LLM: Realizing Low-Latency Memory by Exploiting Embedded Silicon Photonics for Irregular Workloads*

Marjan Fariborz[1], Mahyar Samani[1], Pouya Fotouhi[1], Roberto Proietti[1], Il-Min Yi[2], Venkatesh Akella[1], Jason Lowe-Power[1], Samuel Palermo[2], and S. J. Ben Yoo[1]

University of California Davis, Davis, CA 95616, USA
{mfariborz, msamani, pfotouhi, rproietti, akella, jlowepower, sbyoo}@ucdavis.edu
Texas AM University, College Station, TX, 77843, USA
{ilmin.yi, spalermo}@tamu.edu

Abstract. As emerging workloads exhibit irregular memory access patterns with poor data reuse and locality, they would benefit from a DRAM that achieves low latency without sacrificing bandwidth and energy efficiency. We propose LLM (Low Latency Memory), a codesign of the DRAM microarchitecture, the memory controller and the LLC/DRAM interconnect by leveraging embedded silicon photonics in 2.5D/3D integrated system on chip. LLM relies on Wavelength Division Multiplexing (WDM)-based photonic interconnects to reduce the contention throughout the memory subsystem. LLM also increases the bank-level parallelism, eliminates bus conflicts by using dedicated optical data paths, and reduces the access energy per bit with shorter global bitlines and smaller row buffers. We evaluate the design space of LLM for a variety of synthetic benchmarks and representative graph workloads on a full-system simulator (gem5). LLM exhibits low memory access latency for traffics with both regular and irregular access patterns. For irregular traffic, LLM achieves high bandwidth utilization (over 80% peak throughput compared to 20% of HBM2.0). For real workloads, LLM achieves 3× and 1.8× lower execution time compared to HBM2.0 and a state-of-the-art memory system with high memory level parallelism, respectively. This study also demonstrates that by reducing queuing on the data path, LLM can achieve on average 3.4× lower memory latency variation compared to HBM2.0.

1 Introduction

Emerging applications, such as recommendation systems, mining large sparse graphs, etc., exhibit irregular memory access patterns with little data reuse and poor locality [17]. For these irregular workloads, the memory subsystem is increasingly becoming the bottleneck in modern computing architectures. The memory subsystem should not only provide high bandwidth but also low latency to achieve high performance for irregular applications [14, 9]. In addition,

* This work was supported in part by ARO award W911NF1910470.

variability in memory latency is another concern as it limits the performance of computing systems [9] and increases the burden on the programmer. It is desirable that both the average memory access latency and its variability (e.g., as measured by the 95^{th} percentile) are low.

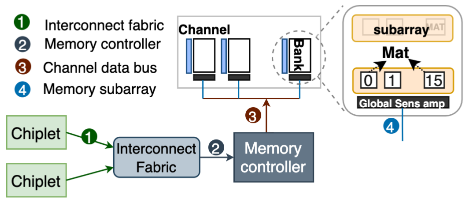
To address these challenges, there has been a resurgence of interest in DRAM microarchitectures and memory system designs. With the emergence of silicon photonics technologies, and chiplet-based architectures with 2.5D/3D packaging, there are new opportunities to co-design the various components of the memory subsystem. Recent advances in DRAM architecture, such as wider I/O enabled by 2.5D/3D packaging (as in HBM and its derivatives [30, 20]), higher data rates with serial links, and increased bank-level parallelism (again with HBM like technologies), have improved DRAM bandwidth significantly. However, often these bandwidth improvements come at the expense of additional latency and variability due to deeper queues in the memory controller to take advantage of the bank-level parallelism and serialization/deserialization (SerDes) latency [10]. There are also proposals [8, 19, 21, 22] in literature that explicitly address the latency question in DRAM microarchitectures, and most of these proposals simply take advantage of *locality* to reduce latency.

We argue that the main source of latency for irregular workloads in the memory subsystem is **contention** caused by *sharing resources* such as buffers, ports, data/command/control buses, and the DRAM cells where the data actually resides. Increasing these resources comes at a significant cost and may have physical limits such as the number of pins (I/O pads) that can be placed in a given space. Thus, we must consider sources of contention in the *entire end-to-end path*, which includes the processor/memory interconnect, memory controller, and DRAM microarchitecture. In the past, end-to-end optimization of the memory subsystem was not feasible in commodity CPUs (though there has been a slow transition in this direction with integrated memory controllers and special-purpose processors with GDDR SDRAM). However, chiplet-based architectures such as AMD's EPYC and recently announced Intel's Sapphire Rapids offer the opportunity to **co-design** the off-chip(let) processor/memory interconnect, memory controller, and the DRAM microarchitecture [4].

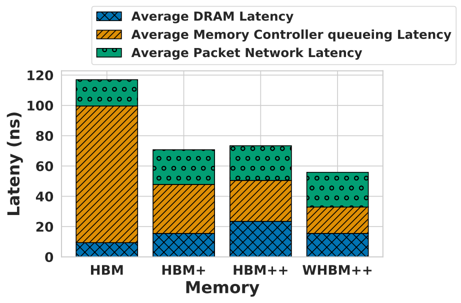
This paper describes our co-design approach, which we call Low Latency Memory (LLM). LLM simultaneously optimizes latency, bandwidth, and energy efficiency by taking advantage of silicon photonics (SiPh) interconnects with optical parallelism and wavelength routing to reduce contention in the entire path from chiplet to the DRAM subarrays. This co-optimization is now possible because silicon photonics offers lower energy/bit [35], high bandwidth density ($Gb/s/mm^2$) with wavelength division multiplexing (WDM) [29], and all-to-all interconnectivity with chip-scale AWGRs (Arrayed Waveguide Grating Routers) [36].

2 Motivation

The primary source of performance degradation for irregular applications is contention among shared resources [14]. Figure 1a shows the high-level schematic of a generic chiplet-based architecture such as AMD EPYC [4]. There are four major components in this system: the interconnect fabric between each chiplet



(a)



(b)

Fig. 1: (a) Generic High-level Architecture of the Memory Subsystem. (b) Breakdown of end-to-end latency. HBM+ increase the pseudo-channels, HBM++ reduces the size of each bank, and WHBM++ increases the data bus width compared to HBM++.

Table 1: DRAM configuration

Category	HBM	HBM+	HBM++	WHBM++
Channels/stack	8	8	8	8
pseudo-channel/channel	2	4	16	8
Banks/channel	16	16	32	32
Pins/pseudo-channel	64	32	8	64
t_{BURST}	4	8	32	4

and the memory controllers, usually a complex crossbar-like structure with high bisection bandwidth; the memory controller, which consists of queues to buffer read/write requests bound for the particular memory channel; and finally the DRAM device, which consists of multiple banks, with each bank itself made up of subarray of cells. It is important to note that the interconnect fabric, the queues inside the memory controllers, data buses within the channel, global sense amplifiers, and global bitlines within the DRAM devices are *shared*, which introduces the potential for contention and additional latency due to arbitration, buffering, and serialization (time multiplexed sharing).

Figure 1b, shows the simulation results of end-to-end latency by adding parallelism only at the DRAM microarchitecture. Here we used eight random traffic generators connected to 4-Hi stack HBM2.0 (eight channels) in gem5 [26]. We used HBM as a baseline model of HBM2.0 working in the pseudo-channel mode, which divides each HBM2.0 channel into two pseudo-channels that share the channel's address/control (ADD/CMD) bus but have their own 64-bit wide I/O interface. Table 1 shows the specification of different memories. WHBM++ has an 8 $\times$ number of pins compared to HBM++ while providing the same number of banks and pseudo-channels as HBM++.

We divided the end-to-end latency into three categories: network latency, the queuing latency at the memory controller, and DRAM access latency. Figure 1b shows that for HBM, most of the latency is in the queuing at the memory controller. When we increase resources without considering co-design, the memory controller bottleneck is alleviated. Still, the other components (the device and the network latency) begin to dominate the total latency, and there are diminishing returns. Thus, a high-performance memory, not only needs higher parallelism to reduce the memory controller queuing latency, but it must also reduce the device and interconnect latency. In fact, we propose to *re-architect the entire end-to-end system* to reduce the latency of the memory subsystem, specially as we scale the system to large number of compute units and run irregular workloads with poor data reuse and locality.

LLM makes the following ***contributions*** towards removing these sources of contention: **(a)** It proposes a ground up co-design of the entire path from the processor/memory interconnect to the DRAM microarchitecture. This co-design enables both bandwidth and latency improvement without sacrificing one for the other. LLM is composed of three pieces: a contention-less optical data plane, a low-bandwidth electrical control plane, and fine-grained memory banks with integrated photonics. **(b)** In the data plane (Figure 2a), LLM provides a dedicated data path from every requestor to every memory bank. An LLM-like architecture is impractical with electrical interconnects because of the energy costs of data movement and the wiring complexity of providing these dedicated data paths. We propose using a passive and contention-less optical interconnect for the data plane with no intermediate buffering, thus reducing the queuing and the interconnect latency compared to other chiplet-based architectures. **(c)** The control plane (shown in Figure 2b) communicates the address and command between chiplets and memory and coordinates the time that a chiplet sends or receives its data. A low bandwidth electrical network is used for carrying this control information. **(d)** LLM uses fine-grained memory units called μbanks that are exposed to the memory controller to exploit massive amounts of parallelism. LLM memory devices have integrated optics to allow low-latency high-bandwidth direct connections from the requestors to the memory μbanks.

3 Silicon photonic enabling technologies

Over the past decade, optical interconnects have shown great potentials in overcoming the bandwidth bottlenecks that limit inter-processor and memory performance [15, 44, 5]. Commercial products (e.g., Ayar Labs in collaboration with Intel) leveraging foundry-enabled (e.g. GlobalFoundries offers SiPh-CMOS fabrication) SiPh fabrics and WDM SiPh transceivers have been announced, making SiPh technology feasible for chiplet-based communications [1].

The first SiPh device we use in this study is a microring resonator. Microrings are compact and energy efficient, WDM-compatible devices that are designed to resonate when presented with specific individual wavelengths and remain quiescent at all other times. Active microrings are designed to tune their resonance frequency as the amount of current in their base layer changes, enabling data modulation and demodulation. Microring modulators encode bits onto the optical medium (electrical-to-optical (EO) conversion), and microring filters extract the optical signal and send it to a photodetector performing optical-to-electrical (OE) conversion.

Earlier proposals used optical buses and large matrices of microrings (consisting of hundreds of microrings) for the memory-to-processor network [5, 12, 23].

In this proposal, we use AWGR [16, 36, 38, 33] which is a passive silicon photonic fabric with a compact layout that offers scalable all-to-all connectivity through wavelength routing. Recent advances in the fabrication process of AWGRs now enable their integration with a significantly reduced footprint (1 mm^2), crosstalk ($<$ -38dB), and loss ($<$ 2dB) [36]. This makes the AWGR a favorable candidate for energy-efficient, high bandwidth, all-to-all connectivity within HPC systems. Initial studies have shown AWGR to be promising choice for processor-to-memory network [16, 15]. Figure 2d shows the wavelength routing in a 5×5 AWGR; all wavelengths inside a waveguide entering one input port of AWGR are evenly distributed over all the output ports, each to a unique output port.

A Vertical Optical Interconnect (VOI) is an optical waveguide that can potentially replace through-silicon vias (TSVs) in 3D stacked memories. Unlike previously demonstrated optical TSVs [32], VOIs have 1-2 μm pitch size [48] and they can provide higher bandwidth density compared to state-of-the-art TSVs (20 μm pitch size [31]).

4 Architecture

In this section we present the detailed design and implementation of LLM that harnesses the benefits of silicon photonics to reduce contention in the entire memory subsystem from the requestor (chiplet or group of chiplets) to the fine grain access units called μbanks inside the DRAM.

4.1 Processor-Memory Interconnect

LLM reduces contention by taking advantage of the lower energy consumption and the higher bandwidth density of optical interconnects for data communication. In addition, it uses a low bandwidth all-to-all electrical interconnect to manage bank conflicts and orchestrate the data movement.

Figure 2a shows the *optical data plane* with an AWGR provideing an all-to-all connection. On the memory-side, each channel is connected to a port of the AWGR using a waveguide. Each waveguide carries a wavelength for each μbank. Inside the memory channel, μbanks modulate/demodulate data on the waveguide through a tuned microring which is tuned to a specific wavelength. To enable simultaneous reads/writes per channel we can assign two waveguides per channel to connect to two separate AWGRs (one for carrying read and another for write data).

While the AWGR can route the optical signal to the destination μbank, the requestors should modulate the data on the intended wavelength and send it to the correct AWGR port. Thus, each chiplet uses an array of tunable microrings where each microring in the array directly connects to a different input ports of the AWGR to send/receive the data. For an $n \times n$ AWGR, each chiplet requires n microrings.

The request's μbank address indicates the wavelength, and its channel address indicates which microring on which waveguide needs to be tuned to the corresponding wavelength. This configuration allows (a) single requestor to send requests to every bank within a single channel using a different wavelength on each of the waveguides connected to different input ports of the AWGR; (b) at a

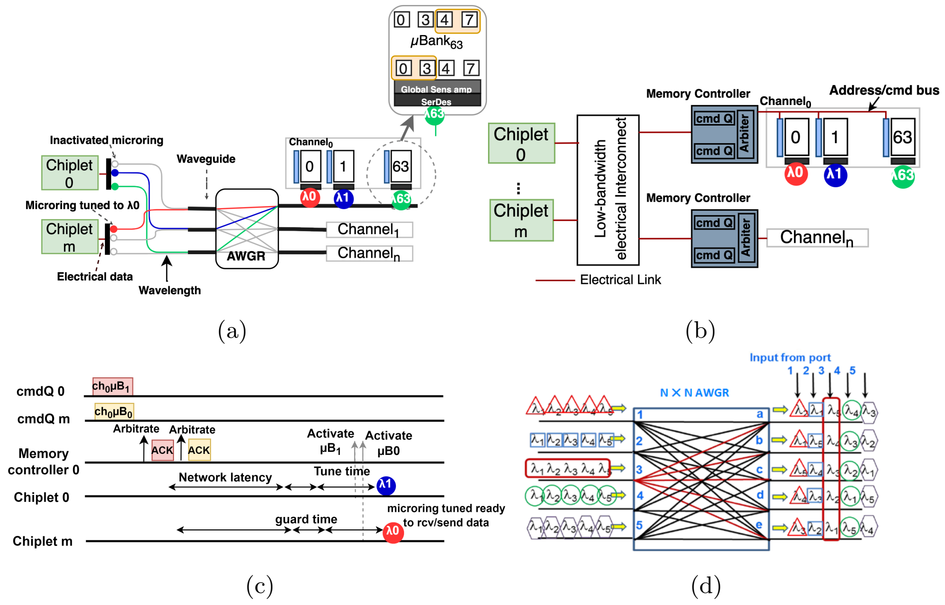


Fig. 2: High-level Overview of (a) data plane, and (b) control plane, (c) demonstrates an example of routing scheme in LLM, and (d) shows the wavelength routing property illustration of AWGR.

particular time, all the requestors can send requests to different channels using different wavelengths on a single waveguide connected to a single port of the AWGR; (c) at a particular time any combination of the above could occur. Note that the *only possible contentions are bank conflicts*, which cannot proceed in parallel anyway and are stalled at the memory controllers.

The choice of the number of waveguides, the number of wavelengths per waveguide, and the data rate in the waveguide are *design parameters* which dictate the maximum number of requestors, memory channels, μbanks, and μbank bandwidth. An $n \times n$ AWGR interconnects n memory channel and n requestors (or group of requestors) each connected to n microrings using n wavelengths. The scalability of the system depends on the scalability of AWGR. The number of ports in an AWGR can easily scale up to 64 ports [11]. For larger systems, multiple smaller AWGRs (lower port count) can be used in parallel to provide the all-to-all interconnection as a large AWGR [33].
Due to the small size of control packets, an electrical interconnect can provide sufficient bandwidth for the communication of command and address bits. Therefore, LLM takes advantage of an electrical interconnect for the implementation of the *control plane*.

Figure 2c illustrates an example of our proposed routing scheme, where multiple chiplets are performing write operations. When request 1 from chiplet 0 wins the arbitration in the memory controller (Explained in Section 4.2), the memory controller sends an acknowledgment signal to chiplet 0, allowing it to send data to the memory. Chiplet 0 uses the second ring and tunes it to the wavelength of its destination (in this example μbank 1 is the destination, which operates with blue wavelength). At the same time, chiplet m can use the red wavelength on a different waveguide connected to another port on the AWGR to reach the μbank 0 in the same channel. After issuing a request to the DRAM, data will be ready in the memory at a predefined time later (which is related to the memory access latency). The requestor uses this latency to tune the correct microring (the channel and μbank address indicate which microring must be tuned to which wavelength). Therefore, the memory device needs to have a deterministic response time. Hence, LLM uses a closed-page policy, where the DRAM row buffer is closed immediately after every read or write.

4.2 Memory Controller

LLM redesigns the memory controller to accomplish three main tasks- (i) issuing request at a high rate to increase throughput, (ii) manage arbitration in case of bank conflicts, and (iii) coordinate between requests and data signals (control flow scheme to enable processors to tune the microrings at a particular time).

To improve throughput, we propose reducing the head-of-line-blocking in memory controllers. In a standard memory controller, a bursty sender can overload the entire queue in the memory controller, forcing other processing units to stall. To avoid this, we assigned a single entry queue per requestor (a single or group of processing units) as shown in Figure 3a. These single-entry queues only store the electrical command signals and the data is buffered at the requestor. Then, instead of requiring a complex priority queue (e.g., first-ready first-come-first-serve), we use a round robin arbiter to select an available request from one queue to a free memory μbank.

To maintain consistency between data and control signal, the memory controller must let the requestors know when to tune their microrings. On an LLC miss or write-back, the requestor sends a request to the memory controller. Then, every cycle, the arbiter selects a ready request from one of the command queues. For read requests, the memory controller asserts the appropriate command and address on the electrical command bus (shown in Figure 3a in red). At the same time, the arbiter sends a notification back to the requestor to inform the requestor when the data will appear on the dedicated data bus for that μbank, allowing the requestor to tune its microring to an specific wavelength. We use electro-optically tunable microrings with few-nanosecond tuning speed [40, 28]. The requestor can tune its microring while memory is activating the corresponding row in the memory. The microring at the requestor needs to be tuned to the corresponding wavelength once the memory row is activated. To ensure this, memory controller delays the activation request by guard time of 10 ns.

4.3 Memory Microarchitecture

For irregular workloads, bank conflicts could cause long latency due to their random memory access pattern. Bank conflicts happen when multiple consecutive requests target different rows in the same bank. The impact of bank conflicts on latency is quite high. For instance, in HBM2.0 this latency is approximately 50 ns (precharge latency plus activation latency) [2].

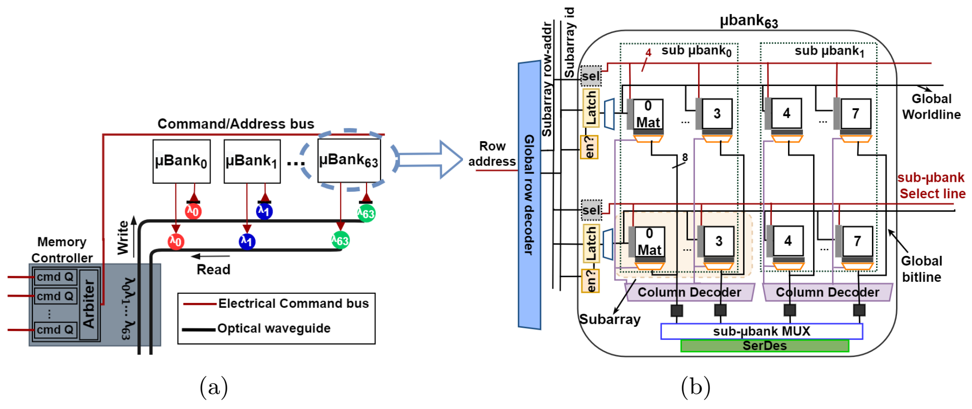


Fig. 3: (a) LLM channel organization. Data and commands are communicated through optical waveguide and electrical bus respectively. (b) μbank architecture which is divided into two sub-μbank that share the same optical data bus through a multiplexer. Each μbank is connected to a microring which is tuned to a certain wavelength.

LLM reduces the probability of bank conflicts by dividing HBM banks into smaller μbanks. In both HBM and LLM, groups of DRAM cells are combined into "mats" which are planar 2D arrays of 512×512 DRAM cells. Mats inside of a subarray are connected to a local sense amplifier and a global bitline connects local sense amplifiers to a global sense amplifier. In LLM μbanks, both the number and size of subarrays are 2× smaller than HBM banks. Lower number of subarrays in LLM μbanks results in shorter global bitlines compared to HBM since each μbank is physically smaller than the HBM banks. LLM further reduces the size of the row buffer by splitting each μbank into two sub-μbanks. This design further reduces the activation energy in LLM which allows for more parallel accesses. Figure 3b shows the detailed architecture of μbank. The impact of our design decisions on the DRAM die size is discussed in Section 5.

In addition to the increased parallelism, this new bank organization also reduces the activation energy. A series of studies have shown that the activation row size directly impacts the DRAM activation energy [30, 13, 18, 47]. Dividing the HBM banks into μbanks and sub-μbanks, reduces the activation row size and the activation energy by 75% compared to HBM2.0.

The second source of contention is the data bus shared by multiple banks inside of one channel. To remove this contention requests targeting different banks need to be t_{BURST} apart. LLM removes the contention on the shared data bus inside the channels by assigning a dedicated optical wavelength to each μbank. Each μbank uses a SerDes and a tuned microring to communicate data.

These microarchitectural changes in DRAM also affect the timing constraint of the memory system. t_{CAS} or t_{CL} defines the time between the column command and the appearance of the data at the memory interface I/O. This makes t_{CAS} the data movement latency within the memory die, which consists of pre-

GSA (global sense amplifier) and post post-GSA latency. Reducing the length of the global bitline (2× smaller), lowers the capacitance which reduces the pre-GSA t_{CAS} by 2×. Post-GSA t_{CAS} also will be 1 ns [16, 43] since the banks send data to the I/O through optical wavelengths. Note that the E-O and O-E latency is discussed in Section 5.

t_{FAW} limits the activation rate in DRAM to limit the drawn current. Since LLM reduces the number of activated bits by 4×, it can activate 4× more rows compared to HBM2.0. In HBM2.0, t_{FAW} is 12 ns. If the command bus works at a high frequency of 2 GHz, memory controller can issue the maximum of 24 activations which is still lower than the limitations of t_{FAW} in LLM (32 activations). Therefore, the parallelism in LLM channels is not limited by the power delivery constraints.

t_{BURST} is the time to transfer the data for a single DRAM request on the I/O bus. With 32 Gb/s data bus bandwidth and 64 byte data, the t_{BURST} in LLM is 16 ns. However, since each μbank in LLM has a dedicated data bus increasing t_{BURST} does not affect the requests targeting different μbanks in one channel. In a system with a shared data bus, the long t_{BURST} increases the serialization effect, enforcing all requests going to different banks in each channel to be t_{BURST} apart. The dedicated data bus eliminates the bus contention in LLM.

4.4 LLM Organization and packaging

LLM dies can be organized as both 3D stacks (similar to HBMs) or non-stacked DRAMs (similar to GDDR memories). In this study, we assume that the LLM dies are organized in 3D stacks to offer increased capacity and bandwidth. To this end, we propose using the innovatively new enabling technology called Vertical Optical Interconnects (VOIs) [48] to replace the TSVs. These optical vias allow substantially higher bandwidth and scaling with number of channels, while keeping the area and number of I/O pins the same. In 3D stacked LLM, data can be moved between μbanks in different layers vertically through optical links. Thus VOIs can replace most of the electrical copper TSVs. Werner et al. explored the bandwidth and scalability advantages of VOIs in 3D stacked memories [45].

We place memory stacks, AWGR, and compute cores on the same package substrate and use a previously proposed technique for intra-package communication [41, 15]. This approach uses dedicated processor node chiplets, and memory node chiplets with embedded SiPh transceivers. For instance the processor node chiplet consists of SerDes, SiPh transceivers, and the compute core dies. The dedicated SiPh transceivers are connected to the chiplets through Si bridges (which are ideal for short-distance electrical interconnection) and optically to AWGR through polymer waveguides. The memory node has SiPh transceivers embedded inside and can use polymer waveguides to connect to AWGR. The polymer waveguides are integrated on top of the organic package substrate and provide connectivity to AWGR. SiPh is ideal for long-distance, inter-package communication, enabling this system to scale out to multiple packages. The multipackage system uses a polymer waveguide for interconnecting separate packages for computing cores, AWGR, and memory stacks without performance and energy degradation.

5 Methodology

To evaluate the performance and latency of LLM, we used the gem5 simulator version 21.0 [26] with both synthetic workloads and full-system (with Linux kernel version 5.2.3). We modeled the network interconnect with Garnet3.0 and the cache hierarchy using Ruby to evaluate the system architecture.

We compared our design with high bandwidth memory systems such as HBM2.0. In addition, we used two state-of-the-art memory systems with more memory level parallelism. The first one is HBM2.0, with added subarray level parallelism for lower memory access latency. We augmented HBM2.0 by adding techniques from Kim et al. [22]. Throughout the paper, we refer to this as HBM-SALP. The second one is a highly concurrent memory system with 4× higher bandwidth than HBM2.0. In this architecture, the memory banks are finer and more independent. A narrow electrical bus with 4× higher datarate compared to HBM2.0 is assigned to these fine-grain memory banks. This design is our interpretation of Fine-Grained DRAM, and we refer to it as FGDRAM [30]. FGDRAM shows the benefits of incorporating μbanks without the contentionless optical data plane.

To be able to fully stress the bandwidth, we used *synthetic traffic* with different access patterns both with high and low locality. We used three different traffic patterns: *Stream*, *Random*, and *GUPS*. The Stream and Random traffic create a sequence of requests with linearly increasing and uniform random distributed addresses respectively. They both generate requests at user-specified frequencies. GUPS is a data dependent application [27] with a random distribution over memory addresses.

Using traffic generators is a *processor architecture agnostic* evaluation allowing these results to be portable whether LLM is used in a CPU, GPU, or accelerator platforms. Using traffic generators also enables experiments with different network injection rates to model memory intensive workloads that can fully stress the high bandwidth of our proposed memory system.

For the synthetic traffic simulation we used 32 traffic generators. For this experiment we scaled our high bandwidth baseline memories to reach the same peak bandwidth as LLM stack which is 4 TB/s (iso-bandwidth). In these iso-bandwidth experiments, both HBM and HBMSALP are given 8× the channels of LLM and FGDRAM 2× compared to LLM.

For latency and overall evaluation, we ran real workloads in the gem5 simulator. We used applications such as GAP benchmark suite (GAPBS) [7] as a representative for irregular workloads due to their random memory access pattern. Table 2 shows the system configuration. We used a multiple core CPU system, each with two levels of cache hierarchy.

Latency Parameters: The memory system needs to model both the network latency (which also includes the O-E and E-O and SerDes latencies) and the DRAM timing constraints. Both of these timings are included in our simulation platform. Due to the different bank and channel organizations, some timing constraints are different from LLM and HBM2.0. Table 2 illustrates the changed timing constraints between HBM, FGDRAM, and LLM. We assumed an optical traversal of 1 ns [16, 24]. We are using a low-power 16 Gb/s SerDes for seriliazing/deserializing 32 bits of data from global sense amplifiers, resulting in 2 ns

Table 2: Full System Simulation Parameters

Parameter	Description	Timing parameter (ns)	HBM2.0[2]	FGDRAM[30]	LLM
CPU	16 cores ; x86 @ 4GHz	tCAS	16	16	5
Caches	private 32 kB L1I/D, 2/8-way per core	tBURST	4	16	16
	private 512 kB, 8-way L2 per core	tFAW	12	12	12
	directory coherence	activates in tFAW	8	32	32
Memory	8 DRAM channels				

Table 3: Silicon Photonic device parameters

Parameter	Value	Parameter	value	Parameter	value
VOI loss	1.3 dB	Photodetector loss	0.1 dB	Modulator Insertion loss	1 dB
Waveguide loss	0.5 dB/cm	Filter through loss	0.1 dB	Power Margin	3 dB
Filter drop loss	1.5 dB	Receiver Sensitivity	-17 dBm	Laser efficiency	14 %
Coupler: Fiber-to-Package	3 dB	AWGR crosstalk	-20 dB	AWGR loss	1.8 dB

latency. We assume that the E-O, O-E conversion latency takes 35 ns [40, 28]. We also modeled the electrical control plane in LLM with a network latency of 20 ns, which is a conservative assumption in our system.

Power Model: For the power modeling of the optical interconnects, we used values for 65 nm CMOS [24, 46] and scaled it down to 28 nm using SPICE models [46, 24]. The laser efficiency is based on commercially-available comb lasers [3]. Table 3 illustrates the details of our silicon photonic devices.

Area: We compared the area of LLM stack based on both microarchitectural changes and the optical circuitry we have added to the memory microarchitecture design. We compared the area for a 4 die stack (4Hi) LLM and HBM. The dimensions of HBM dies are typically 5.5 mm × 7.7 mm [25].

Each μbank includes SiPh transmitter and receiver circuitry (5 μm pitch size), and a 16 Gb/s serializer-deserializer (SerDes) with an area of 0.0045 mm^2 (estimated using TSMC 28 nm CMOS process). Two waveguides are connected to each memory channel, each with 2 μm pitch size [48]. A 4Hi HBM requires 1024 TSVs for data but LLM requires only 32 VOIs. Overall, optical circuitry add 4.94% area overhead compare to a HBM stack.

LLM also requires 2× more column decoders and 4× more global sense amplifiers. Dividing each μbanks to sub-μbanks adds additional circuitry such as 4 bit wordline-select, and sub-μbank multiplexer. These area overhead are equal to FGDRAM and subchannel [30, 10] which are 4.67%. LLM also requires latches to enable subarray level parallelism. Each latch requires 2 μm^2 area. In total microarchitectural changes to DRAM adds an additional 4.8% area overhead. A 4 stack-high LLM requires 9.74% area overhead compare to HBM2.0.

6 Evaluation

6.1 Synthetic Traffic Evaluation

In the first experiment, we ran stream and random synthetic traffic with different traffic rates to see how latency and throughput change as we increase the traffic rate. Figure 4 shows both the achieved throughput and the average access latency for read-only memory requests under varying injection rates. With stream traffic, all memories can achieve high throughput. However, under high injection rates, LLM has lower latency than the other designs due to its low latency interconnect and zero data queuing at the memory controller. At very low injection rates, HBMSALP has a lower average latency due to increased page hit rate and the SALP optimizations [22]. Since LLM uses closed-page policy for applications with high locality LLM will not show significant reduction in latency compered to HBMSALP. However, at all injection rates LLM has lower latency than FGDRAM and HBM.

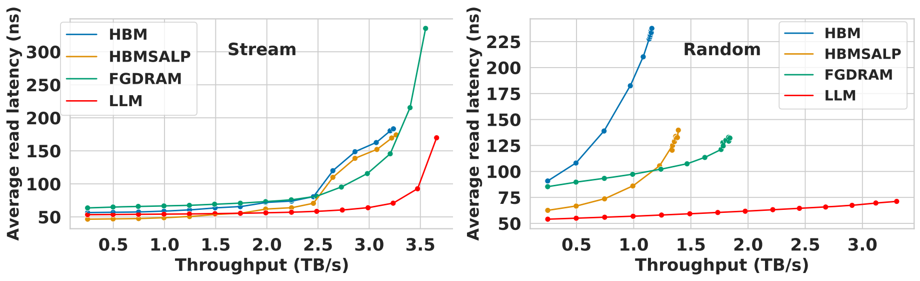
For random traffic, Figure 4b shows that LLM has much lower latency for all injection rates. The main reason HBM's latency increases even under a relatively low injection rate is due to DRAM row buffer misses which incur high latency. These row buffer misses cause contention in the memory controller which results in a high queuing delay. For LLM, reducing the size of the queues in the memory controller and using a closed-page policy leads to low latency under high injection rates. This low queuing is unlike HBM and FGDRAM which experience significant increase in latency as the traffic rate increases. Figure 4b shows the biggest difference between LLM and prior technologies. *LLM can achieve nearly the same throughput with random traffic as with streaming traffic.* In contrast, the best other technology, FGDRAM, can only achieve approximately 50% of its peak theoretical bandwidth under a random access pattern. The difference between LLM and FGDRAM, also shows that simply adding parallelism in the memory subsystem (μbanking) without re-architecting the entire datapath will not remove the contention in the system; it will simply move the contention to another point in the datapath.

To increase complexity in our synthetic traffic experiments, we applied the Giga Updated Per Second (GUPS) benchmark which has data dependent accesses. We measured the performance of these systems based on the GUPS as defined by the benchmark. Similar to Random and stream we used iso-bandwidth test for GUPS. Figure 4c show that even when given significantly more I/O (and cost) HBM and FGDRAM cannot match LLM's performance for this irregular workload.

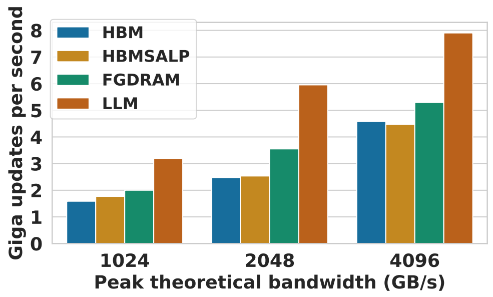
Although HBMSALP adds more intra-bank parallelism compared to HBM, Figure 4c shows it does not achieve considerable performance improvements. This result demonstrates the importance of optimizing the memory system for both bandwidth and latency. Even for latency-critical workloads like GUPS, the bandwidth can also be the limiting factor. Only optimizing for latency does not necessarily lead to the best performance.

6.2 Irregular Workloads

In a more realistic setup, we used gem5 21.0 full system mode to compare LLM with, HBM, HBMSALP, and FGDRAM in a system with 8 processing cores and 8 memory channels (iso-capacity configuration of different memory technologies) as opposed to the iso-bandwidth tests used in the synthetic traffic experiments. Though it is difficult for us to estimate the costs of each technology, this iso-capacity experiment compares the performance in a real system setting with each technology given approximately the same amount of resources. Due to the extensive time of simulation for each system configuration, we created traces for



(a) (b)



(c)

Fig. 4: Iso-bandwidth synthetic traffic with (a) Stream, (b) Random, and (c) GUPS traffic pattern. (a-b) Comparing the average read latency and throughput for different injection rates and access patterns. (c) GUPS traffic, shows even with the same peak bandwidth LLM provides more parallelism resulting in 2× improvement on average performance compared to HBM (with 8× more channels).

8 core system and extended it to 16 core configuration. This enabled us to stress the bandwidth of the system under the same traffic pattern. We used 64 × 64 AWGRs with 64 wavelengths.

For the first experiment we compared the average latency for DRAM access, the queuing latency at the memory controller, and the average network latency. Figures 6(a–c) show the normalized comparison between these memory systems. For all workloads LLM has significantly lower queuing at the memory controller which is what we expected based on lack of data queuing at the memory controller. Also, the network latency for LLM remains smaller for all workloads because in large scale systems with higher crossbar radix electrical interconnect latency is higher. Compared to HBM, FGDRAM shows lower queuing latency which indicates the benefits of added parallelism at the memory microarchitecture without the optical datapath. Comparing LLM and FGDRAM, the queuing latency is on average 3× lower which shows the benefit of the co-design architecture of the memory controller, the interconnect design, and the all-optical data path. Finally, for the device latency (Figure 5c), all systems have approximately the same latency except FGDRAM which is higher due to the larger t_{BURST}.

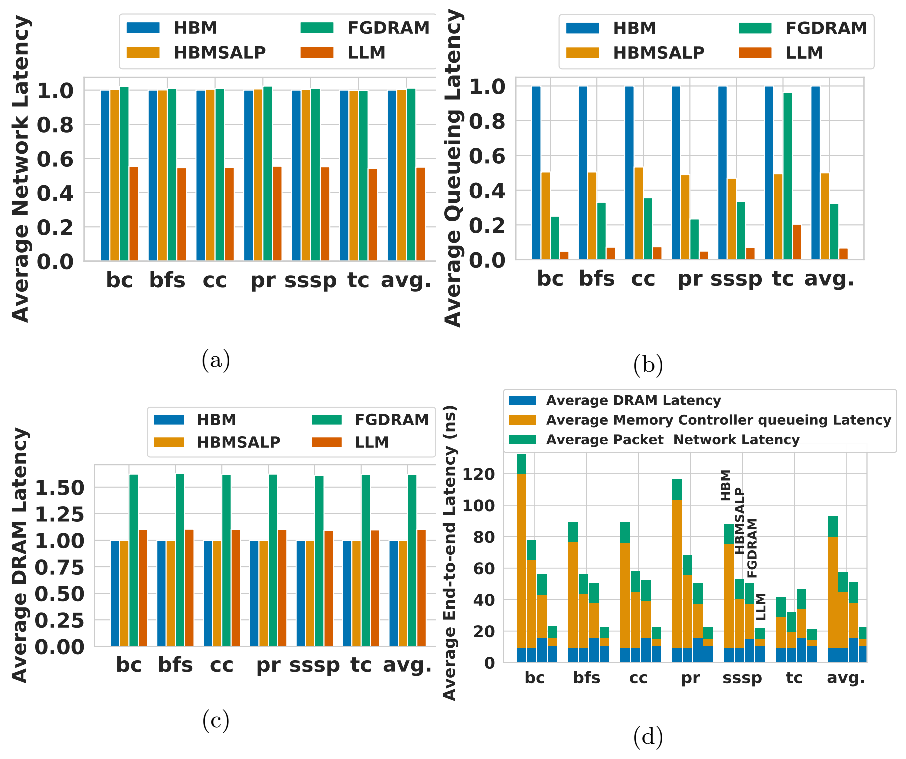
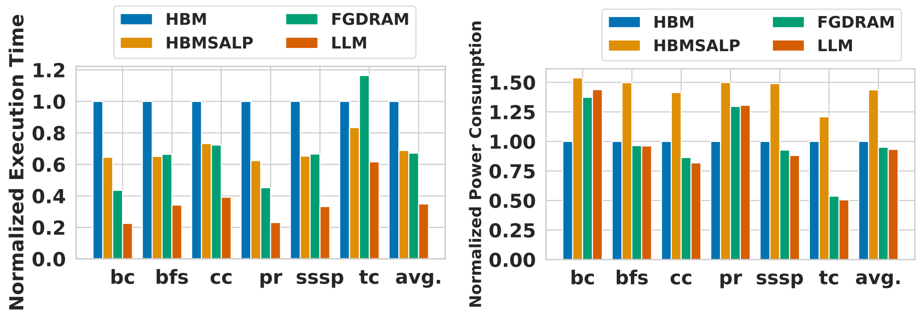


Fig. 5: Average latency normalized to HBM2.0 for (a) network (b) queuing (c) memory device, and (d) shows the average end-to-end latency. (a) shows LLM achieves in average 2× lower network latency, 1.1× higher DRAM latency due to the long bus latency, and (b) indicates 10× lower queuing latency compared to HBM2.0.

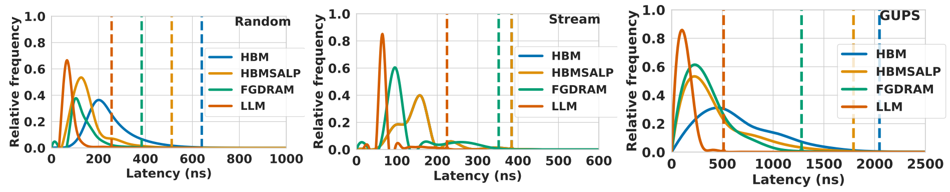
Figure 5d shows the total average latency of the three components (device, queuing, and network latency). This shows that for all systems except LLM, queuing latency is the dominant portion of the time (broken out in Figure 5b). Figure 5d indicates the memory intensity of the workloads as well. For instance, *tc* has lower average end-to-end latency with lower queuing compared to the other workloads. Thus, optimizing just for throughput will not improve the execution time for this workload (e.g., FGDRAM does not improve performance for *tc* as shown in Figure 6a since it sacrifices latency for bandwidth).

Figure 6a compares the execution time of GAPBS workloads for HBM, HBM-SALP, FGDRAM, and LLM. Compared to HBM, LLM provides 3× reduction on average execution time. For the more memory intensive workloads, the increased bandwidth of LLM provides reduced execution time. Importantly, for the lower intensity workloads, LLM also provides an improvement over the other technologies (most notably FGDRAM running *tc*) due to its lower contention on the shared data bus.



(a) (b)

Fig. 6: Execution time (a) and power consumption (b) normalized based on HBM2.0. LLM provides in average 3× lower execution time while maintaining same power consumption compared to HBM2.0.



(a) (b) (c)

Fig. 7: The latency distribution for different memory systems under 3 types of synthetic traffic: (a) Random, (b) Stream, and (c) GUPS. LLM has a lower 95th percentile (shown as dashed lines) and therefore has lower latency variation. In (b) HBM and HBMSALP have the same distribution of latency.

6.3 Energy and power analysis

The DRAM access energy consists of activation energy, data movement energy, and I/O energy. We used the HBM2.0 energy values from O'Conner et al. [30]. The activation energy directly depends on the number of bits in a row that get activated. Similar to FGDRAM [30], LLM reduces the size of the row by a factor of 4×, and therefore, we reduce the activation energy to 227 pJ for LLM from 909 pJ in HBM 2.0. Pre-GSA energy is the energy of moving data from local and master bitlines to the global row buffer, and it depends on the length of bitline. Since we are reducing the size of the global bitlines, this energy will also be reduced to 0.755 pJ/bit from 1.51 pJ/bit in HBM2.0.

LLM uses optical links to move data between μbanks and processing cores. Therefore, both I/O and post-global sense amplifier energy values are equal and are independent of laser, SerDes, and modulation circuitry. For this SiPh stack, we used the parameters shown in Table 3 to match realistic current technologies. We found the total I/O energy (including laser, SerDes, modulation circuitry) to be 760 fJ/bit. In comparison, for conventional DRAM the I/O requires 800 fJ/bit [30], which is expected to increase as the height of DRAM stacks increases.

Figure 6b illustrates a comparison of overall memory power consumption normalized to HBM between a DRAM stack interconnected electrically with TSVs against LLM with SiPh DRAM stacks. As shown, the LLM is approximately the same power as the electrically implemented FGDRAM showing the SiPh implementation is feasible to integrate in a chiplet-based package. In some cases, the power is higher, mostly due to the higher bandwidth that FGDRAM and LLM enable compared to HBM.

6.4 Latency Variation

Finally, we analyzed the latency variation in each memory system. In current systems, the main cause of latency variation in the system is queuing. Thus, one of the byproducts of our low contention memory system should be lower latency variation. Figure 7 shows the distribution of access times for each technology under stream, random, and GUPS synthetic traffics using 16 memory channels. This figure also shows the 95^{th} percentile latency with dashed vertical bars.

Figure 7 shows that LLM achieves significantly lower and more predictable latency compared to other technologies. In general HBM has the broadest distribution, with FGDRAM and HBMSALP having slightly less variation than HBM for Random and GUPS traffics. On average LLM has $3\times$ lower 95^{th} percentile latency compared to HBM which can be translated into $3\times$ lower memory latency variations. We see similar results for the full system graph workloads as well.

7 Related Work

Several studies have shown the benefits of using photonics to increase bandwidth and reduce data movement energy for processor/memory communication [39, 6, 5, 34, 45, 37]. Although these studies reduce contention at the interconnect, they did not contribute to increasing memory performance at the microarchitectural level. LLM extends these prior works by (a) reducing in-memory activation and data movement energy, allowing for higher parallelism, and (b) integrating optics inside of the memory channel and co-designing the memory controller to facilitate both bandwidth and *latency* improvements.

Previous work on DRAM energy [47, 18, 13, 30] showed the benefits of reducing activation energy while maintaining a higher bandwidth than HBM2.0. These studies are still bounded by the processor/memory data movement energy. LLM extends these prior works by exploiting silicon photonic interconnects. Optical links do not suffer from the distance/bandwidth trade-off that impacts electrical interconnects. This allows LLM to achieve a low energy data movement in a chiplet based architecture while achieving higher peak bandwidth than the previous studies.

Creating smaller channels with narrower data bus and higher datarate is the technique used both in in the industry (with HBM2.0 and HBM2.0 pseudo-channel mode, and GDDR) and research [30] to enable high throughput memory systems. However, they do not consider optimizing the memory for latency. Furthermore, they use deep queues for bandwidth improvements which will result in higher latency. In contrast, LLM is a redesign of the complete memory subsystem. Decoupling data and control signals in the LLM allows for bandwidth and latency improvement at the same time.

Previous work, has explored many different avenues for decreasing DRAM latency including changing the DRAM controller [8], segmenting and shortening bitlines [22] and caching and paging policies [21]. Although these techniques proved to be effective in reducing the DRAM access latency, they are not optimized for irregular applications and in some cases can increase memory access latency variability. Wang et al. improved latency for irregular workloads by creating a low-cost DRAM substrate that enables data relocation [42]. Although effective for irregular workloads they have not shown any benefits for applications with high locality and the effects on memory latency variations. LLM reduces the amount of data queuing on the entire path and assigns a dedicated data path between each requestor and memory μbank. This technique reduces latency in both regular and irregular workloads but it also reduces memory access variability due to low queuing on the path.

8 Conclusion

In this paper, we investigated a new memory system that is optimized for applications with both regular and irregular access patterns with poor spacial locality. LLM introduces lower execution time compared to the baseline HBM2.0 systems. It also utilizes an all optical data communication fabric that provides a direct contention-free data link between processing cores and memory banks. The use of optical interconnects, optical links, and the new memory microarchitecture improve data movement, reduces activation energy and provides higher bandwidth/mm^2. By incorporating all these methods, LLM can reduce the execution time and energy with a modest area overhead. The cost increase for optoelectronic integrated LLM would be around 30% compared to electronic only HBM2.0. However, LLM achieves around 3$\times$ better execution time while maintaining the same power consumption as HBM2.0.

Due to low-contention data access in LLM, we believe that LLM-like designs can improve the performance in other computing systems. As future work we would like to evaluate the architectural impact and benefits of LLM in other systems such as graph accelerators.

References

1. Ayar Labs Realizes Co-Packaged Silicon Photonics – WikiChip Fuse, https://fuse.wikichip.org/news/3233/ayar-labs-realizes-co-packaged-silicon-photonics/.
2. JEDEC, https://www.jedec.org/sites/default/files/docs/JESD212.pdf.
3. Thermistor Specification Fiber Specification an exemplary Eye Diagram of one F-P mode Externally modulated at 2.5GHz filtered-out single channel www.innolume.com.
4. Zen - Microarchitectures - AMD - WikiChip, https://en.wikichip.org/wiki/amd/microarchitectures/zen.

5. Batten, C., et al.: Building many-core processor-to-dram networks with monolithic cmos silicon photonics. International Symposium on Microarchitecture (MICRO) pp. 8–21 (2009)
6. Beamer, S., et al.: Re-architecting dram memory systems with monolithically integrated silicon photonics. In: Proceedings International Symposium on Computer Architecture (ISCA). p. 129–140. IEEE (2010)
7. Beamer, S., et al.: The gap benchmark suite. arXiv preprint arXiv:1508.03619 (2015)
8. Carter, J., et al.: Impulse: Building a smarter memory controller. In: Proceedings Fifth International Symposium on High-Performance Computer Architecture. pp. 70–79. IEEE (1999)
9. Chatterjee, N., et al.: Managing dram latency divergence in irregular gpgpu applications. In: Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis (SC). pp. 128–139 (2014)
10. Chatterjee, N., et al.: Architecting an energy-efficient dram system for gpus. In: IEEE International Symposium on High Performance Computer Architecture (HPCA). pp. 73–84. IEEE (2017)
11. Cheung, S., et al.: Ultra-compact silicon photonic 512$\times$ 512 25 ghz arrayed waveguide grating router. IEEE Journal of Selected Topics in Quantum Electronics pp. 310–316 (2013)
12. Cianchetti, M.J., et al.: Phastlane: a rapid transit optical routing network. In: Proceedings of the International Symposium on Computer Architecture (ISCA). pp. 441–450 (2009)
13. Cooper-Balis, E., et al.: Fine-grained activation for power reduction in dram. International Symposium on Microarchitecture (MICRO) pp. 34–47 (2010)
14. Eklov, D., et al.: Bandwidth bandit: Quantitative characterization of memory contention. In: Proceedings of the 2013 IEEE/ACM CGO. pp. 1–10 (2013)
15. Fotouhi, P., et al.: Enabling scalable chiplet-based uniform memory architectures with silicon photonics. In: Proceedings of the International Symposium on Memory Systems (MEMSYS). pp. 222–334 (2019)
16. Grani, P., et al.: Design and evaluation of awgr-based photonic noc architectures for 2.5 d integrated high performance computing systems. In: IEEE International Symposium on High Performance Computer Architecture (HPCA). pp. 289–300. IEEE (2017)
17. Gupta, U., et al.: The architectural implications of facebook's dnn-based personalized recommendation. In: IEEE International Symposium on High Performance Computer Architecture (HPCA). pp. 488–501. IEEE (2020)
18. Ha, H., et al.: Improving energy efficiency of dram by exploiting half page row access. In: International Symposium on Microarchitecture (MICRO). pp. 1–12. IEEE (2016)
19. Hassan, H., et al.: Chargecache: Reducing dram latency by exploiting row access locality. In: IEEE International Symposium on High Performance Computer Architecture (HPCA). IEEE (2016)
20. JESD235A, J.: High bandwidth memory (hbm) dram. JEDEC Solid State Technology Association (2015)
21. Kaseridis, D., et al.: Minimalist open-page: A dram page-mode scheduling policy for the many-core era. In: International Symposium on Microarchitecture (MICRO). pp. 24–35. IEEE (2011)
22. Kim, Y., et al.: A case for exploiting subarray-level parallelism (salp) in dram. In: Proceedings of the International Symposium on Computer Architecture (ISCA). pp. 368–379. IEEE (2012)
23. Kirman, N., et al.: Leveraging optical technology in future bus-based chip multiprocessors. In: International Symposium on Microarchitecture (MICRO). pp. 492–503. IEEE (2006)

24. Li, H., et al.: A 25 gb/s, 4.4 v-swing, ac-coupled ring modulator-based wdm transmitter with wavelength stabilization in 65 nm cmos. IEEE Journal of Solid-State Circuits pp. 3145–3159 (2015)
25. Li, L., et al.: 3d sip with organic interposer for asic and memory integration. In: IEEE 66th Electronic Components and Technology Conference (ECTC). pp. 1445–1450. IEEE (2016)
26. Lowe-Power, et al.: The gem5 simulator: Version 20.0+. arXiv preprint arXiv:2007.03152 (2020)
27. Luszczek, P.R., et al.: The hpc challenge (hpcc) benchmark suite. In: Proceedings of the 2006 ACM/IEEE Conference on Supercomputing. p. 213–es (2006)
28. Matsuo, S.a.o.: Microring-resonator-based widely tunable lasers. IEEE Journal of Selected Topics in Quantum Electronics pp. 545–554 (2009)
29. Nitta, C.J., et al.: On-chip photonic interconnects: A computer architect's perspective. Synthesis Lectures on Computer Architecture pp. 1–111 (2013)
30. O'Connor, M., et al.: Fine-grained dram: Energy-efficient dram for extreme bandwidth systems. In: International Symposium on Microarchitecture (MICRO). pp. 41–54. IEEE (2017)
31. Papistas, I., et al.: Bandwidth-to-area comparison of through silicon vias and inductive links for 3-d ics. In: European Conference on Circuit Theory and Design (ECCTD). pp. 1–4. IEEE (2015)
32. Parekh, M.S., et al.: Electrical, optical and fluidic through-silicon vias for silicon interposer applications. In: IEEE Electronic Components and Technology Conference (ECTC). pp. 1992–1998. IEEE (2011)
33. Proietti, R., et al.: Experimental demonstration of a 64-port wavelength routing thin-clos system for data center switching architectures. Journal of Optical Communications and Networking (JOCN) pp. B49–B57 (2018)
34. Rumley, S., et al.: Silicon photonics for exascale systems. Journal of Lightwave Technology (JLT) (2015)
35. Shacham, A., et al.: Photonic networks-on-chip for future generations of chip multiprocessors. IEEE Transactions on Computers pp. 1246–1260 (2008)
36. Shang, K., et al.: Low-loss compact silicon nitride arrayed waveguide gratings for photonic integrated circuits. IEEE Photonics Journal pp. 1–5 (2017)
37. Shen, Y., et al.: Silicon photonics for extreme scale systems. Journal of Lightwave Technology (JLT) pp. 245–259 (2019)
38. Takada, K., et al.: Low-crosstalk 10-ghz-spaced 512-channel arrayed-waveguide grating multi/demultiplexer fabricated on a 4-in wafer. IEEE Photonics Technology Letters pp. 1182–1184 (2001)
39. Udipi, A.N., et al.: Rethinking dram design and organization for energy-constrained multi-cores. In: Proceedings of the International Symposium on Computer Architecture (ISCA). pp. 175–186 (2010)
40. de Valicourt, et al.: Dual hybrid silicon-photonic laser with fast wavelength tuning. In: Optical Fiber Communications Conference and Exhibition (OFC). pp. 1–3 (2016)
41. Wade, M., et al.: Teraphy: A chiplet technology for low-power, high-bandwidth in-package optical i/o. International Symposium on Microarchitecture (MICRO) pp. 63–71 (2020)
42. Wang, Y., et al.: Figaro: Improving system performance via fine-grained in-dram data relocation and caching. In: International Symposium on Microarchitecture (MICRO). pp. 313–328. IEEE (2020)
43. Werner, S., et al.: Amon: An advanced mesh-like optical noc. In: IEEE 23rd Annual Symposium on High-Performance Interconnects. pp. 52–59 (2015)
44. Werner, S., et al.: Awgr-based optical processor-to-memory communication for low-latency, low-energy vault accesses. In: Proceedings of the International Symposium on Memory Systems (MEMSYS). pp. 269–278 (2018)

45. Werner, S., et al.: 3d photonics as enabling technology for deep 3d dram stacking. In: Proceedings of the International Symposium on Memory Systems (MEMSYS). pp. 206–221 (2019)
46. Yu, K., et al.: A 25 gb/s hybrid-integrated silicon photonic source-synchronous receiver with microring wavelength stabilization. IEEE Journal of Solid-State Circuits (JSSC) pp. 2129–2141 (2016)
47. Zhang, T., et al.: Half-dram: A high-bandwidth and low-power dram architecture from the rethinking of fine-grained activation. In: Proceedings of the International Symposium on Computer Architecture (ISCA). pp. 349–360. IEEE (2014)
48. Zhang, Y., et al.: High-Density Wafer-Scale 3-D Silicon-Photonic Integrated Circuits. IEEE Journal of Selected Topics in Quantum Electronics pp. 1–10 (2018)