

# OCDIMM: Scaling the DRAM Memory Wall Using WDM based Optical Interconnects

Amit Hadke Tony Benavides S. J. Ben Yoo Rajeevan Amirtharajah Venkatesh Akella

Department of Electrical & Computer Engineering

University of California, Davis, CA - 95616

Email: akella@ucdavis.edu

**Abstract**—We present OCDIMM (Optically Connected DIMM), a CPU-DRAM interface that takes advantage of multiwavelength optical interconnects. We show that OCDIMM has at least three key benefits when compared to alternatives such as FBDIMM (Fully Buffered DIMM), which is used in recent products from Sun [1] and Intel. First, replacing the multi-hop store-and-forward network in the FBDIMM architecture by a WDM (wavelength-division multiplexing) based optical interconnect results in significantly lower latency (up to 50% reduction in some configurations). Second, it is scalable to much higher capacities (such as 32 DIMMs per channel) with only a modest degradation in latency. Third, due to the higher data rate of an optical interface and the concurrency offered by multiple wavelengths, OCDIMM offers up to a 90% improvement in memory bandwidth. Most importantly, these benefits can be obtained using off-the-shelf DRAM devices, by making simple modifications to the DIMM circuit board and the memory controller.

## I. INTRODUCTION

As we integrate more cores per chip to satisfy the insatiable demand for high performance computing in both, servers and high-end machines used in gaming and entertainment, there is a pressure to increase the processor-memory bandwidth. In addition, as the application working sets grow, especially with a larger number of cores per chip and more threads per core, there is a need to increase the memory capacity of the system. In a seminal article, Jim Gray and Gordon Bell [2] argue that it is important to build *balanced* computing systems. In a study sponsored by NSF, a blue-ribbon panel chaired by Atkins [3] outlined the requirements of a balanced computing system as a system that provides roughly one byte/sec/flop in terms of bandwidth and one byte of storage for each flop. So, a balanced tera-flop chip of the future, will require 1TB of memory and 1TB/s pin bandwidth. Traditionally, higher bandwidth requirements have been met by wider parallel buses and higher clock frequencies. This approach is not scalable because of the limitations on the number of pins on a package and inherent complexity of running wide parallel buses at very high clock frequencies. Typically, larger capacities have been met by using denser DRAM chips. For example, 2Gb DRAM chips are possible today. However, due to the electrical signaling constraints, the maximum number of DIMMs per channel has been decreasing in successive generations of DRAMs. SDRAM channels supported up to 8 DIMMs, some types of DDR channels support 4 DIMMs, and DDR3 channels are expected to have only 3 DIMMs per channel.

The state-of-the-art technology today to achieve high bandwidth and reasonably high capacity is the FBDIMM (fully-buffered DIMM) architecture [4], which uses a narrow high-speed *point-to-point* interface between the memory controller and memory modules (DIMMs) based on a split-bus mechanism. Even though FBDIMM promises a significant improvement over DDR2/DDR3 based interfaces, it is not scalable beyond 8 DIMMs per channel. Because of the limitation of the store-and-forward protocol employed within the FBDIMM, as we add more DIMMs on a channel the latency increases significantly. Although, there is a variable latency mode of operation in FBDIMM to alleviate this problem, it is not very useful because it makes the memory controller more complex and in most of the cases fixed-latency mode of operation is preferable. As a result, current FBDIMMs might not be able to simultaneously solve the capacity and bandwidth scalability that is required in future multicore chips.

In this paper we evaluate the potential of multi-wavelength optical interconnects between the processor and the DRAM to address these challenges. We propose a simple extension to the FBDIMM architecture that can take advantage of the recent advances in CMOS-compatible nanoscale silicon photonic integrated circuitry [5], [6]. We call this OCDIMM (optically connected DIMM) architecture. OCDIMM does not involve any changes to the DRAM devices, i.e. it is capable of using existing off-the-shelf DRAM chips. The changes are mainly localized to the AMB (advanced memory buffer) on the DIMM and in the memory controller. We modify the DRAM simulator from University of Maryland [7] to model the OCDIMM architecture including the optical-to-electrical conversions (modulators, propagation delay through the waveguide and the detectors). We demonstrate three key benefits of OCDIMM over the state-of-the-art FBDIMM technology.

- 1) OCDIMM is *scalable* to much higher capacities (32 DIMMs per channel) compared to FBDIMM without significant impact on latency.
- 2) By replacing the store-and-forward network and protocol in FBDIMM with a WDM-based optical *bus* we can reduce the *average-latency* significantly. Up to 50% reduction in the latency can be obtained in some configurations even after taking into account the additional delays due to optical to electrical and electrical to optical

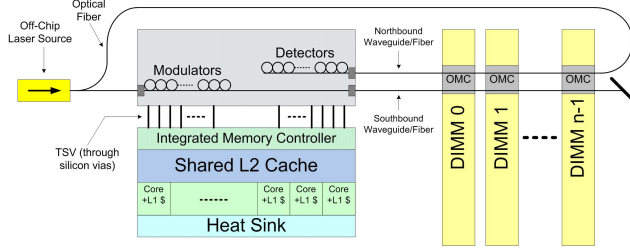


Fig. 1. Top Level System

conversions.

- 3) We can improve the bandwidth significantly over FBDIMM (up to 90% in some configurations) by taking advantage of the additional dimension for concurrency in terms of multiple wavelengths.

The rest of this paper is organized as follows. First we start with an overview of the OCDIMM architecture and explain how it works. Next, we describe our experimental methodology including how OCDIMM was modeled in the DRAM simulator and the different workloads that were considered. Then we present our results where first, we validate our experimental methodology by comparing our results with published results for DDR2 and FBDIMM [8]. Next we compare the latencies of different OCDIMM configurations with fixed-mode and variable mode FBDIMM as a function of number of DIMMs per channel. This will demonstrate the scalability benefit of OCDIMM. Then we evaluate the bandwidth advantages of OCDIMM over FBDIMM. Finally, we explore some advanced strategies that could be implemented in the memory controller to take advantage of multiple wavelengths. We conclude the paper with a discussion on related work and directions of future work.

## II. OCDIMM PHYSICAL ARCHITECTURE

OCDIMM is derived from fully-buffered DIMM (FBDIMM) architecture [4]. Next we present a very short overview of the FBDIMM architecture. The reader is referred to the excellent textbook [8] and tutorial paper [9] for more details on the FBDIMM architecture and its performance evaluation. FBDIMM memory architecture replaces the shared parallel interface between the memory controller and the DRAM chips with a point-to-point serial interface between the memory controller and an intermediate buffer, the Advanced Memory Buffer (AMB). The on-DIMM interface between the AMB and the DRAM modules is identical to that seen in DDR2 and DDR3 systems. The serial interface is split into two unidirectional buses, one for read traffic (northbound link) and one for command/write traffic (southbound link). FBDIMMs adopt a packet-based protocol that bundles commands and data into frames that are transmitted on the channel and then converted to the DDR2/DDR3 protocol by the AMB. Since each DIMM-to-DIMM connection is a point-to-point link, a channel becomes a multi-hop store and forward network. All frames are processed by the AMB to determine whether the

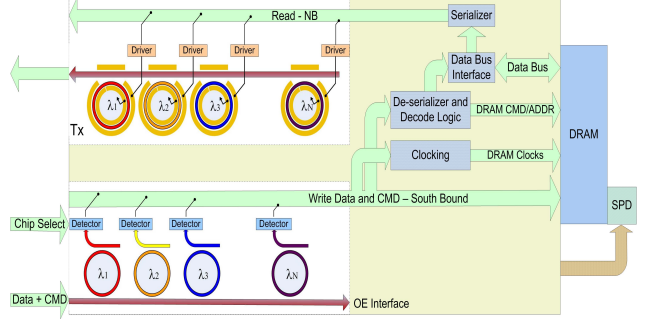


Fig. 2. Internal Details of the OMC

data and commands are addressed to the local DIMM. This is a limitation with FBDIMM which is naturally overcome with OCDIMM by using a optical bus instead of multi-hop network.

Figure 1 shows the top-level physical architecture of a computing system that uses OCDIMM. We assume the CPU is a 3D stacked die with the electronics on one layer and optical transceivers (modulators and demodulators) in a different layer with an off-chip laser source powering the optical interconnect. The northbound and southbound buses in the FBDIMM architecture are replaced by optical fibers. Furthermore, we assume that each fiber can transport multiple wavelengths up to a maximum of 64. However, as we will show, it is not necessary to have 64 wavelengths to reap the benefits of an optical interconnect. Even with much fewer wavelengths (e.g. 4), significant benefits can be realized, as described in the results section of this paper. We replace the AMB on each DIMM with an Optical Memory Controller (OMC) (details shown in Figure 2), which is responsible for the communication to the DRAM from the bus. As in the FBDIMM protocol, the commands and write data will be injected via the southbound fiber to the OMC and the read data is received on the northbound fiber. We assume a dedicated wavelength for clock distribution. The clock will be extracted from its transportation wavelength and the data will be de-serialized in order to be put out as commands and data to the DRAM device.

Microresonator [10]–[12] based modulators convert the electrical signals of the on-chip integrated memory controller to the optical domain for transport over the optical fibers. The modulators and demodulators are quite compact (order of tens of microns) and capable of data rates of 10Gbps or higher. The optical signals are demodulated at the OMC using microresonators and processed by the traditional electrical signal chain in a FBDIMM AMB as shown in Figure 2. So, basically, we have a single-hop (broadcast) bus instead of a multi-hop store and forward network.

The read and write data is organized as a packetized frame relay protocol which is realized using multiple transmissions on the optical bus. The number of transmissions can be reduced if there are more wavelengths available. Writing data to the DRAM is simple. Depending on the DRAM mode, a

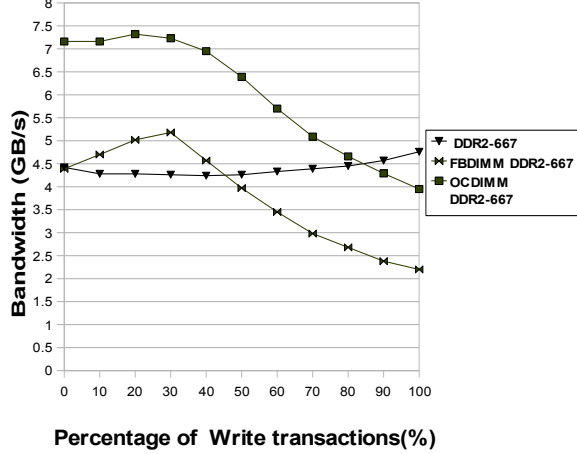


Fig. 3. **Bandwidth characteristics of different memory architectures varying write transaction percentage.** This figure shows bandwidth characteristics for a randomized trace with 2 DIMMs on a single channel in open page mode, with greedy scheduling and queue depth of 8. Percentage of WRITE transactions is varied from 0% to 100% along the X axis and bandwidth is plotted on Y axis in GB/s.

RAS command, followed by some number of CAS commands is sent to the target DIMM, interspersed with the data on the write channel. Once the commands and data are converted to the electrical domain via the E/O/E interface in the target domain, the operation is exactly the same as with a standard DDR2/DDR3 device and FBDIMM. No modification is necessary to the existing DRAM devices. The only additional hardware is at the DIMM level, in the form of the modulators and demodulators to convert the data from electrical to optical domain and vice versa as shown in Figure 2. Reading data from a given DIMM is more complicated because the read subchannel is shared. Once a DIMM is ready to send data, it has to acquire the read-subchannel because another DIMM could be using it to send data back to the memory controller. In general, this requires an arbiter, but to keep the design simple, initially, like FBDIMM, we assume that the memory controller statically schedules the read transactions such that there are no bus-conflicts.

If extra wavelengths are supported, OCDIMM can do certain optimizations in order to keep small number of DIMMs in the active state. OCDIMM use these extra wavelengths as a chip select signal to activate a DIMM before sending out the command/data frame on southbound bus. Only destination AMBs will read the data on southbound bus. Additional cost in preselecting is negligible in terms of the delay but as the number of DIMMs on a channel increases, demand on such additional wavelengths increases.

It is important to note that in the operation of OCDIMM the frame cycle(time required to send a frame) for the northbound and southbound buses will be kept the same as in the FB-

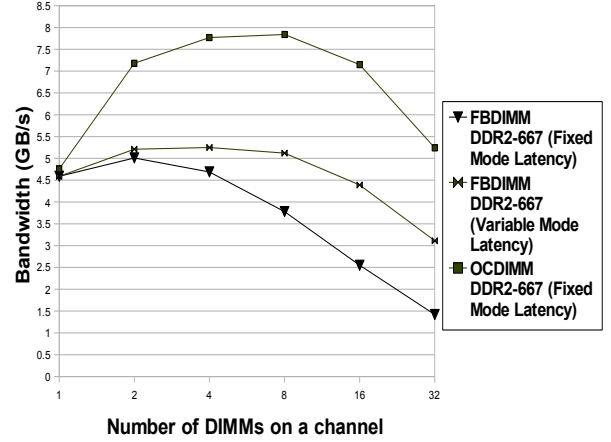


Fig. 4. **Bandwidth Characteristics of FBDIMM and OCDIMM** Figure shows the single channel bandwidth observed against the channel capacity in open page mode, with greedy scheduling, READ-to-WRITE transaction ratio kept at 2:1 and queue depth of 16. Number of DIMMs on a single channel is varied from 1 to 32 along the X axis and the Y axis plots bandwidth in GB/s. All DIMMs are identical DDR2-667Mbps with 8 banks, 8 byte channel width.

DIMM protocol. Given that the optical interface is clocked at 10 Gbps, the controller clock to DRAM clock ratio is now 15:1 instead of 6:1 as in FBDIMM. This increased ratio allows new DRAM interface to transfer 32 bytes per bundle for reads on the northbound lane while also transferring 6 commands per bundle or 1 command plus 16 bytes of data on the southbound lane.

### III. EXPERIMENTAL METHODOLOGY

We modified the DRAM simulator (DRAMsim) from University of Maryland [7] to model the OCDIMM. We chose to use the DDR2-667Mbps to model latencies within the DIMM. All DIMMs are identical and have 8 banks with 8 byte channel width, leading to 5.3GB/s peak transfer rate from each DIMM. We used ranks to represent DIMMs, as done in [9]. Each DIMM has only one rank, and to add a DIMM we add a rank in the simulator configuration. The FBDIMM memory controller is set to run at 6x DDR2-667Mbps (i.e at 4002 MHz). Each AMB uses up/down buffers for write/read requests, and the buffer count is set to the number of banks on a DIMM (8). The flight time from the controller to the first DIMM is assumed to be zero, while the time from DIMM-to-DIMM is modeled as 1.5ns (this includes the minimum resample mode delay). For simulations of OCDIMM, the DIMM-to-DIMM flight time is set to 300ps, which includes the transmission delay for a distance of up to 2cm, O/E/O conversions, and skew adjustment. All optical buses operate at 10GHz.

OCDIMM is modeled in two different ways:

**When frame cycle = DRAM cycle** - The time required to send a frame is equal to one DRAM cycle. (For DDR2-

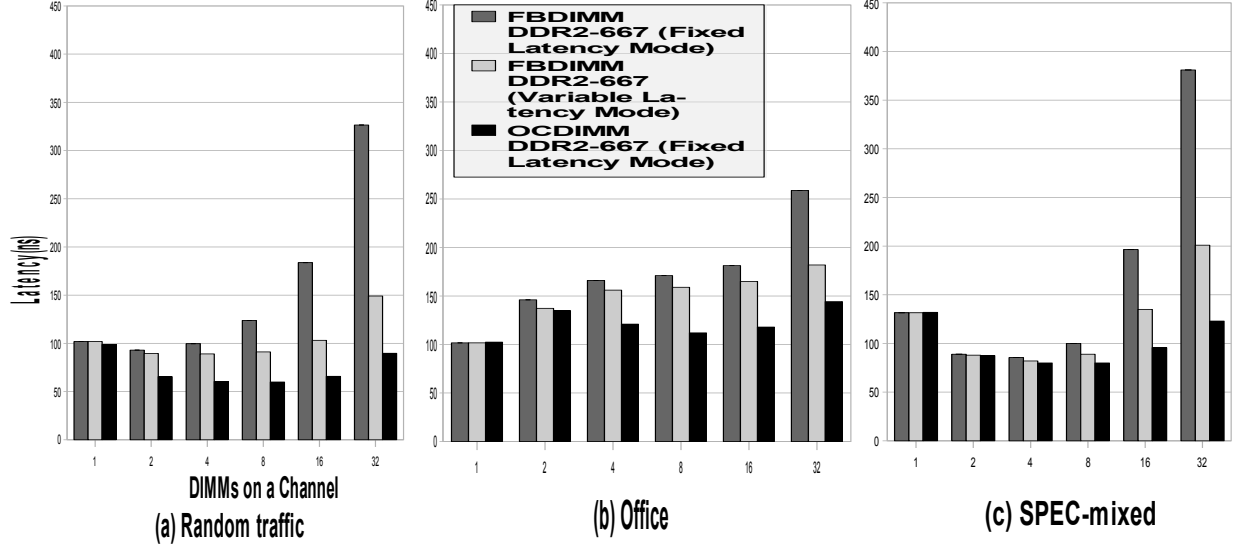


Fig. 5. **Latency Characteristics of FBDIMM and OCDIMM** This figure shows the average latency observed against the channel capacity. Fig (a) Random traffic is modeled with open page mode, greedy scheduling, READ-to-WRITE transaction ratio kept at 2:1 and queue depth of 16. Fig (b) and (c) shows average latency observed for OpenOffice and SPEC-mixed workloads. Number of DIMMs on a single channel is varied along X axis and the Y axis plots latency in nano-seconds. All DIMMs are identical DDR-667Mbps with 8 banks, 8 byte channel width. Average latency is calculated as time between an arrival of a transaction and completion of a transaction.

667Mbps, frame cycle = DRAM cycle = 3ns). Depending upon the available wavelengths ( $\lambda$ ) the amount of data that can be carried by a frame varies.

**When frame cycle  $\neq$  DRAM cycle** - The frame time is varied according to  $\lambda$ . To model this correctly, when the frame cycle is less than the DRAM cycle, extra delays are added in the AMB to ensure that it issues commands on the correct DRAM clock cycle.

#### IV. RESULTS AND DISCUSSION

In this section we describe the simulation results of OCDIMM. First, we validate our simulation model by comparing the performance results of DDR2 and FBDIMM configurations with published results. Next, we evaluate OCDIMM with respect to scalability, latency and bandwidth. Finally, we present the impact of an intelligent memory controller that can exploit the wavelength-level concurrency offered by the OCDIMM architecture. We begin with a brief overview of the address mapping strategy and and workload characteristics.

*a) Transaction address mapping::* Every transaction is assigned a uniformly distributed random address. We clubbed rank address bits with channel address bits, so as to map consecutive cachelines to different DIMMs. This gives maximum possible parallelism and deeper channels can schedule more transactions at any given time.

*b) Workload characteristics::* We use three types of workloads - random, a mixture of SPEC traces and traces from OpenOffice which are described next

**Random traffic** We used a random, uniformly distributed request arrival rate. This type of workload attempts to schedule as many transactions as possible, hence it is used in the experiments to measure maximum bandwidth with  $\approx 100\%$  utilization. Each transaction reads or writes a cacheline of 64 bytes. Read-to-write transaction ratio was kept at 2:1 as most of the workloads are observed to have more read request than writes (Page 550-551 of [8]).

**SPEC-mixed traces** SimpleScalar [13] was used to extract main memory access traces of four SPEC-CPU2000 benchmarks (gcc, gzip, parser, vortex). L1 data and instruction caches were 64KB, 2-way set-associative, with 64 bytes blocks and unified L2 cache was 1MB, 16-way set-associative with 64 bytes blocks.

**OpenOffice trace** Bochs 2.3.5 [14] was modified to include a 16 way set associative 2MB L2 cache with 64 byte cachelines. The simulated system was booted using Knoppix 5.1.1, a GNU/Linux bootable live CD image. The OpenOffice session consisted of opening and converting to postscript a 100 page OpenOffice document. The length of the office trace was reduced, so as to simulate only middle 4 million DRAM accesses. The read to write transaction ratio was found to be  $\approx 3:2$ .

Figure 3 shows the results of our simulation methodology validation runs, in which we compare our data for DDR2 and FBDIMM simulations with those published in [8]. In these simulations each DIMM has a peak bandwidth of 5.3GB/s for Northbound (reads) and 2.65GB/s on Southbound (writes)

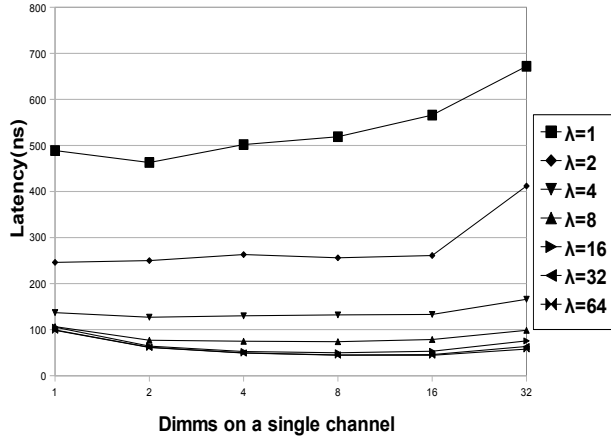


Fig. 6. **OCDIMM Latency characteristics for different wavelengths.** Figure shows average transaction latency on the Y axis for different wavelengths. NB and SB frame cycle is varied with respect to wavelengths (frame cycle  $\neq$  DRAM cycle). Channel capacity is varied along X axis in open page mode, with greedy scheduling, READ-to-WRITE transaction ratio kept at 2:1 and queue depth of 16.

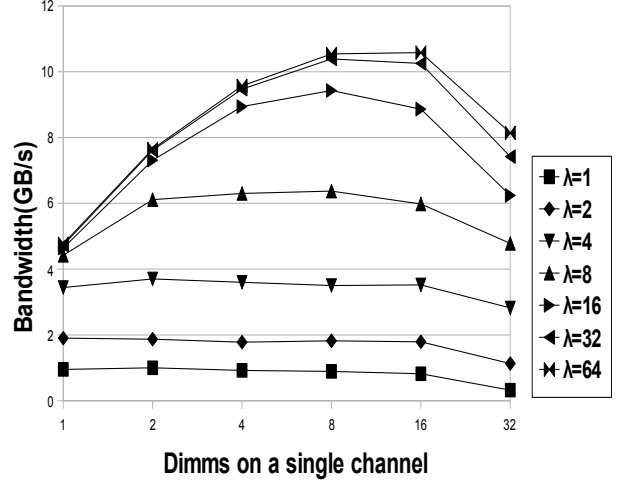


Fig. 7. **OCDIMM Bandwidth characteristics for different wavelengths.** Figure shows single channel bandwidth in GB/s on Y axis for different wavelengths. NB and SB frame cycle is varied with respect to wavelengths (frame cycle  $\neq$  DRAM cycle). Channel capacity is varied along X axis in open page mode, with greedy scheduling, READ-to-WRITE transaction ratio kept at 2:1 and queue depth of 16.

lanes. The figure shows that as the fraction of write transactions increases, the northbound channel on FBDIMM and OCDIMM is not fully utilized, reducing the total bandwidth. In the case of 100% writes OCDIMM achieves a bandwidth of 4GB/s, twice that of FBDIMM, since OCDIMM transfers 16 bytes per frame instead of the eight bytes transferred by FBDIMM.

Figure 5 compares OCDIMM with FBDIMM with respect to latency. We use an *open page* row buffer management policy and a greedy scheduling algorithm along the lines of [9]. As expected, OCDIMM exhibits a clear advantage over FBDIMM since it does not use a multi-hop store and forward scheme. FBDIMM also has a variable latency mode to overcome some of the drawbacks of the store and forward scheme, which comes at the expense of more complexity in the memory controller. Our results indicate that an OCDIMM outperforms both the variable and fixed latency FBDIMM configurations.

The figure also shows that as the number of DIMMs per channel increases, the OCDIMM latency falls, and then climbs slightly. With very few DIMMs per channel, there is not enough concurrency, so the latency is higher. And as the number of DIMMs increases above a certain point, the increase in the propagation delay in the waveguide (note that the propagation delay is around 10.45ps/mm) becomes a dominant factor in the average latency. As a result, there is an optimal configuration around 4 to 8 DIMMs per channel for the specific set of design parameters.

Figure 4 shows that up to 90% improvements in bandwidth can be obtained with OCDIMM when compared to FBDIMM. The benefits are due to both WDM (using multiple wave-

lengths per fiber) and higher data rate (around 10 Gbps).

Figure 6 shows the impact of increasing the number of available wavelengths. Clearly, an increase in wavelengths means more concurrency, hence decreases latency. There is an obvious benefit in going from 2 to 4 wavelengths, but not much in going from 8 to 64. This means that a very large number of wavelengths are not necessary to realize the benefits of optical interconnects.

Figure 7 analyzes the bandwidth against the number of wavelengths used against the number of DIMMs in a channel. It is observed that adding more wavelengths beyond 16 does not help improving the bandwidth due to slower DRAM devices. Beyond 16 DIMMs the transmission delay becomes dominant factor in the latency, reducing the overall channel bandwidth.

#### Advanced OCDIMM Design:

OCDIMM has a single physical channel which supports multiple wavelengths. In order to take advantage of these wavelengths, advanced OCDIMM memory controller dedicates group of wavelengths to particular group of DIMMs, creating optical subchannels within a single fiber. In order to support large capacities, FBDIMM memory controller provides multiple channels and assigns less number of DIMMs per channel. Optical subchannel is based on similar concept but without using electric channel or port. Memory controller needs to route data on correct wavelengths based on the mapping. This feature is useful when increasing memory capacity, DIMMs can be assigned fixed colors(wavelengths) and added to corresponding optical subchannel on-fly. Thus

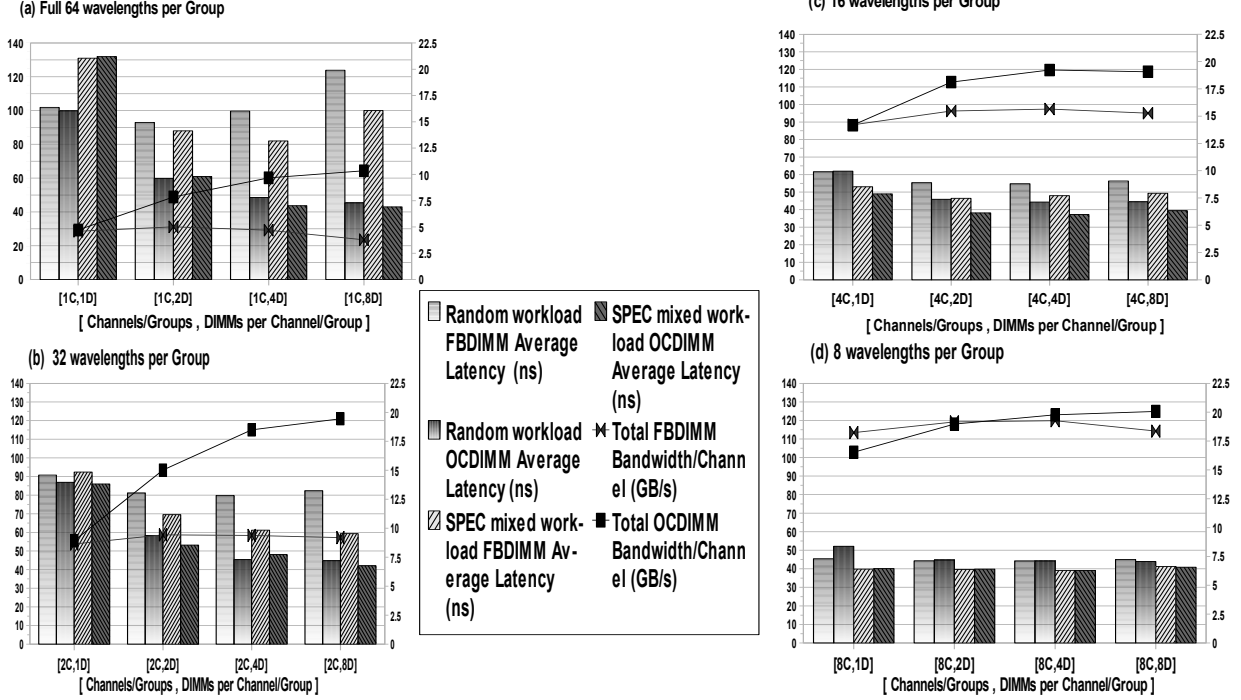


Fig. 8. **Bandwidth and Latency characteristics of advanced OCDIMM and corresponding FBDIMM** Figure compares performance of multi-channel FBDIMM with advanced OCDIMM which divides total spectrum of 64 wavelengths equally amongst the group of DIMMs. A data point on X axis represents number of FBDIMM channels and DIMMs on each channel and its equivalent configuration in OCDIMM representing number of groups and DIMMs per group. All DIMMs in a group share the statically allocated set of wavelengths to that group. Random workload is modeled as standard DDR2-667Mbps, open page mode, with greedy scheduling. Y axis on left side plots latency in nano-seconds and Y axis to the right side plots total system bandwidth in GB/s. For example, In part (c) [4C, 4D] configuration represents 4 FBDIMM channels and 4 DIMMs per channel and it also represents OCDIMM with 64 wavelengths divided into 4 groups of 16 wavelengths each and there are 4 DIMMs sharing these 16 wavelengths within the group.

a DIMM will now have less optical modulator-demodulators depending upon optical subchannel's width, while memory controller has 64 modulators-demodulators.

For example, by dividing 64 wavelengths into 4 groups of 16 wavelengths each, we create 4 optical channels. Using this approach an address belonging to rank 3 (3rd DIMM from controller) is mapped as channel 3, rank 3. An important point to note here is that on optical channel X, ranks 1,2..X-1 are simply absent, so all transactions on X are assigned to the  $X^{th}$  DIMM providing the parallelism desired. Figure 8 presents latency and bandwidth results for the advanced OCDIMM controller model, where we divide 64 wavelengths into channels by grouping 64, 32, 16 and 8 wavelengths each, thus creating 1, 2, 4 and 8 optical channels respectively. The data shows that as you increase the number of DIMMs per channel the latency goes up due to the O/E/O conversions and optical transmission latency. For all the groups shown, the latency seems to level out when having 8 DIMMs or greater per channel. This leveling of latency is apparent because the amount of concurrency is starting to mask the DRAM latency. In Figure 8(a) where all 4 available wavelengths are shared by all the DIMMs on a channel, OCDIMM seems to improve latency by 50% for deeper channels. This is because the frame now transfers 64 bytes of data on northbound and 32 bytes

of data in southbound which enables the entire cacheline to be transferred in one shot. As a result, bandwidth improves but not to a great extent due to the sharing of a single bus. To exploit parallelism in optics, Figure 8(b),(c),(d) show the results when we statically assign set of wavelengths to each group. Thus more than one group can send/receive data at the same time. An equivalent FBDIMM configuration will need multiple physical ports/channels to achieve the same effect. From Figure 8(d), for a large number of channels the advanced OCDIMM design matches multichannel FBDIMM systems, and for smaller number of channels it outperforms FBDIMM Figure 8(a,b,c).

## V. RELATED RESEARCH

Drost et. al. [15] point out the challenges to building a flat-bandwidth memory hierarchy and offer proximity communication as an alternative. The work proposed in this paper addresses the similar challenge (how to provide high bandwidth and high capacity simultaneously). We propose the use of WDM-based optical interconnect to address this challenge. Recently, there has been lot of interest in using optical technology for on-chip networks. Cornell researchers [16] describe a methodology to leverage optical technology to reduce power and latency for on-chip networks. In [6], researchers describe

circuit topologies for efficient on-chip networks based on mirroring resonators. Haas and Vogt introduced FBDIMM technology in [4]. Maryland researchers evaluate the potential of FBDIMM on real workloads in [9]. The Multi-Wavelength Assemblies for Ubiquitous Interconnects (MAUI) has been researching the idea of interfacing fiber to the processor (FTTP) and how the cost will affect the overall penetration of the market [17]. A technology that could escalate the optics architecture in industry is a wafer-scale optical alignment tool [17]. Another interesting research is regarding a High Speed Opto-Electric Memory System (HOLMS) interface which target memory system interaction to processors via an optical connection. This research acknowledges the need for a controller to DIMM type of interface to take advantage of the known optical networking success in the telecommunications industry [18]. Recently there has been significant amount of work in the design of nanoscale modulators and detectors with support for multiple wavelengths [5], [10]–[12], [19], [20] which can be used to realize the optical interconnect structures inherent in the OCDIMM architecture. To the best of our knowledge the work reported here is the first comprehensive proposal and performance evaluation of a practical architecture for incorporating WDM based optical interconnects to overcome the processor-DRAM memory wall and enable *balanced computing* systems.

## VI. CONCLUSIONS AND FUTURE WORK

With decreasing feature sizes we can easily pack a large number of cores on a given chip. The problem shifts to how to feed the cores, especially how to increase the pin bandwidth and reduce the memory latency. In addition, it is important to build *balanced* computing system those scale with the performance, which means capacity and bandwidth both have to increase simultaneously and this is a challenging problem [15]. In this paper, we propose a simple extension to the widely accepted FBDIMM architecture that takes advantage of the multiwavelength optical interconnects. The resultant architecture called OCDIMM is found to improve the capacity, reduce latency and increase bandwidth.

To keep the analysis focused, we purposely avoided discussing the power implications of OCDIMM. Preliminary calculations indicate that despite the O/E/O conversions, optical interconnects will have significant power advantages compared to all electrical signaling methodology like FBDIMM. We are in the process of evaluating the exact power benefits of OCDIMM. Finally, as pointed out by HP researchers [5], WDM based nanophotonics offers an additional dimension of concurrency that can be exploited in innovative ways to further improve the bandwidth and latency of an optical interconnect. We are investigating more advanced memory controller designs based on this idea.

**Acknowledgments:** The authors would like to thank Prof. Bruce Jacob of University of Maryland and his team for their extraordinary textbook on Memory Systems and DRAM simulator (DRAMSim), which helped this project immensely.

## REFERENCES

- [1] U. Nawathe, M. Hassan, K. Yen, A. Kumar, A. Ramachandran, and D. Greenhill, "Implementation of an 8-core, 64-thread, power-efficient sparse server on a chip," *IEEE Journal of Solid-State Circuits*, vol. 43, no. 1, pp. 6–20, Jan. 2008.
- [2] G. Bell, J. Gray, and A. Szalay, "Petascale Computational Systems," *Computer*, vol. 39, no. 1, pp. 110–112, 2006.
- [3] D. Atkins, K. Droegmeier, S. I. Feldman, H. Garcia-Molina, P. Messina, and J. Ostriker, "Revolutionizing science and engineering through cyberinfrastructure - report of blue ribbon panel on cyberinfrastructure," National Science Foundation, Tech. Rep., 2003. [Online]. Available: <http://www.nsf.gov/od/oci/reports/atkins.pdf>
- [4] J. Haas and P. Vogt, "Fully-buffered dimm technology moves enterprise platforms to the next level," *Intel Technology Magazine*, February 2005.
- [5] R. Beausoleil, P. Kuekes, G. Snider, S.-Y. Wang, and R. Williams, "Nanoelectronic and nanophotonic interconnect," *Proceedings of the IEEE*, vol. 96, no. 2, pp. 230–247, Feb. 2008.
- [6] A. Shacham, K. Bergman, and L. Carloni, "On the design of a photonic network-on-chip," in *First International Symposium on Networks-on-Chip*, May 2007, p. 12pp.
- [7] D. Wang, B. Ganesh, N. Tuaycharoen, K. Baynes, A. Jaleel, and B. Jacob, "Dramsim: a memory system simulator," *SIGARCH Comput. Archit. News*, vol. 33, no. 4, pp. 100–107, 2005.
- [8] B. Jacob, S. Ng, and D. Wang, *Memory Systems - Cache, DRAM, disk*. Morgan Kaufman Publishers, 2007.
- [9] B. Ganesh, A. Jaleel, D. Wang, and B. Jacob, "Fully-buffered DIMM memory architectures: Understanding mechanisms, overheads and scaling," in *International Symposium on High Performance Computer Architecture (HPCA)*. Los Alamitos, CA, USA: IEEE Computer Society, Feb 2007, pp. 109–120.
- [10] L. Chen, N. Sherwood-Droz, and M. Lipson, "Compact bandwidth-tunable microring resonators," *Optical Letters*, vol. 32, pp. 3361–3363, 2007.
- [11] B. A. Small, B. G. Lee, K. Bergman, Q. Xu, , and M. Lipson, "Multiple-wavelength integrated photonic networks based on microring resonator devices," *Journal of Optical Networking*, vol. 6, no. 2, pp. 112–120, 2006.
- [12] Q. Xu, B. Schmidt, S. Pradhan, and M. Lipson, "Micrometre-scale silicon electro-optic modulator," *Nature*, no. 435, pp. 325–327, May 2005.
- [13] T. Austin, E. Larson, and D. Ernst, "SimpleScalar: An infrastructure for computer system modeling," *Computer*, vol. 35, no. 2, pp. 59–67, 2002.
- [14] "Bochs ia-32 emulator project <http://bochs.sourceforge.net/>."
- [15] R. Drost, C. Forrest, B. Guenin, R. Ho, A. V. Krishnamoorthy, D. Cohen, J. E. Cunningham, B. Tourancheau, A. Zingher, A. Chow, G. Lauterbach, and I. Sutherland, "Challenges in building a flat-bandwidth memory hierarchy for a large-scale computer with proximity communication," in *HOTI '05: Proceedings of the 13th Symposium on High Performance Interconnects*. Washington, DC, USA: IEEE Computer Society, 2005, pp. 13–22.
- [16] N. Kirman, M. Kirman, R. K. Dokania, J. F. Martinez, A. B. Apsel, M. A. Watkins, and D. H. Albonesi, "Leveraging optical technology in future bus-based chip multiprocessors," in *MICRO 39: Proceedings of the 39th Annual IEEE/ACM International Symposium on Microarchitecture*. Washington, DC, USA: IEEE Computer Society, 2006, pp. 492–503.
- [17] B. Lemoff, M. Ali, G. Panotopoulos, G. Flower, B. Madhavan, A. Levi, and D. Dolfi, "MAUI: enabling fiber-to-the-processor with parallel multiwavelength optical interconnects," *Journal of Lightwave Technology*, vol. 22, no. 9, pp. 2043–2054, Sept. 2004.
- [18] P. Lukowicz, J. Jahns, R. Barbieri, P. Benabes, T. Bierhoff, A. Gauthier, M. Jarczyński, G. Russel, J. Schrage, J. Snowdon, M. Wirz, and G. Troster, "Optoelectronic interconnection technology in the holms system," *IEEE Journal of selected topics in quantum electronics*, 1995.
- [19] W. M. J. Green, M. J. Rooks, L. Sekaric, and Y. A. Vlasov, "Ultra-compact, low RF power, 10 gb/s silicon mach-zehnder modulator," *Optical Express*, vol. 15, no. 25, p. 8, december 2007.
- [20] O. Liboiron-Ladouceur, H. Wang, and K. Bergman, "Transparent, low power optical wdm interface for off-chip interconnects," *Lasers and Electro-Optics Society, 2007. LEOS 2007. The 20th Annual Meeting of the IEEE*, pp. 680–681, 21–25 Oct. 2007.