounts of cache write bandwidth. Therefore, **we chose write-select as the write policy for Meduza**. Readers can refer to Sec. 4 for evaluation details.

4 EVALUATION

This section describes the evaluation details and shows that Meduza is a good fit for multi-chiplet systems and indeed helps achieve better performance by employing more inter-chiplet communication.



(a) Write to exclusively owned line

(b) Write to the shared line

(c) Fetching line in ST state from another cache

Figure 6: Examples of write-select functioning. Orange arrows show the data propagation through the system. Cache line states are defined in Fig. 4.

4.1 Evaluation Details

We evaluate Meduza in a multi-chiplet system with a structure similar to that of AMD EPYC Milan-X [24] chips. It implements a chiplet design: groups of 8 cores are located on different physical dies (Fig. 8); these groups are called Core Complexes (CCX). All CCXes are connected to the Input/Output (IO) die. It routes the data between CCXes, memory controllers, and IO ports. The system directory is located inside the IO die.

This design comes with a cost of relatively high data transmission latency between chiplets. To partially mitigate this effect, cores from the same CCX share one large L3 cache, which allows for a much faster exchange between cores on one CCX. However, this optimization does not help with data movement between cores on different CCXes. In all further experiments, we compare this configuration using the traditional MESIF protocol and the novel Meduza protocol. The evaluated baseline system's parameters are listed in Table 1.

We use the PriME simulator [29] to evaluate the proposed solution. PriME is a parallel simulator similar to Sniper [17] and Graphite [54]. Being a classic parallel simulator, PriME trades off some accuracy (about 12% total average error relative to native execution [29]) in exchange for high execution speed and the ability to model systems with high core counts. PriME can vary many system parameters such as cache sizes, system topologies, latencies, and

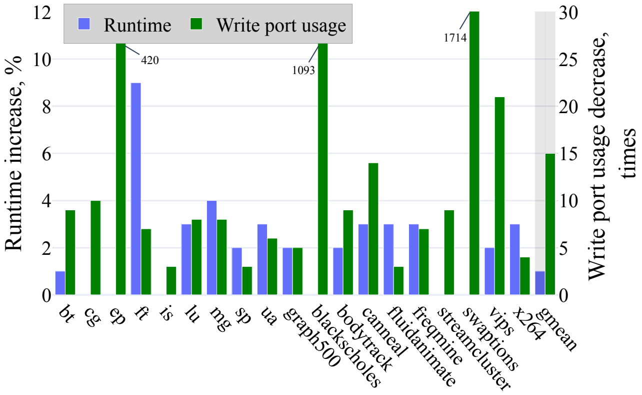


Figure 7: Meduza results with write-select policy relative to write-through. Write-select significantly decreases the number of writes to the LLC at the cost of slight performance degradation.

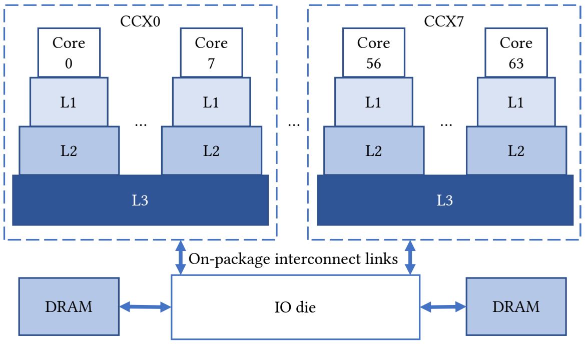


Figure 8: Logical layout of the studied system. Only four out of eight CCXes are shown to preserve space.

throughputs without changing source code or recompilation, allowing us to see the parameters' influences on achieved results. The coherence simulation logic was modified to include a write-update option. This also includes changes for the write-select feature (see Sec. 3). We used benchmarks from three different benchmark suites and formed them into two groups based on their characteristics.

The HPC group consists of the NAS Parallel Benchmark suite (NPB3.4-OMP) [8] and Graph500 benchmark [58]. NPB3.4-OMP is a benchmark suite of HPC applications. It scales well with the number of cores and employs various communication strategies across benchmarks. Graph500 is a benchmark consisting of graph applications. It was specifically designed to stress the memory and communication systems of supercomputers. Because our solution concentrates specifically on inter-core communication in CPUs, we expect this benchmark to perform particularly well with Meduza.

The second group consists of benchmarks from PARSEC 3.0 [87]. PARSEC 3.0 is a general benchmark suite for multicore CPUs. It does not focus on particular execution characteristics and imitates the behavior of various multithreaded programs from both server and client segments. We excluded *dedup*, *ferret*, and *facesim* benchmarks because their simulation requires more memory than our computational system provides.

Table 1: Parameters of the studied system

Parameter	Value
Non-memory IPC	2
Store buffer entries	128
Number of cores	64
# of cache levels	3
L1 latency, cycles	4
L1 size, KB	32
L2 latency, cycles	12
L2 size, MB	0.5
L3 latency, cycles	40
L3 size (per core), MB	12
LLC type	Shared inside CCX
Coherence Protocol	MESIF/Meduza
Interconnect Topology	Crossbar
Interconnect Latency, cycles/hop	150
Interconnect BW, B/c	32
DRAM latency, cycles	300

4.2 Upper-Bound Performance

We first measure the "upper bound" performance of Meduza versus a conventional MESIF protocol. The system in this study uses the baseline parameters, except in the inter-chiplet interconnect links. This study does not model interconnect congestion, effectively simulating links with "unlimited" bandwidth to find an upper bound.

The relative performance of Meduza in this system compared to MESIF is shown in Fig. 9. Meduza reduces the execution time by up to 96%, with a geometric mean performance increase of 22%. As expected, these gains stem from the significant increase in LLC hit rate: +22% on average. It is also worth noting that our simulations show no statistically significant changes in the behavior of L1 and L2 caches.

In general, Meduza provides the best performance when evaluated with HPC applications. A separate analysis of the HPC suite shows a 29% decrease in execution time and a 24% LLC hit rate increase. Other applications show slightly worse results, as the PARSEC benchmarks experience only a 19% increase in LLC hit rate, which translates to a 14% performance increase on average.

The result of the *streamcluster* benchmark stands out in particular because the online clustering problem algorithm involves constant data exchanges between threads when transitioning between iterations. In this case, the Meduza protocol propagates the updates to the shared line immediately after the write and removes the coherence actions from the critical path on the next read. Thus, *streamcluster* is an excellent example of the advantages that Meduza provides over MESIF.

Fig. 10 shows the same upper bound experiment with varying core count. The core count is changed by adding or removing chiplets from the system, making the LLC-to-core ratio constant. The left part of the chart shows the reduction in execution time in all benchmarks. In general, Meduza works better in larger-scale systems. Results vary from a 12% performance increase in a 16-core system to a 22% performance increase in a 64-core system. This trend is seen more clearly when concentrating on HPC applications that scale better: performance scales from +14% for a 16-core system to +29% for a 64-core system. Results for PARSEC benchmarks scale

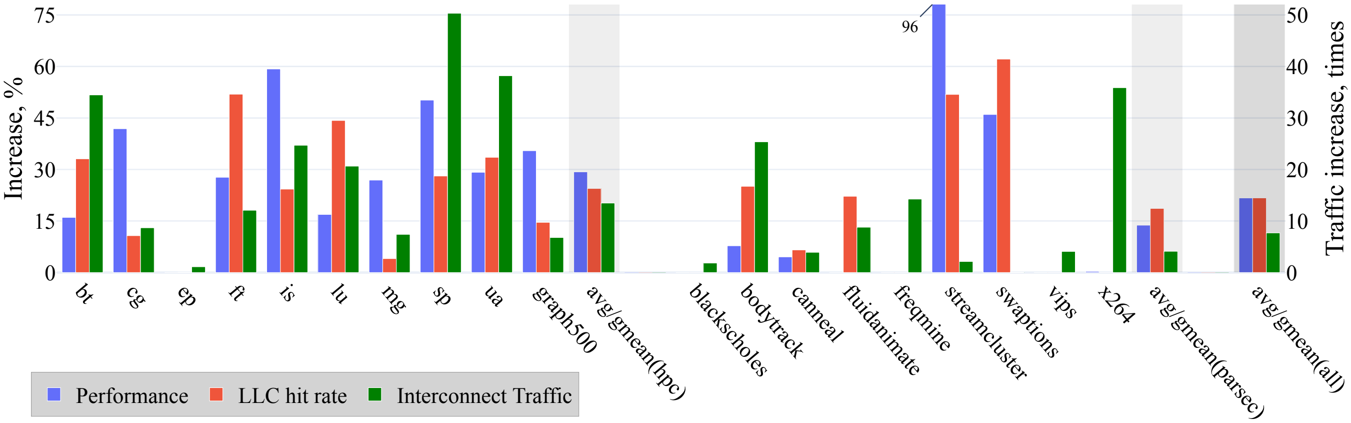


Figure 9: Meduza gain relative to MESIF in "bandwidth-unconstrained" system: the congestion in inter-chiplet links is not simulated. Runtime and LLC hit rate differences are shown in % on the left y-axis, and interconnect traffic difference is shown on the right y-axis. Runtime and traffic are aggregated with the geometric mean, and hit rate - with arithmetic.

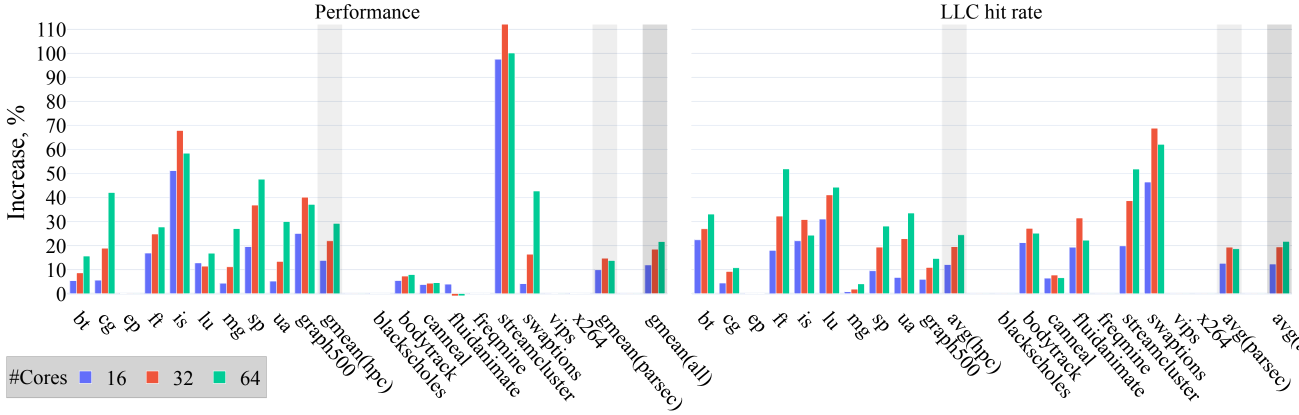


Figure 10: Meduza performance increase relative to MESIF with a varying number of cores. The right part of the chart shows the gains in relative performance, and the left – in LLC hit rate.

worse due to their limited general scalability [35, 77]. The right part of the chart shows the LLC hit rate increase in all benchmarks. Hit rate difference numbers correlate with performance difference numbers. This again confirms that Meduza outperforms MESIF due to a lower number of coherence misses.

There are two main reasons why the performance of Meduza scales with the number of cores. First, bigger systems have larger LLCs, allowing the system to retain more shared data in on-chip memory. In contrast, with a smaller LLC, more data is kept in off-chip memory, and its access time does not depend on coherence protocol. Second, more cores in the system mean more competition for the shared lines and worse performance with write-invalidate.

This experiment clearly shows that the advantage of write-update protocols scales with the system size. As more cores are integrated into future systems, we expect Meduza to provide an even larger advantage over MESIF.

4.3 Performance vs Interconnect Latency

Fig. 11 shows how interconnect latency affects the performance of Meduza when compared to MESIF. The miss-penalty in this study grows proportionally to the interconnect latency, making coherence misses more expensive and MESIF performance worse. For example, the difference between Meduza and MESIF in *streamcluster* benchmark in the system with interconnect latency of 40 cycles equals 60%, and with interconnect latency of 320 cycles equals 114%. On average, these differences equal 9% and 28%, respectively. As described in previous sections, Meduza hides latency between cores by proactively replicating data at the expense of interconnect bandwidth.

In the future, we expect that larger systems will integrate more chiplets [57], and this will increase the average interconnect length and latencies. This means that Meduza's advantage will be even larger in the upcoming systems. In general, write-update protocols help build more disaggregated systems, as long as their interconnects possess enough bandwidth.

4.4 Performance vs Interconnect Throughput

As of this point, all the results were obtained by using an upper-bound model of the interconnects that does not model link congestion. In this subsection, we analyze how the performance of

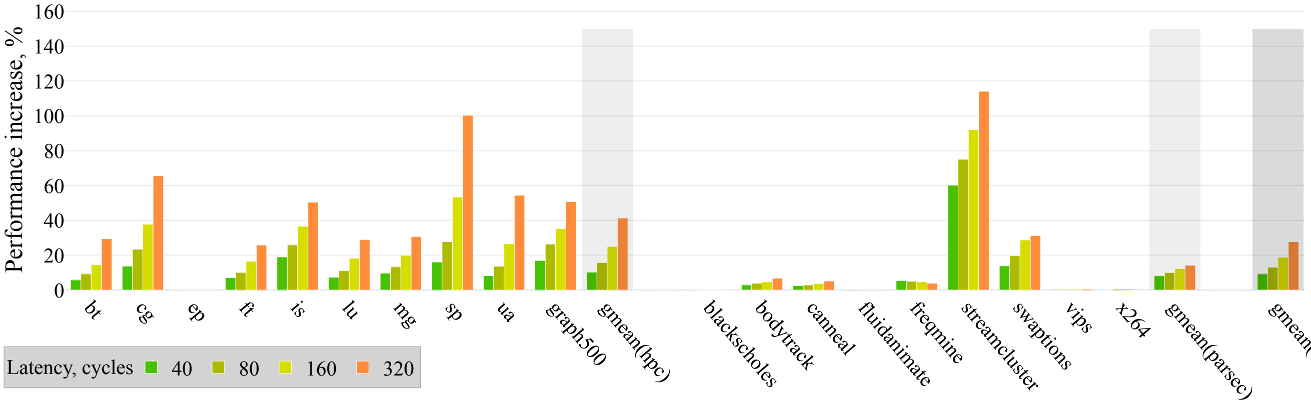


Figure 11: Meduza performance increase relative to MESIF with varying interconnect latency.

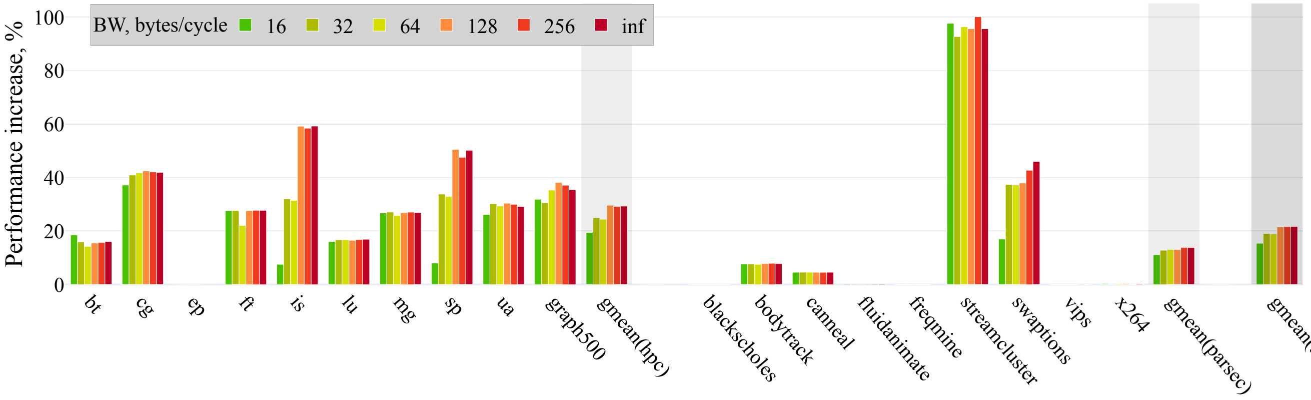


Figure 12: Meduza performance relative to MESIF with varying bandwidth of inter-chiplet interconnect.

write-update protocols changes when using different bandwidth interconnects. We use a simulator built-in congestion model based on queueing theory.

Fig. 12 shows the real-world results with interconnect congestion modeling enabled. The chart shows how the performance of Meduza gradually increases in each benchmark with increasing interconnect bandwidth. For example, in the integer sorting (*is*) benchmark, the difference between Meduza and MESIF is 9 times higher with 128 bytes/cycle interconnect compared to the difference with 16 bytes/cycle interconnect. On average, performance difference increases from +15% with 16 bytes/cycle interconnect to +22% difference with 256 bytes/cycle interconnect. This result is again clearer in the HPC suite than in the PARSEC suite.

It is notable that even with the current inter-chiplet bandwidth of about 32 bytes/cycle available in modern MCM chips like AMD EPYC, write-update shows better performance than write-invalidate. At the same time, more advanced interconnect technologies (like silicon interposers or silicon bridges) can provide more bandwidth density and allow extraction of more performance using write-update protocols. Moreover, an even larger performance advantage can be achieved in the future with further advancements in interconnect technologies and the widening of bandwidth-transistor density disparity fueled by the ending/slowing of Moore's Law.

We find it interesting that **even a 16 bytes/cycle interconnect allows for better performance with Meduza**. Practically, this means that **the on-package interconnects in contemporary systems are underutilized and this presents an opportunity for further optimizations.**

4.5 Interconnect Energy Measurements

To estimate the interconnect energy difference between Meduza and MESIF, we measured the difference in interconnect traffic when using these two protocols (Fig. 9). The estimates below do not include additional energy from increased cache writes.

The system interconnect transmits eight times more data with Meduza than with MESIF on average. Two factors contribute to this: first, repeated writes to the shared lines all initiate data transmission on the interconnect, and second, these transmissions contain the full 64 bytes of the modified line.

We can estimate how much the added traffic affects chip power by using the following formula:

$$\frac{\Delta E}{E} = \frac{E_{bit} \cdot \Delta traffic}{TDP \cdot t},$$

where

- ΔE is the change in chip energy consumption,
- E_{bit} is the energy cost of transferring 1 bit across on-package interconnect and equals 2 pJ for our system [60],
- $\Delta traffic$ is the amount of traffic added by Meduza,
- E is the full energy chip consumed while executing the benchmark,
- TDP is the thermal design power of the chip and equals 280 Watts for our system [24],
- t is the time to execute the benchmark.

Using these numbers and simulator measurements, we estimate that the added traffic increases chip power by up to 2%, and by 0.4% on average.

Our estimates show that Meduza provides reasonable performance/power trade-off in multi-chiplet systems **on average**. However, for some benchmarks (e.g. *x264* from the PARSEC benchmark suite) the increased traffic provides almost no additional performance. This might be a reason to consider enhancing Meduza to support adaptive behavior. In this case, Meduza could switch between write-update and write-invalidate depending on the type of workload currently running. The idea of adaptive behavior is not new, but in previous works, it was considered a way to avoid losing performance in programs where interconnect bandwidth is not enough for write-update protocol. However, we show in this paper that for modern systems, this is usually not necessary.

4.6 Discussion

The results we have presented in this section demonstrate that using underutilized interconnects is a productive way to increase performance in future systems made in a Post-Moore's law era. In particular, write-update policies are a great candidate for multi-chiplet systems.

The bandwidth sensitivity study shows that the required interconnect bandwidth is not as high as we could expect. Currently, available bandwidth in MCM systems allows for a 19% performance increase, close to the upper bound of 22% achieved in a theoretical system with 'unlimited' interconnect bandwidth.

The benefits of switching to write-update protocols will be even larger for future systems. New architectures will have increasingly higher core counts, which inevitably will lead to more disaggregation and higher intra-chiplet latencies. As we discussed in Sec. 3, interconnect bandwidth is expected to continue to grow. We have shown that all these parameters increase the performance of write-update protocols against write-invalidate. Write-update will continue to be the best solution for coherence protocols in the future.

5 RELATED WORK

5.1 Post-Moore's Law Performance Scaling

The approaching end of Moore's Law is forcing researchers to think about systematic ways to increase the performance of future computational systems.

One such approach is the extensive use of specialized accelerators for various tasks. There have been proposals for specialized architectures for machine learning [20, 38], graph processing [19, 34, 62, 63], genome sequencing [79], database processing [83], zero-knowledge proofs [88], array sorting [46, 71], and more. Many researchers advocate for open-source hardware as a way to increase the pace of innovation in the post-Moore's Law era [9–11, 22, 47, 52, 53, 72]. Another idea for future scaling is the usage of new technologies like GaAs [73], quantum computing [80], or cryogenic computing [16].

Employing high-bandwidth interconnects for performance enhancements is orthogonal to all these ideas and can be used together with accelerators, open-source hardware, or new device technologies.

5.2 Write-Update Coherence Protocols

The Dragon protocol [7] and the Firefly protocol [82] were developed and evaluated around the same time in the 1980s for machines of that time. Notably, the interconnect bandwidth of those machines is limited to only 2 bytes per clock cycle since cores were located on separate chips. More recent works on cache coherence protocols concentrate on hybrid/adaptive designs or usage in accelerators.

Meduza is based on the Dragon protocol. However, in contrast to both Firefly and Dragon, Meduza includes support for multi-level cache hierarchies in the form of the write-select policy. Moreover, this paper provides a comprehensive performance analysis of Meduza for modern and future multi-chiplet systems, which does not exist for both Firefly and Dragon.

Grahn et. al [31] propose a write-update solution for newer chips with multi-level cache subsystems. However, they consider a write-through policy as the only solution to write-update challenges in multi-level systems. In contrast, this work proposes a write-select write policy and performs a quantitative comparison to determine the optimal solution to be used in Meduza. Moreover, unlike [31], our work provides evaluation for novel multi-chiplet systems.

Much of the research that uses write-update policy focuses on adaptive/hybrid protocols, which have characteristics of both write-update and write-invalidate protocols and/or change their behavior depending on the characteristics of executed workload [2, 23, 25, 26, 30, 61, 67]. More recent works [21, 40, 41] base their design on newer ccNUMA systems.

Unlike all these protocols, Meduza is neither hybrid nor adaptive, making it easier to implement and reason about. Instead, Meduza is designed to be the best coherence protocol for systems with high-bandwidth interconnects. Write-update protocols always operate with a lower number of coherence misses; having enough throughput in the interconnect it always outperforms hybrid and adaptive approaches. As we have shown in Sec. 4, making protocol adaptive or hybrid does not make it perform better in future chips because they possess enough bandwidth in the interconnects to make write-update protocol an optimal choice in every application in terms of execution time. Moreover, these protocols are evaluated only in monolithic systems, whereas Meduza is evaluated in multi-chiplet systems.

Another recent work makes a similar argument but for using a write-update protocol in disaggregated systems [89]. The authors use write-update to enhance performance in systems with Next Generation Last Level Caches (NG-LLCs) - large caches based on DRAM technology with high capacities and high latencies. Unlike this previous work, Meduza targets multi-chiplet systems and is also evaluated in the context of multi-chiplet systems.

The authors of the VIPS [69] work propose to use a mechanism very similar to the write-select policy called Dynamic Write-policy. This mechanism also chooses between write-through and write-update policies dynamically on a per-line basis. However, VIPS used this idea to simplify the coherence protocol and eliminate read-indirection through the directory. In contrast, Meduza uses write-select in order to enable write-update cache coherence in multi-level cache hierarchies.

5.3 Comparisons of Write-Update Protocols Versus Write-Invalidate Protocols

Stenström [78] analytically compared protocols for multiple algorithms. His comparison shows that write-update protocols have fewer coherence misses but can sometimes lead to much more traffic on the interconnect.

Archibald et al. [5] used a probabilistic model and synthetic traces to compare many protocols, including Dragon, Firefly, Illinois, and other write-invalidate protocols. Simulations show that write-update protocols are better than any write-invalidate protocol regardless of workload characteristics. This result is not surprising since this work does not simulate the interconnect congestion.

Terasawa et al. [81] compared different combinations of write-update/write-invalidate, write-through/write-back, and with/without line forwarding in coherence protocols and evaluated them using an instruction-level simulator and real benchmarks. Our work uses a more realistic model with interconnect congestion, in which write-invalidate protocols outperform write-update protocols.

Rohde et al. [68] evaluated the Dragon protocol for coherence in reconfigurable accelerators. Their results show an up to an 18% performance advantage when using Dragon vs. MOESI. Unlike mentioned work, our work concentrates on designing and evaluating general-purpose multicore CPUs.

6 CONCLUSION

This paper discusses modern trends in computer architecture and the semiconductor industry and how they affect future designs. We find that advancements in inter-die, inter-chip, and off-chip interconnect bandwidth, as well as the gradual decline in transistor density due to Moore's Law ending, are slowly leading to an eventual mismatch between interconnect throughputs and the silicon's ability to utilize them. We argue that researchers should start thinking now about design optimizations that convert excess interconnect bandwidth into execution performance.

We discuss the write-update protocols as an example of such design optimizations. The write-update protocols remove the data movement between cores from a memory request's critical path by eliminating coherence misses at the cost of higher interconnect utilization.

We present Meduza: a coherence protocol for modern and future multi-chiplet systems that adapts the Dragon protocol to the modern multi-level cache hierarchies. We study the properties of this protocol and compare performance metrics in the baseline multi-chiplet system with the MESIF protocol. Our simulations show an average performance increase of +19%. Interconnect latency and bandwidth, as well as the core count of the system, are found to have a large impact on Meduza's performance. As future systems obtain better interconnects, become bigger, and even more disaggregated, Meduza and write-update can provide even more performance for these architectures.

ACKNOWLEDGMENTS

We would like to thank Ang Li, Rohan Prabhakar, August Ning, Marcelo Orenes-Vera, Alexey Lavrov, Jonathan Balkind, Georgios Tziantzioulis, Ting-Jung Chang, and anonymous reviewers for their valuable feedback. This material is based on research sponsored by the National Science Foundation under Grant No. CCF-1822949, Air Force Research Laboratory (AFRL) and Defense Advanced Research Projects Agency (DARPA) under agreement No. FA8650-18-2-7862. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of Air Force Research Laboratory (AFRL) and Defense Advanced Research Projects Agency (DARPA) or the U.S. Government. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

REFERENCES

[1] 2020. Google Cloud TPU Website. https://cloud.google.com/tpu.

[2] Craig Anderson and Anna R. Karlin. 1996. Two Adaptive Hybrid Cache Coherency Protocols. In *Proceedings of the 2nd IEEE Symposium on High-Performance Computer Architecture (HPCA '96)*. IEEE Computer Society, USA, 303.

[3] Ian Cutress Anton Shilov. 2018. GlobalFoundries Stops All 7nm Development: Opts To Focus on Specialized Processes. https://www.anandtech.com/show/13277/globalfoundries-stops-all-7nm-development.

[4] Apple. 2022. Apple unveils M1 Ultra, the world's most powerful chip for a personal computer. https://www.apple.com/newsroom/2022/03/apple-unveils-m1-ultra-the-worlds-most-powerful-chip-for-a-personal-computer.

[5] James Archibald and Jean-Loup Baer. 1986. Cache Coherence Protocols: Evaluation Using a Multiprocessor Simulation Model. *ACM Trans. Comput. Syst.* 4, 4 (Sept. 1986), 273–298. https://doi.org/10.1145/6513.6514

[6] Akhil Arunkumar, Evgeny Bolotin, Benjamin Cho, Ugljesa Milic, Eiman Ebrahimi, Oreste Villa, Aamer Jaleel, Carole-Jean Wu, and David Nellans. 2017. MCM-GPU: Multi-Chip-Module GPUs for Continued Performance Scalability. *SIGARCH Comput. Archit. News* 45, 2 (jun 2017), 320–332. https://doi.org/10.1145/3140659.3080231

[7] Russell R. Atkinson and Edward M. McCreight. 1987. The Dragon Processor. In *Proceedings of the Second International Conference on Architectual Support for Programming Languages and Operating Systems* (Palo Alto, California, USA) *(ASPLOS II)*. IEEE Computer Society Press, Washington, DC, USA, 65–69. https://doi.org/10.1145/36206.36185

[8] David H. Bailey. 2011. *NAS Parallel Benchmarks*. Springer US, Boston, MA, 1254–1259. https://doi.org/10.1007/978-0-387-09766-4_133

[9] Jonathan Balkind, Ting-Jung Chang, Paul J. Jackson, Georgios Tziantzioulis, Ang Li, Fei Gao, Alexey Lavrov, Grigory Chirkov, Jinzheng Tu, Mohammad Shahrad, and David Wentzlaff. 2020. OpenPiton at 5: A Nexus for Open and Agile Hardware Design. *IEEE Micro* 40, 4 (2020), 22–31. https://doi.org/10.1109/MM.2020.2997706

[10] Jonathan Balkind, Katie Lim, Michael Schaffner, Fei Gao, Grigory Chirkov, Ang Li, Alexey Lavrov, Tri M. Nguyen, Yaosheng Fu, Florian Zaruba, Kunal Gulati, Luca Benini, and David Wentzlaff. 2020. *BYOC: A "Bring Your Own Core" Framework for Heterogeneous-ISA Research*. Association for Computing Machinery, New York, NY, USA, 699–714. https://doi.org/10.1145/3373376.3378479

[11] Jonathan Balkind, Michael McKeown, Yaosheng Fu, Tri Nguyen, Yanqi Zhou, Alexey Lavrov, Mohammad Shahrad, Adi Fuchs, Samuel Payne, Xiaohua Liang, Matthew Matl, and David Wentzlaff. 2016. OpenPiton: An Open Source Manycore Research Framework. *SIGARCH Comput. Archit. News* 44, 2 (mar 2016), 217–232. https://doi.org/10.1145/2980024.2872414

[12] Bahareh Banijamali, Suresh Ramalingam, Kumar Nagarajan, and Raghu Chaware. 2011. Advanced reliability study of TSV interposers and interconnects for the 28nm technology FPGA. In *2011 IEEE 61st Electronic Components and Technology Conference (ECTC)*. 285–290. https://doi.org/10.1109/ECTC.2011.5898527

[13] Scott Beamer, Krste Asanović, Christopher Batten, Ajay Joshi, and Vladimir Stojanović. 2009. Designing Multi-Socket Systems Using Silicon Photonics. In *Proceedings of the 23rd International Conference on Supercomputing* (Yorktown Heights, NY, USA) *(ICS '09)*. Association for Computing Machinery, New York, NY, USA, 521–522. https://doi.org/10.1145/1542275.1542360

[14] Arijit Biswas. 2021. Sapphire Rapids. In *2021 IEEE Hot Chips 33 Symposium (HCS)*. IEEE Computer Society, Los Alamitos, CA, USA.

[15] David Blythe. 2021. Ponte Vecchio. In *2021 IEEE Hot Chips 33 Symposium (HCS)*. IEEE Computer Society, Los Alamitos, CA, USA.

[16] Ilkwon Byun, Dongmoon Min, Gyu-hyeon Lee, Seongmin Na, and Jangwoo Kim. 2020. CryoCore: A Fast and Dense Processor Architecture for Cryogenic Computing. In *Proceedings of the ACM/IEEE 47th Annual International Symposium on Computer Architecture* (Virtual Event) *(ISCA '20)*. IEEE Press, 335–348. https://doi.org/10.1109/ISCA45697.2020.00037

[17] Trevor E. Carlson, Wim Heirman, and Lieven Eeckhout. 2011. Sniper: Exploring the level of abstraction for scalable and accurate parallel multi-core simulation. In *SC '11: Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis*. 1–12. https://doi.org/10.1145/2063384.2063454

[18] William Chen and Bill Bottoms. 2019. Heterogeneous Integration Roadmap: Driving Force and Enabling Technology for Systems of the Future. In *2019 Symposium on VLSI Technology*. T50–T51. https://doi.org/10.23919/VLSIT.2019.8776484

[19] Xuhao Chen, Tianhao Huang, Shuotao Xu, Thomas Bourgeat, Chanwoo Chung, and Arvind Arvind. 2021. FlexMiner: A Pattern-Aware Accelerator for Graph Pattern Mining. In *2021 ACM/IEEE 48th Annual International Symposium on Computer Architecture (ISCA)*. 581–594. https://doi.org/10.1109/ISCA52012.2021.00052

[20] Yu-Hsin Chen, Tushar Krishna, Joel S. Emer, and Vivienne Sze. 2017. Eyeriss: An Energy-Efficient Reconfigurable Accelerator for Deep Convolutional Neural Networks. *IEEE Journal of Solid-State Circuits* 52, 1 (2017), 127–138. https://doi.org/10.1109/JSSC.2016.2616357

[21] Liqun Cheng and John B. Carter. 2008. Extending CC-NUMA systems to support write update optimizations. In *SC '08: Proceedings of the 2008 ACM/IEEE Conference on Supercomputing*. 1–12. https://doi.org/10.1109/SC.2008.5215354

[22] Grigory Chirkov and David Wentzlaff. 2023. SMAPPIC: Scalable Multi-FPGA Architecture Prototype Platform in the Cloud. In *Proceedings of the 28th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2* (Vancouver, BC, Canada) *(ASPLOS 2023)*. Association for Computing Machinery, New York, NY, USA, 733–746. https://doi.org/10.1145/3575693.3575753

[23] Alan L. Cox and Robert J. Fowler. 1993. Adaptive Cache Coherency for Detecting Migratory Shared Data. *SIGARCH Comput. Archit. News* 21, 2 (may 1993), 98–108. https://doi.org/10.1145/173682.165146

[24] Ian Cutress and Andrei Frumusanu. 2021. AMD 3rd Gen EPYC Milan Review: A Peak vs Per Core Performance Balance. https://www.anandtech.com/show/16529/amd-epyc-milan-review.

[25] F. Dahlgren. 1995. Boosting the performance of hybrid snooping cache protocols. In *Proceedings 22nd Annual International Symposium on Computer Architecture*. 60–69.

[26] Fredrik Dahlgren and Per Stenstrom. 1994. Reducing the Write Traffic for a Hybrid Cache Protocol. In *1994 International Conference on Parallel Processing Vol. 1*, Vol. 1. 166–173. https://doi.org/10.1109/ICPP.1994.175

[27] Yigit Demir, Yan Pan, Seukwoo Song, Nikos Hardavellas, John Kim, and Gokhan Memik. 2014. Galaxy: A High-Performance Energy-Efficient Multi-Chip Architecture Using Photonic Interconnects. In *Proceedings of the 28th ACM International Conference on Supercomputing* (Munich, Germany) *(ICS '14)*. Association for Computing Machinery, New York, NY, USA, 303–312. https://doi.org/10.1145/2597652.2597664

[28] Pouya Fotouhi, Sebastian Werner, Jason Lowe-Power, and S. J. Ben Yoo. 2019. Enabling Scalable Chiplet-Based Uniform Memory Architectures with Silicon Photonics. In *Proceedings of the International Symposium on Memory Systems* (Washington, District of Columbia, USA) *(MEMSYS '19)*. Association for Computing Machinery, New York, NY, USA, 222–334. https://doi.org/10.1145/3357526.3357564

[29] Yaosheng Fu and David Wentzlaff. 2014. PriME: A parallel and distributed simulator for thousand-core chips. In *2014 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)*. 116–125. https://doi.org/10.1109/ISPASS.2014.6844467

[30] Håkan Grahn and Per Stenström. 1996. Evaluation of a Competitive-Update Cache Coherence Protocol with Migratory Data Detection. *J. Parallel Distrib. Comput.* 39, 2 (Dec. 1996), 168–180. https://doi.org/10.1006/jpdc.1996.0164

[31] Håkan Grahn, Per Stenström, and Michel Dubois. 1995. Implementation and Evaluation of Update-Based Cache Protocols under Relaxed Memory Consistency Models. *Future Gener. Comput. Syst.* 11, 3 (jun 1995), 247–271. https://doi.org/10.1016/0167-739X(94)00067-O

[32] David Greenhill, Ron Ho, David Lewis, Herman Schmit, Kok Hong Chan, Andy Tong, Sean Atsatt, Dana How, Peter McElheny, Keith Duwel, Jeffrey Schulz, Darren Faulkner, Gopal Iyer, George Chen, Hee Kong Phoon, Han Wooi Lim, Wei-Yee Koay, and Ty Garibay. 2017. 3.3 A 14nm 1GHz FPGA with 2.5D transceiver integration. In *2017 IEEE International Solid-State Circuits Conference (ISSCC)*. 54–55. https://doi.org/10.1109/ISSCC.2017.7870257

[33] Ted Greenwald. 2018. Intel's Chip Stumble Is Letting Rivals Pull Ahead. https://www.wsj.com/articles/intels-chip-stumble-is-letting-rivals-pull-ahead-1529845200.

[34] Tae Jun Ham, Lisa Wu, Narayanan Sundaram, Nadathur Satish, and Margaret Martonosi. 2016. Graphicionado: A high-performance and energy-efficient accelerator for graph analytics. In *2016 49th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*. 1–13. https://doi.org/10.1109/MICRO.2016.7783759

[35] Oved Itzhak, Idit Keidar, Avinoam Kolodny, and Uri C Weiser. 2014. Performance scalability and dynamic behavior of Parsec benchmarks on many-core processors. In *The 4th Workshop on Systems for Future Multicore Architectures, http://sfma14.cs. washington. edu/(accessed June 15, 2015)*. Citeseer.

[36] Christian Jacobi. 2021. Real-time AI for Enterprise Workloads: the IBM Telum Processor. In *2021 IEEE Hot Chips 33 Symposium (HCS)*. IEEE Computer Society, Los Alamitos, CA, USA.

[37] JEDEC. 2023. JEDEC Standards. https://www.jedec.org/category/technology-focus-area/main-memory-ddr3-ddr4-sdram.

[38] Norman Jouppi, Cliff Young, Nishant Patil, and David Patterson. 2018. Motivation for and Evaluation of the First Tensor Processing Unit. *IEEE Micro* 38, 3 (2018), 10–19. https://doi.org/10.1109/MM.2018.032271057

[39] Ajaykumar Kannan, Natalie Enright Jerger, and Gabriel H. Loh. 2015. Enabling interposer-based disintegration of multi-core processors. In *2015 48th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*. 546–558. https://doi.org/10.1145/2830772.2830808

[40] Abdullah Kayi, Olivier Serres, and Tarek El-Ghazawi. 2012. Bandwidth Adaptive Write-update Optimizations for Chip Multiprocessors. In *2012 IEEE 10th International Symposium on Parallel and Distributed Processing with Applications*. 199–206. https://doi.org/10.1109/ISPA.2012.34

[41] Abdullah Kayi, Olivier Serres, and Tarek El-Ghazawi. 2015. Adaptive Cache Coherence Mechanisms with Producer–Consumer Sharing Optimization for Chip Multiprocessors. *IEEE Trans. Comput.* 64, 2 (2015), 316–328. https://doi.org/10.1109/TC.2013.217

[42] David Kehlet et al. 2017. Accelerating innovation through a standard chiplet interface: The advanced interface bus (AIB). *Intel White Paper* (2017).

[43] Hassan Khan, David Hounshell, and Erica Fuchs. 2018. Science and research policy at the end of Moore's law. *Nature Electronics* 1 (01 2018). https://doi.org/10.1038/s41928-017-0005-9

[44] Jinwoo Kim, Venkata Chaitanya Krishna Chekuri, Nael Mizanur Rahman, Majid Ahadi Dolatsara, Hakki Torun, Madhavan Swaminathan, Saibal Mukhopadhyay, and Sung Kyu Lim. 2020. Silicon vs. Organic Interposer: PPA and Reliability Tradeoffs in Heterogeneous 2.5D Chiplet Integration. In *2020 IEEE 38th International Conference on Computer Design (ICCD)*. 80–87. https://doi.org/10.1109/ICCD50377.2020.00030

[45] Kevin Krewell. 2019. NVIDIA Is A Data Center Company Now. https://www.forbes.com/sites/tiriasresearch/2019/03/29/nvidia-is-a-data-center-company-now.

[46] Ang Li, August Ning, and David Wentzlaff. 2023. Duet: Creating Harmony between Processors and Embedded FPGAs. In *2023 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*. 745–758. https://doi.org/10.1109/HPCA56546.2023.10070989

[47] Ang Li and David Wentzlaff. 2021. PRGA: An Open-Source FPGA Research and Prototyping Framework. In *The 2021 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays* (Virtual Event, USA) *(FPGA '21)*. Association for Computing Machinery, New York, NY, USA, 127–137. https://doi.org/10.1145/3431920.3439294

[48] M. Lin, T. Huang, C. Tsai, K. Tam, C. Hsieh, T. Chen, W. Huang, J. Hu, Y. Chen, S. K. Goel, C. Fu, S. Rusu, C. Li, S. Yang, M. Wong, S. Yang, and F. Lee. 2019. A 7nm 4GHz Arm®-core-based CoWoS® Chiplet Design for High Performance Computing. In *2019 Symposium on VLSI Circuits*. C28–C29. https://doi.org/10.23919/VLSIC.2019.8778161

[49] Mu-Shan Lin, Chien-Chun Tsai, Cheng-Hsiang Hsieh, Wen-Hung Huang, Yu-Chi Chen, Shu-Chun Yang, Chin-Ming Fu, Hao-Jie Zhan, Jinn-Yeh Chien, Shao-Yu Li, Y.-H. Chen, C.-C. Kuo, Shih-Peng Tai, and Kazuyoshi Yamada. 2016. A 16nm 256-bit wide 89.6GByte/s total bandwidth in-package interconnect with 0.3V swing and 0.062pJ/bit power in InFO package. In *2016 IEEE Hot Chips 28 Symposium (HCS)*. 1–32. https://doi.org/10.1109/HOTCHIPS.2016.7936211

[50] Gabriel H. Loh, Natalie Enright Jerger, Ajaykumar Kannan, and Yasuko Eckert. 2015. Interconnect-Memory Challenges for Multi-Chip, Silicon Interposer Systems. In *Proceedings of the 2015 International Symposium on Memory Systems* (Washington DC, DC, USA) *(MEMSYS '15)*. Association for Computing Machinery, New York, NY, USA, 3–10. https://doi.org/10.1145/2818950.2818951

[51] R. Mahajan, R. Sankman, N. Patel, D. Kim, K. Aygun, Z. Qian, Y. Mekonnen, I. Salama, S. Sharan, D. Iyengar, and D. Mallik. 2016. Embedded Multi-die Interconnect Bridge (EMIB) – A High Density, High Bandwidth Packaging Interconnect. In *2016 IEEE 66th Electronic Components and Technology Conference (ECTC)*. 557–565. https://doi.org/10.1109/ECTC.2016.201

[52] Paolo Mantovani, Davide Giri, Giuseppe Di Guglielmo, Luca Piccolboni, Joseph Zuckerman, Emilio G. Cota, Michele Petracca, Christian Pilato, and Luca P. Carloni. 2020. Agile SoC Development with Open ESP. In *Proceedings of the 39th International Conference on Computer-Aided Design* (Virtual Event, USA) *(ICCAD '20)*. Association for Computing Machinery, New York, NY, USA, Article 96, 9 pages. https://doi.org/10.1145/3400302.3415753

[53] Paolo Mantovani, Davide Giri, Giuseppe Di Guglielmo, Luca Piccolboni, Joseph Zuckerman, Emilio G. Cota, Michele Petracca, Christian Pilato, and Luca P. Carloni. 2020. Agile SoC Development with Open ESP. In *Proceedings of the 39th International Conference on Computer-Aided Design* (Virtual Event, USA) *(ICCAD '20)*. Association for Computing Machinery, New York, NY, USA, Article 96, 9 pages. https://doi.org/10.1145/3400302.3415753

[54] Jason E. Miller, Harshad Kasture, George Kurian, Charles Gruenwald, Nathan Beckmann, Christopher Celio, Jonathan Eastep, and Anant Agarwal. 2010. Graphite: A distributed parallel simulator for multicores. In *HPCA - 16 2010 The Sixteenth International Symposium on High-Performance Computer Architecture*. 1–12. https://doi.org/10.1109/HPCA.2010.5416635

[55] Hassan Mujtaba. 2020. Intel 10nm Sapphire Rapids Xeon Scalable Family With DDR5 and PCIe 5.0 Coming 2021, Will Compete Against AMD's EPYC Genoa 'Zen 4' Server Chips. https://wccftech.com/intel-10nm-sapphire-rapids-xeon-scalable-family-amd-epyc-genoa-5nm-2021-launch/.

[56] Hassan Mujtaba. 2020. Intel Unveils 3rd Gen Ice Lake-SP Xeon CPU Family: 10nm+ Sunny Cove Cores, New Instructions, 28 Core Chip Showcased. https://wccftech.com/intel-unveils-ice-lake-sp-xeon-cpu-family-10nm-sunny-cove-cores-28-core-die.

[57] Hassan Mujtaba. 2021. AMD EPYC Genoa CPU Platform Detailed – Up To 96 Zen 4 Cores, 192 Threads, 12-Channel DDR5-5200, 128 PCIe Gen 5 Lanes, SP5 'LGA 6096' Socket. https://wccftech.com/amd-epyc-genoa-cpu-platform-detailed-up-to-96-zen-4-cores-12-channel-ddr5-5200-sp5-lga-6096-socket.

[58] Richard C Murphy, Kyle B Wheeler, Brian W Barrett, and James A Ang. 2010. Introducing the graph 500. *Cray Users Group (CUG)* 19 (2010), 45–74.

[59] Samuel Naffziger, Noah Beck, Thomas Burd, Kevin Lepak, Gabriel H. Loh, Mahesh Subramony, and Sean White. 2021. Pioneering Chiplet Technology and Design for the AMD EPYC™ and Ryzen™ Processor Families : Industrial Product. In *2021 ACM/IEEE 48th Annual International Symposium on Computer Architecture (ISCA)*. 57–70. https://doi.org/10.1109/ISCA52012.2021.00014

[60] S. Naffziger, K. Lepak, M. Paraschou, and M. Subramony. 2020. 2.2 AMD Chiplet Architecture for High-Performance Server and Desktop Products. In *2020 IEEE International Solid- State Circuits Conference - (ISSCC)*. 44–45. https://doi.org/10.1109/ISSCC19947.2020.9063103

[61] Håkan Nilsson and Per Stenström. 1994. An Adaptive Update-Based Cache Coherence Protocol for Reduction of Miss Rate and Traffic. In *Proceedings of the 6th International PARLE Conference on Parallel Architectures and Languages Europe (PARLE '94)*. Springer-Verlag, Berlin, Heidelberg, 363–374.

[62] Marcelo Orenes-Vera, Aninda Manocha, Jonathan Balkind, Fei Gao, Juan L. Aragón, David Wentzlaff, and Margaret Martonosi. 2022. Tiny but Mighty: Designing and Realizing Scalable Latency Tolerance for Manycore SoCs. In *Proceedings of the 49th Annual International Symposium on Computer Architecture* (New York, New York) *(ISCA '22)*. Association for Computing Machinery, New York, NY, USA, 817–830. https://doi.org/10.1145/3470496.3527400

[63] Marcelo Orenes-Vera, Esin Tureci, David Wentzlaff, and Margaret Martonosi. 2023. Dalorex: A Data-Local Program Execution and Architecture for Memory-bound Applications. In *2023 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*. 718–730. https://doi.org/10.1109/HPCA56546.2023.10071089

[64] Mark S. Papamarcos and Janak H. Patel. 1984. A Low-Overhead Coherence Solution for Multiprocessors with Private Cache Memories. In *Proceedings of the 11th Annual International Symposium on Computer Architecture (ISCA '84)*. Association for Computing Machinery, New York, NY, USA, 348–354. https://doi.org/10.1145/800015.808204

[65] John W. Poulton, William J. Dally, Xi Chen, John G. Eyles, Thomas H. Greer, Stephen G. Tell, John M. Wilson, and C. Thomas Gray. 2013. A 0.54 pJ/b 20 Gb/s Ground-Referenced Single-Ended Short-Reach Serial Link in 28 nm CMOS for Advanced Packaging Applications. *IEEE Journal of Solid-State Circuits* 48, 12 (2013), 3206–3218. https://doi.org/10.1109/JSSC.2013.2279053

[66] Sandeep Sane Ravi Mahajan. 2021. Advanced Packaging Technologies for Heterogeneous Integration (HI). In *2021 IEEE Hot Chips 33 Symposium (HCS)*. IEEE Computer Society, Los Alamitos, CA, USA.

[67] A. Raynaud, Zheng Zhang, and J. Torrellas. 1996. Distance-adaptive update protocols for scalable shared-memory multiprocessors. In *Proceedings. Second International Symposium on High-Performance Computer Architecture*. 323–334. https://doi.org/10.1109/HPCA.1996.501197

[68] Johanna Rohde, Lukas Johannes Jung, and Christian Hochberger. 2019. Update or Invalidate: Influence of Coherence Protocols on Configurable HW Accelerators. In *Applied Reconfigurable Computing*, Christian Hochberger, Brent Nelson, Andreas Koch, Roger Woods, and Pedro Diniz (Eds.). Springer International Publishing, Cham, 305–316. https://doi.org/10.1007/978-3-030-17227-5_22

[69] Alberto Ros, Mahdad Davari, and Stefanos Kaxiras. 2015. Hierarchical private/shared classification: The key to simple and efficient coherence for clustered cache hierarchies. In *2015 IEEE 21st International Symposium on High Performance Computer Architecture (HPCA)*. 186–197. https://doi.org/10.1109/HPCA.2015.7056032

[70] Ananda Samajdar, Tushar Garg, Tushar Krishna, and Nachiket Kapre. 2019. Scaling the Cascades: Interconnect-Aware FPGA Implementation of Machine Learning Problems. In *2019 29th International Conference on Field Programmable Logic and Applications (FPL)*. 342–349. https://doi.org/10.1109/FPL.2019.00061

[71] Nikola Samardzic, Weikang Qiao, Vaibhav Aggarwal, Mau-Chung Frank Chang, and Jason Cong. 2020. Bonsai: High-Performance Adaptive Merge Tree Sorting. In *2020 ACM/IEEE 47th Annual International Symposium on Computer Architecture (ISCA)*. 282–294. https://doi.org/10.1109/ISCA45697.2020.00033

[72] Maico Cassel dos Santos, Tianyu Jia, Martin Cochet, Karthik Swaminathan, Joseph Zuckerman, Paolo Mantovani, Davide Giri, Jeff Jun Zhang, Erik Jens Loscalzo, Gabriele Tombesi, Kevin Tien, Nandhini Chandramoorthy, John-David Wellman, David Brooks, Gu-Yeon Wei, Kenneth Shepard, Luca P. Carloni, and Pradip Bose. 2022. A Scalable Methodology for Agile Chip Development with Open-Source Hardware Components. In *Proceedings of the 41st IEEE/ACM International Conference on Computer-Aided Design* (San Diego, California) *(ICCAD '22)*. Association for Computing Machinery, New York, NY, USA, Article 20, 9 pages. https://doi.org/10.1145/3508352.3561102

[73] R. Sarmiento, F. Tobajas, V. de Armas, R. Esper-Chain, J.F. Lopez, J.A. Montiel-Nelson, and A. Nunez. 1998. A CORDIC processor for FFT computation and its implementation using gallium arsenide technology. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 6, 1 (1998), 18–30. https://doi.org/10.1109/92.661241

[74] C. L. Schow, F. E. Doany, B. G. Lee, R. Budd, C. Baks, R. Dangel, R. A. John, F. Libsch, J. A. Kash, B. Chan, H. Lin, C. Carver, J. Huang, J. Berry, and D. Bajkowski. 2011. 225 Gb/s Bi-Directional Integrated Optical PCB Link, In Optical Fiber Communication Conference/National Fiber Optic Engineers Conference 2011. *Optical Fiber Communication Conference/National Fiber Optic Engineers Conference 2011*, PDPA2. https://doi.org/10.1364/NFOEC.2011.PDPA2

[75] Debendra Das Sharma. 2020. What's the Difference Going from PCIe 3.0 to PCIe 6.0? https://www.electronicdesign.com/industrial-automation/article/21136215/

whats-the-difference-going-from-pcie-30-to-pcie-60.

[76] Ryan Smith. 2021. AMD Announces Instinct MI200 Accelerator Family: Taking Servers to Exascale and Beyond. https://www.anandtech.com/show/17054/amd-announces-instinct-mi200-accelerator-family-cdna2-exacale-servers.

[77] Gabriel Southern and Jose Renau. 2015. Deconstructing PARSEC scalability. In *Proc. of the Annual Workshop on Duplicating, Deconstructing, and Debunking (WDDD)*.

[78] Per Stenström. 1990. A Survey of Cache Coherence Schemes for Multiprocessors. *Computer* 23, 6 (June 1990), 12–24. https://doi.org/10.1109/2.55497

[79] Arun Subramaniyan, Jack Wadden, Kush Goliya, Nathan Ozog, Xiao Wu, Satish Narayanasamy, David Blaauw, and Reetuparna Das. 2021. Accelerated Seeding for Genome Sequence Alignment with Enumerated Radix Trees. In *2021 ACM/IEEE 48th Annual International Symposium on Computer Architecture (ISCA)*. 388–401. https://doi.org/10.1109/ISCA52012.2021.00038

[80] Wei Tang, Teague Tomesh, Martin Suchara, Jeffrey Larson, and Margaret Martonosi. 2021. *CutQC: Using Small Quantum Computers for Large Quantum Circuit Evaluations*. Association for Computing Machinery, New York, NY, USA, 473–486. https://doi.org/10.1145/3445814.3446758

[81] Takuya Terasawa, Keisuke Inoue, Hitoshi Kurosawa, and Hideharu Amano. 1997. A study on snoop cache systems for single-chip multiprocessors. *Systems and Computers in Japan* 28, 2 (1997), 62–72. https://doi.org/10.1002/(SICI)1520-684X(199702)28:2<62::AID-SCJ7>3.0.CO;2-P

[82] Charles P. Thacker and Lawrence C. Stewart. 1987. Firefly: A Multiprocessor Workstation. *SIGARCH Comput. Archit. News* 15, 5 (Oct. 1987), 164–172. https://doi.org/10.1145/36177.36199

[83] Matthew Vilim, Alexander Rucker, and Kunle Olukotun. 2021. Aurochs: An Architecture for Dataflow Threads. In *2021 ACM/IEEE 48th Annual International Symposium on Computer Architecture (ISCA)*. 402–415. https://doi.org/10.1109/ISCA52012.2021.00039

[84] S. W. Yoon, D. W. Yang, J. H. Koo, M. Padmanathan, and F. Carson. 2009. 3D TSV processes and its assembly/packaging technology. In *2009 IEEE International Conference on 3D System Integration*. 1–5. https://doi.org/10.1109/3DIC.2009.5306535

[85] Douglas Yu. 2021. TSMC Packaging Technologies for Chiplets and 3D. In *2021 IEEE Hot Chips 33 Symposium (HCS)*. IEEE Computer Society, Los Alamitos, CA, USA.

[86] Meghan Zea. 2019. PCI-SIG® Achieves 32GT/s with New PCI Express® 5.0 Specification. https://www.businesswire.com/news/home/20190529005766/en/PCI-SIG%C2%AE-Achieves-32GTs-New-PCI-Express%C2%AE-5.0.

[87] Xusheng Zhan, Yungang Bao, Christian Bienia, and Kai Li. 2017. PARSEC3.0: A Multicore Benchmark Suite with Network Stacks and SPLASH-2X. *SIGARCH Comput. Archit. News* 44, 5 (Feb. 2017), 1–16. https://doi.org/10.1145/3053277.3053279

[88] Ye Zhang, Shuo Wang, Xian Zhang, Jiangbin Dong, Xingzhong Mao, Fan Long, Cong Wang, Dong Zhou, Mingyu Gao, and Guangyu Sun. 2021. PipeZK: Accelerating Zero-Knowledge Proof with a Pipelined Architecture. In *2021 ACM/IEEE 48th Annual International Symposium on Computer Architecture (ISCA)*. 416–428. https://doi.org/10.1109/ISCA52012.2021.00040

[89] Mingcan Zhu, Amna Shahab, Antonios Katsarakis, and Boris Grot. 2021. Invalidate or Update? Revisiting Coherence for Tomorrow's Cache Hierarchies. In *2021 30th International Conference on Parallel Architectures and Compilation Techniques (PACT)*. 226–241. https://doi.org/10.1109/PACT52795.2021.00024