CACTI 7: New Tools for Interconnect Exploration in Innovative Off-Chip Memories

RAJEEV BALASUBRAMONIAN, University of Utah
ANDREW B. KAHNG, University of California, San Diego
NAVEEN MURALIMANOHAR, Hewlett Packard Labs
ALI SHAFIEE, University of Utah
VAISHNAV SRINIVAS, University of California, San Diego

Historically, server designers have opted for simple memory systems by picking one of a few commoditized DDR memory products. We are already witnessing a major upheaval in the off-chip memory hierarchy, with the introduction of many new memory products—buffer-on-board, LRDIMM, HMC, HBM, and NVMs, to name a few. Given the plethora of choices, it is expected that different vendors will adopt different strategies for their high-capacity memory systems, often deviating from DDR standards and/or integrating new functionality within memory systems. These strategies will likely differ in their choice of interconnect and topology, with a significant fraction of memory energy being dissipated in I/O and data movement. To make the case for memory interconnect specialization, this paper makes three contributions.

First, we design a tool that carefully models I/O power in the memory system, explores the design space, and gives the user the ability to define new types of memory interconnects/topologies. The tool is validated against SPICE models, and is integrated into version 7 of the popular CACTI package. Our analysis with the tool shows that several design parameters have a significant impact on I/O power.

We then use the tool to help craft novel specialized memory system channels. We introduce a new relay-on-board chip that partitions a DDR channel into multiple cascaded channels. We show that this simple change to the channel topology can improve performance by 22% for DDR DRAM and lower cost by up to 65% for DDR DRAM. This new architecture does not require any changes to DIMMs, and it efficiently supports hybrid DRAM/NVM systems.

Finally, as an example of a more disruptive architecture, we design a custom DIMM and parallel bus that moves away from the DDR3/DDR4 standards. To reduce energy and improve performance, the baseline data channel is split into three narrow parallel channels and the on-DIMM interconnects are operated at a lower frequency. In addition, this allows us to design a two-tier error protection strategy that reduces data transfers on the interconnect. This architecture yields a performance improvement of 18% and a memory power reduction of 23%.

The cascaded channel and narrow channel architectures serve as case studies for the new tool and show the potential for benefit from re-organizing basic memory interconnects.

Categories and Subject Descriptors: B.3.1 [**Memory Structures**]: Dynamic Memory (DRAM)

General Terms: Design, Algorithms, Performance

Additional Key Words and Phrases: Memory, DRAM, NVM, interconnects, tools

This work was supported in part by NSF grants CNS-1302663, CNS-1423583, and CCF-1564302, and the Center for Design-Enabled Nanofabrication (C-DEN).
Contact Author's address: R. Balasubramonian, 50 S. Central Campus Drive, Rm. 3190, Salt Lake City, UT 84112.

DOI: http://dx.doi.org/10.1145/3085572

ACM Reference Format:
Rajeev Balasubramonian, Andrew B. Kahng, Naveen Muralimanohar, Ali Shafiee, and Vaishnav Srinivas. 2017. CACTI 7: New tools for interconnect exploration in innovative off-chip memories. ACM Trans. Archit. Code Optim. 14, 2, Article 14 (June 2017), 25 pages.
DOI: http://dx.doi.org/10.1145/3085572

1. INTRODUCTION

Memory products have long been standardized and commoditized. Most server memory architectures have remained "traditional"—DDR channels emerge from a processor socket and support a small number of plug-in DDR DIMMs. The past few years have already seen the first signs of upheaval in the memory system. Several new memory products have emerged recently or will soon be available, including buffer-on-board (BoB) [Intel 2014], LRDIMM [Micro 2015a], HyperCloud DIMMS (HC-DIMMs) [Netlist 2012], NVDIMMs [SanDisk 2014], HMC [Pawlowski 2011], NVMs [Burr et al. 2010; Strukov et al. 2008], and memory blades [Lim et al. 2009].

Server vendors are not viewing the memory system as a commodity any longer—the memory system is now viewed as a differentiating feature, especially for customers who deal with big data workloads. There are many example platforms and workloads that rely on processing of large in-memory datasets, such as SAP HANA [SAP 2013], SAS in-memory analytics [SAS 2013], RAMCloud [Ousterhout et al. 2009], SPARK [Zaharia et al. 2010], and memory blades [Lim et al. 2009]. Memory system design choices dominate the performance, power, and cost metrics for such systems. For example, memory accounts for 50% of the power and 40% of the cost of 6TB HP servers [HP 2015].

Vendors are therefore considering new approaches to design memory systems that best serve the needs of their customers. The apparent absence of a future DDR5 standard is an indicator of industry's need for specialization within the memory system. Further, future systems must efficiently support a combination of DRAM and NVM modules. We make the hypothesis that revisiting the design of the basic DDR channel and introducing new memory interconnect topologies can yield large benefits.

To test the hypothesis, we first create a tool to precisely model interconnect power (referred to as I/O power) in the memory system. Instead of using the Micron power calculator's generic I/O model (as is done in most research evaluations today), we use SPICE simulations to model the effects of several parameters on interconnect power. These models build on the ones in CACTI-IO [Jouppi et al. 2015], and a design space exploration has been added to identify the best design points. The overall tool has been integrated into version 7 of the popular CACTI package.[1] Our analysis with this tool shows that I/O power is indeed a significant contributor of power in large memory systems and is greatly affected by many parameters (e.g., the number of DIMMs per channel (DPCs)). The tool uses a simple API that allows users to define nontraditional interconnect topologies for the memory system.

Next, to make the case that memory interconnect specialization can yield significant benefit, and to test the value of our tool, we introduce and evaluate two novel interconnect architectures. We use the tool to carry out a simple design space exploration to identify the best design points for a given memory capacity requirement. Our analysis shows that higher bandwidth and lower cost can be achieved if the processor socket can somehow support a larger number of memory channels. This insight paves the way for the following two proposals:

(1) A cascaded channel architecture that is DDR compliant and a good fit for a DRAM/NVM hierarchy. By partitioning a DDR channel into multiple cascaded

[1]CACTI 7 can be downloaded from Hewlett Packard Labs (https://www.labs.hpe.com/downloads) or mirror sites at UCSD (http://vlsicad.ucsd.edu/CACTI/) or Utah (http://arch.cs.utah.edu/cacti/).

segments, it is able to support high memory capacity and bandwidth. It is also able to support a given memory capacity with a large number of smaller-capacity DIMMs, thus lowering the overall cost for memory modules. DRAM DIMMs can be placed on high-frequency channels that emerge from the processor socket, whereas NVM DIMMs can be placed on lower-frequency channels that are further from the processor. The cascaded channels are enabled by the introduction of a new relay-on-board chip and a simple memory controller scheduler. We evaluate the new topologies in the context of a memory cartridge and show how our tool can be adapted to quantify the impact on performance, cost, and power.

(2) A narrow-channel architecture that partitions a wide channel into parallel, independent, narrow, and higher-frequency channels. The corresponding channel and DIMMs are not DDR compliant. This opens up the possibility of defining new custom DIMM architectures. We therefore add new power-efficient features to the DIMM. Since a DIMM has lower external bandwidth, it can lower power by operating on-DIMM interconnects at lower frequencies. We also modify the error protection strategy to reduce data transfer overheads. Even though we consider DIMMs and channels that are not DDR compatible, we assume that memory chips are unmodified and are always DDR compliant. We again leverage our tool to demonstrate the benefits in memory bandwidth, cost, and power with this approach.

These new architectures thus make the case that rethinking basic memory interconnect topologies can yield a rich space of very efficient memory architectures. Whereas earlier versions of CACTI [Muralimanohar et al. 2007] have focused on cache hierarchies, the new version adds a new capability—the modeling of off-chip memory interconnects.

2. TOOL CREATION

2.1. Motivation

Emerging and future memory systems will use a combination of serial and parallel buses to connect a large number of memory modules to the processor and support high memory capacities. The memory modules usually adopt a DIMM form factor, and they can be designed without on-DIMM buffers (an unbuffered DIMM or UDIMM), with on-DIMM buffers for command/address (a registered DIMM or RDIMM), or with on-DIMM buffers for all signals (a load-reduced DIMM or LRDIMM). A large number of memory organizations are therefore possible, each with varying memory capacity, bandwidth, power, performance, and cost.

With each new memory generation, the energy per bit transfer is reduced, but at the same time, memory system bandwidth is also doubled. DDR4 DIMM designers are now targeting 3,200 MT/s for multirank DIMMs [Wikipedia 2014]. Given this increase in bandwidth, and given the increase in memory capacity, memory power is increasing. I/O power is a significant contributor to overall memory power, especially when many ranks share a channel or when the channel operates at high frequencies.

Unfortunately, I/O power is overlooked in many research evaluations. For example, the Micron power calculator [Micron 2015b], the most popular memory power model, considers I/O power of a single dual-rank UDIMM for all cases. For instance, the Micron power calculator's output is oblivious to the DIMM and channel configuration.

The Micron power calculator reports that an 800MHz channel operating at 80% utilization with a single dual-rank UDIMM, with a read/write ratio of 2 and a row buffer hit rate of 50%, dissipates 5.6W of power, with 37% of that power being dissipated in I/O components (ODT, drivers). Since I/O power is such a significant contributor, we create and integrate several interconnect power models in a memory architecture exploration tool. As we show later, the I/O power strongly depends on

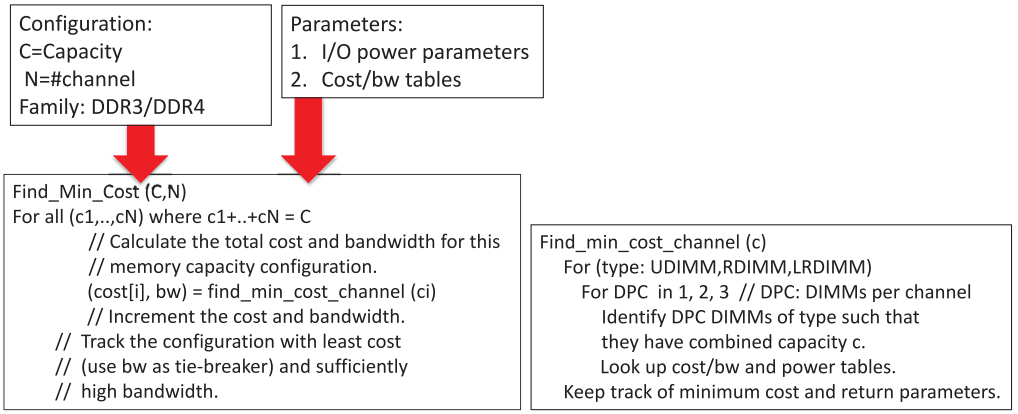


Fig. 1. Basic flowchart for the tool's design space exploration for traditional DDR topologies.

technology (DDR3/DDR4), DIMM type (RDIMM, UDIMM, LRDIMM), the number of DPCs, the channel frequency, and so forth.

2.2. Tool Inputs and Outputs

An out-of-the-box version of our tool can receive a small number of high-level inputs from the user, explore the design space, and report the memory configurations that yield the best user-defined metrics. This may be used by researchers/practitioners to determine the properties of a baseline memory system. For novel nonstandard memory architectures, the tool provides a simple API to represent the key interconnection features of the new architecture. A researcher can use this API to either define the properties of a custom memory network or augment the tool's design space exploration to consider a wider range of network topologies. This article shows examples of all of these use cases. The tool receives the following parameters as inputs:

(1) *Memory capacity in multiples of 4GB and if ECC support is provided*.
(2) *Processor restrictions*: The following restrictions define how the memory is connected to the processor: (i) the number of memory channels, (ii) the number of wires in each memory channel, (iii) the memory type (i.e., DDR3 or DDR4), and (iv) if the processor connects to a BoB and the number of DDR channels connected to the BoB.
(3) *Minimum memory bandwidth requirement*.
(4) *Memory access pattern*: The user can either specify a known memory traffic rate, row buffer hit rate, and read/write ratio, or it can allow the tool to iterate over a range of these values and report an average.
(5) *Goodness metric*: This determines how the tool identifies the best memory organizations. The user can prioritize either power or cost or bandwidth, or provide his or her own metric that combines these metrics.

The new tool sweeps through all possible configurations that meet the input constraints and identifies those that maximize the goodness metric while also providing a power and cost breakdown.

2.3. Tool Modeling

2.3.1. High-Level Loops. The overall flowchart for the tool, written in C++, is described in Figure 1. Elements of this tool can be easily integrated in other architectural simulators so that workload characteristics can be used to estimate memory power.

Table I. Pricing Data for Different DIMM Types and Capacities

DDR3					
DIMM	4GB	8GB	16GB	32GB	64GB
UDIMM	$40.4	$76.1	—	—	—
RDIMM	$42.2	$ 64.2	$122.6	$304.3	—
LRDIMM	—	—	$211.3	$287.5	$,079.5
DDR4					
DIMM	4GB	8GB	16GB	32GB	64GB
UDIMM	$26	$46.00	—	—	—
RDIMM	$33	$ 60.45	$126	$	—
LRDIMM	—	—	$279	$331.3	$1,474.7

Given the inputs specified in the previous section, the tool first identifies each required on-board channel (either parallel or serial). For the parallel DDR channels, it first runs through a loop that enumerates every possible allocation of the memory capacity across the DDR channels. For each channel, we then enumerate every combination of available DIMMs that can achieve that memory capacity, at both high-performance and low-power voltages. Based on these parameters, the channel frequency is determined (we provide more details on these steps shortly). We confirm that the total bandwidth of all specified channels is above the minimum specified requirement. For each valid configuration, we then estimate the cost to build a server, which is based on a look-up table that enumerates pricing for each type of DIMM. The power estimation is more nontrivial. We first use the Micron power calculator to estimate the non-I/O power consumed within the memory chips. For power consumed within each interconnect (on-DIMM, on-board, parallel, and serial), we develop our own methodology, which is described in Section 2.4. After estimating power and cost, we keep track of the memory configuration that maximizes the user-provided goodness metric (some combination of bandwidth, power, and cost).

2.3.2. Cost Model. To compute the cost to build a server, we need empirical pricing data for various DIMM types. We obtained DIMM prices from www.newegg.com and averaged the price for the first 10 products produced by our search. These prices are maintained by the tool in a look-up table. Some of this data is reproduced in Table I and shows prices for a variety of DIMM types and capacities for DDR3 and DDR4.

We must point out the obvious caveats in any pricing data. They are only meant to serve as guidelines because we cannot capture the many considerations that might determine final price (e.g., the pricing data might include some discount that may not be available a month later). That said, we feel the need to provide this pricing data because memory pricing considerations do drive the configurations of many server designs. Skeptical users can either use our pricing data to validate their own analytical pricing model or can remove cost from the goodness metric altogether. The data in Table I is fairly representative of the memory market, where the highest-capacity DIMMs see a sharp rise in price per bit. To keep the tool up-to-date, we plan to periodically release updated pricing data.

2.3.3. Bandwidth Model. Memory channel frequency depends on the DIMM voltage, DIMM type, and the number of DPCs. This dependency is obtained from memory guideline documents of various server vendors. We obtained our specifications for frequency of DDR3 and DDR4 channels from Dell PowerEdge servers (12th generation) [ESG Memory Engineering 2012] and Super Micro's X10 series [Supermicro 2015], respectively. Table II enumerates this frequency data.

Table II. Channel Frequencies for Various DIMM Configurations

DDR3						
DIMM	1 DPC (MHz)		2 DPC (MHz)		3 DPC (MHz)	
Type Ranking	1.35V	1.5V	1.35V	1.5V	1.35V	1.5V
RDIMM-DR	—	800	—	800	—	533
RDIMM-DR	667	667	667	667	—	533
UDIMM-DR	533	667	533	667	—	—
LRDIMM-QR	400	667	400	400	—	—
LRDIMM-QR	667	667	667	667	533	533

DDR4			
DIMM	1 DPC (MHz)	2 DPC (MHz)	3 DPC (MHz)
Type Ranking	1.2V	1.2V	1.2V
RDIMM-DR	1,066	933	800
RDIMM-QR	933	800	—
LRDIMM-QR	1,066	1,066	800

2.4. Power Modeling

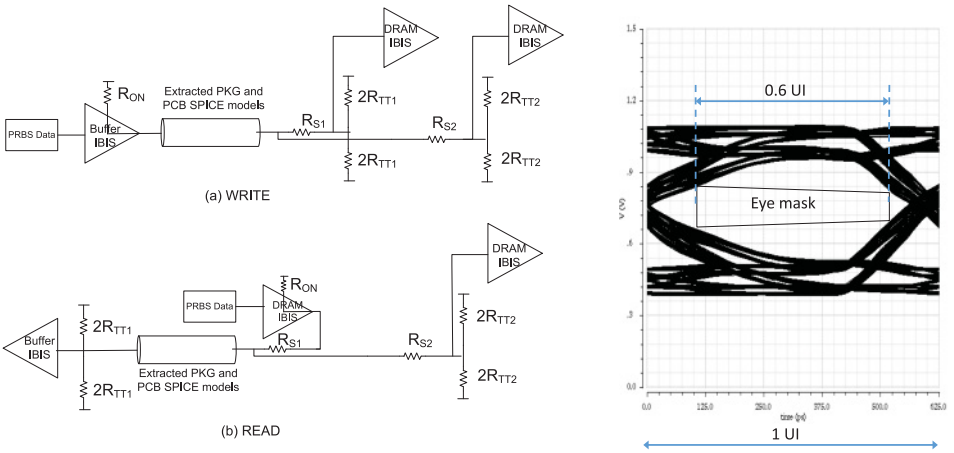
We use CACTI-IO [Jouppi et al. 2015] as a starting point for our I/O models and introduce several extensions to enable a comprehensive exploration of memory systems:

(1) DDR4 and SERDES models were added to the models already present in CACTI-IO (DDR3, WideIO, LPDDR3).
(2) The DDR3 and DDR4 models are provided for three different types of segments:
—On-DIMM (i.e., from the DIMM buffer to the DRAM chip)
—Main-board, for the processor or BoB to the DIMM buffer
—Both, for the host or BoB to the DRAM for an unbuffered signal.
(3) Support has been added to compute termination power as a function of DDR3/DDR4 channel frequencies and channel loads (number of DIMMs/buffers on channel).

Each of the preceding models was constructed with detailed HSPICE simulations, following similar methodology as for CACTI-IO [Jouppi et al. 2015]. To construct the HSPICE models, we obtained IBIS [2014] models for various components, we assumed an 8-bit datapath on a PCB metal layer to account for crosstalk, and we used a simple RLC model to approximate the DIMM connector [AMP 2014]. For example, Figure 2(i) shows the SPICE testbench used for WRITE and READ simulations for a DDR3 on-DIMM case.

For each interconnect end point, a termination resistance is required to damp signal reflections. The value of the termination resistance determines interconnect power and signal quality. We therefore perform HSPICE time-domain analyses to plot eye diagrams for the data bus for each candidate memory configuration—different frequencies, different DPCs, and different topologies (on-DIMM, on-board, with/without a buffer). For each configuration, we sweep through different termination resistance values until the eye lines up to 0.6 UI quality, as shown in Figure 2(ii) (UI is short for unit interval, which is the ideal full eye opening).

With the preceeding methodology, our tool is equipped with appropriate termination resistance values for a variety of parallel bus configurations. These termination resistance values are used by previously validated I/O power equations in CACTI-IO to compute power for DDR3 interconnects. The DDR4 model has one significant change from that for DDR3: the termination resistors are referenced to VDDQ to allow for lower idle termination power. Equations (1) and (2) describe the WRITE and READ



(i) DDR3 SPICE testbench. (ii) Eye diagram.

Fig. 2. (i) DDR3 SPICE testbench. The testbench includes IBIS models for the buffer and DRAM, extracted SPICE models for the package and PCB and a PRBS input stimulus pattern. We sweep the termination resistances R_{TT1} and R_{TT2} to identify a configuration with high signal integrity. (ii) DDR3 eye diagram.

termination power for DDR4 DQ:

$$P_{DQ\_Term\_Write} = 0.5 \cdot V_{dd}^2 \cdot \left(\frac{1}{R_{ON} + R_{||\_Write}} \right), \tag{1}$$

$$P_{DQ\_Term\_Read} = 0.5 \cdot V_{dd}^2 \cdot \left(\frac{1}{R_{ON} + R_{S1} + R_{||\_Read}} \right), \tag{2}$$

where V_{dd} is the supply voltage, and R_{ON}, R_{S1}, $R_{||\_Write} = (R_{TT1} + R_{S1})||(R_{TT2} + R_{S2})$, and $R_{||\_Read} = R_{TT1}||(R_{TT2} + R_{S2})$ are DDR4 resistances similar to those shown in the DDR3 testbench in Figure 2(i). In addition, we allow technology scaling, borrowing the same methodology as CACTI-IO [Jouppi et al. 2015].

The timing budgets in CACTI-IO [Jouppi et al. 2015] are based off bit error rate (BER) models [Keller 2012]; when termination resistance values are extracted for a given topology, frequency, and voltage, the eye mask feeds into the timing budget that is based on a specified BER. Our tool does not allow the user to modify the interconnect design while tolerating lower/higher BER.

For serial buses, the SERDES I/O power is modeled as a look-up table based on the length of the interconnect and the frequency. Because of the high variation in SERDES link architectures, it is difficult to capture them with general analytical models. Figure 3 shows the typical components of a SERDES link, including the transmitter, termination, and the receiver. The power numbers for the various components are derived from a survey [Lee et al. 2009a; Poulton et al. 2009; OMahony et al. 2010; Palmer et al. 2008; Pawlowski 2011; Intel 2014] and divided into three types based on the length of the interconnect: short ($<$ 2inches), mid ($<$ 10inches), and long ($>$ 10inches). We further provide power numbers for each type at three different frequencies: slow ($<$ 1 Gbps), medium ($<$ 5Gbps), and fast (up to 10Gbps). We provide these parameters for static, dynamic, and clock power. This allows for scaling with bandwidth, and also to investigate amortization of clock power over different numbers of data lanes, as shown in Equation (3):

$$P_{component} = P_{clock} + N_{lanes} \cdot (P_{static} + BW \cdot P_{dynamic}), \tag{3}$$

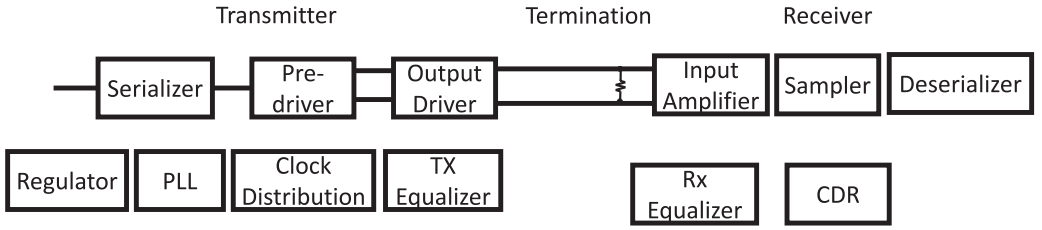


Fig. 3. Typical components in a SERDES link.

where $P_{component}$ is the power of the component of the SERDES link shown in Figure 3, P_{clock} is the clock power of that component, P_{static} is its static power, $P_{dynamic}$ is its dynamic energy/bit, BW is the bandwidth of the whole link, and N_{lanes} is the number of data lanes.

To summarize this section, we have created look-up tables in our tool that can help capture the I/O power for a very large design space of memory interconnects. Some of these tables capture termination resistances that are obtained with detailed HSPICE simulations (DDR3 and DDR4)—analytical equations are then used to compute final I/O power. Other look-up tables (SERDES) directly capture I/O power from literature surveys.

2.5. An API to Define New Interconnects

The earlier discussion describes how our tool evaluates the design space of commercially available commodity memory products. As new products emerge, the data in the preceding tables and the power calculator equations can be augmented to widen the design sweep. In addition to new memory products, we expect that researchers will likely experiment with new memory network topologies and new memory hierarchies that combine multiple different memory devices. To facilitate such new models, we introduce the following API to define new links in the memory network. The new ideas explored in the second half of the article leverage these APIs for their evaluations.

For each interconnect, the API requires us to define the type of link and the width of that link. Our tool categorizes links into three types: serial link, parallel DDR (double data rate) link, and parallel SDR (single data rate) link. Serial links are used for high-speed point-to-point connections, such as the SERDES links used by the HMC or by the FBDIMM [Vogt 2004]. DDR links are used for the data bus in multidrop memory channels. SDR links are used for the command/address bus in multidrop memory channels.

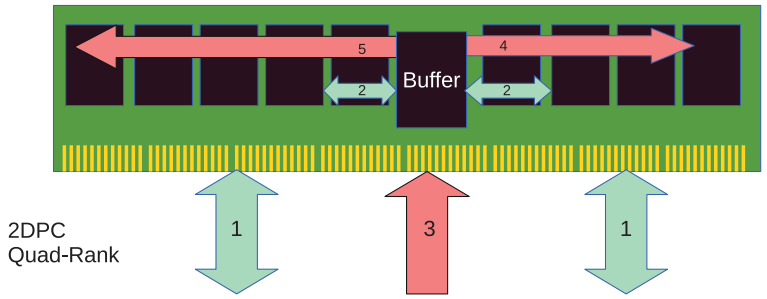
Each link type has a few key knobs as input parameters, which allow the user to define the configuration and physical location of the segments of the interconnect. These knobs include the following and are summarized in Table III.[2]

Figure 4 provides a detailed example of how the preceding API is used to describe an LRDIMM. A similar process would be used to define a new DIMM that (say) contains customized links between an accelerator on the DIMM and specific memory chips, or a new memory topology where a BoB is connected to an HMC and a DDR channel.

[2]*Range* refers to the length of the interconnect (as described previously). *Frequency* refers to the clock frequency on the bus. *Num_wire* refers to the number of wires or the bus width. *Num_drop* refers to the number of other devices (typically DIMMs and/or buffers) in a multidrop interconnect. *Type* refers to the memory technology (DDR3, DDR4, WideIO, LPDDR3, etc.). *Connection* refers to the location of the segment—that is, on-DIMM or main-board or both (as in the unbuffered DIMM described previously).

Table III. Parameters Used to Define Each Interconnect Type

Type	Parameters
SERDES (Serial)	range, frequency, num_wire Range: short or long
DDR	range, frequency, num_wire, num_drop, type, connection Type: DDR3, DDR4, LPDDR2 Connection: on_dimm, on_main_board
SDR	range, frequency, num_wire, num_drop, type, connection Type: DDR3, DDR4, LPDDR2 Connection: on_dimm, on_main_board



1. DDR(long_range, frequency, 2(dimms), DDR3, 72, on-main-bus)
2. DDR(short_range, frequency, 4(ranks), DDR3, 72, on-dimm)
3. SDR(long_range, frequency, 2(dimms), DDR3, 23, on-main-bus)
4. SDR(short_range, frequency, 4(dies), DDR3, 23, on-dimm)
5. SDR(short_range, frequency, 5(dies), DDR3, 23, on-dimm)

Fig. 4. Link description for a DDR3 LRDIMM.

2.6. Validation

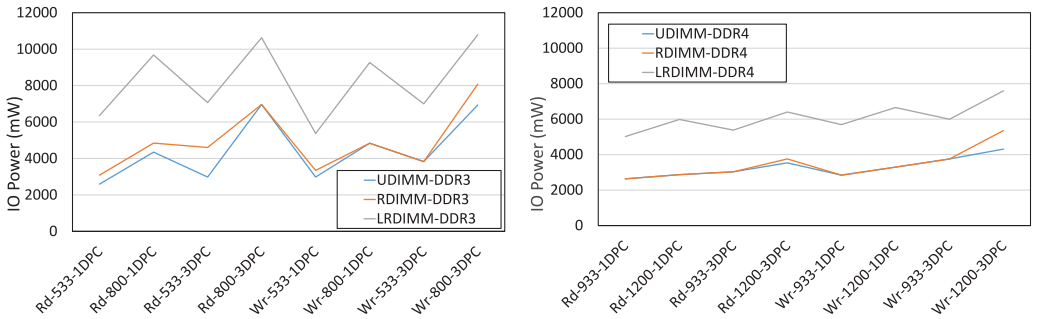
The key to precise I/O power calculations is the estimation of termination resistances that yield sufficiently open eye diagrams; as described earlier, that step is being performed with detailed SPICE simulations. These resistance values are then fed into our tool's equations to obtain the I/O power. The DDR3 equations have already been validated by CACTI-IO [Jouppi et al. 2015]. Here we validate the DDR4 equations against SPICE DC simulations. As shown in Table IV, we compare the termination powers of one DQ lane driving low (driving a high result in close to 0 termination power). We assume $V_{dd} = 1.2$, $R_{on} = 34$, $R_s = 10$ for DDR4 reads. This comparison is performed for a range of R_{TT1} and R_{TT2} termination values. Table IV shows that the analytical equations used by our tool for the DDR4 termination power (as shown in Equations (1) and (2)) are aligned with SPICE simulations. It should be noted that our power models further include dynamic switching power (of the loads and the interconnect) and PHY power similar to the DDR3 case, as described and validated in Jouppi et al. [2015].

2.7. Contributions

We have thus created a tool and API, integrated into CACTI 7, that makes it easy to carry out design space exploration for modern and emerging memory topologies, while correctly modeling I/O power as a function of various parameters (not done by the Micron power calculator), and correctly considering DDR4, SERDES, cost, and bandwidth (not done by CACTI-IO).

Table IV. Validation Data

Rtt1 (Ω)	Rtt2 (Ω)	CACTI 7 Termination Power (mW)	SPICE Termination Power (mW)
120	120	13.53	13.6
120	60	16.32	16.4
120	40	18.16	18.1
60	120	16.93	17.1
60	60	18.87	19.1
60	40	20.20	20.1
40	120	19.30	19.4
40	60	20.73	20.8
40	40	21.74	21.5



(a) DDR3 configurations. (b) DDR4 configurations.

Fig. 5. Memory I/O power (in megawatts) for several DDR3/DDR4 design points that vary DIMM type, read/write intensity, frequency, and DPCs.

3. TOOL ANALYSIS

3.1. Contribution of Memory I/O Power

The previous section described (i) how CACTI-IO power models have been augmented to handle a much larger interconnect design space and (ii) how a large memory organization design space can be evaluated in terms of power, cost, and bandwidth.

To test our initial hypothesis that design choices can significantly impact I/O power and overall memory power, we use our tool to evaluate several memory configurations that differ in terms of DIMM types, read/write intensity, channel frequency, and DPCs. The power values are reported in Figure 5 for DDR3 and DDR4.

First, this paragraph compares I/O power to the DRAM chip power without any I/O components (obtained from the Micron power calculator). The configurations shown in Figure 5 dissipate I/O power between 2.6W and 10.8W for fully utilized channels. Even if we assume a low row buffer hit rate of 10%, eight ranks sharing the channel would collectively dissipate only 10.3W in non-I/O DRAM power. Similarly, two-rank and one-rank configurations would dissipate only 5.6W and 4.8W, respectively. This highlights the significant role of I/O power in accessing the memory system.

In looking at each graph in Figure 5, we see that RDIMMs and UDIMMs dissipate similar amounts of power, but LRDIMMs dissipate nearly 2× more power. We also observe that at least in DDR4, there is a clear trend where write-intensive traffic (the right half of the figures) consumes more power than read-intensive traffic (the left half of the figures). Not surprisingly, power increases with channel frequency, and power increases as more DIMMs are added to the channel. To a large extent, I/O power is

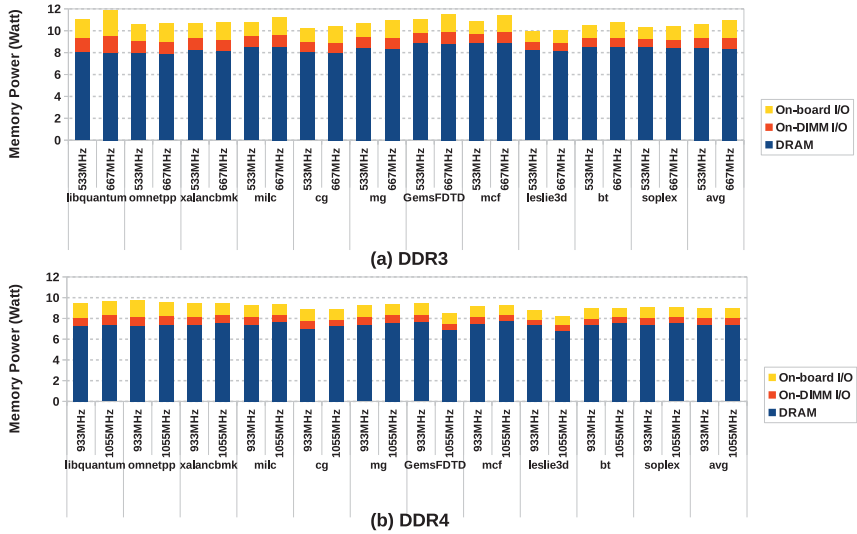


Fig. 6. Breakdown of memory power (in megawatts) for DDR3/DDR4 design points for each benchmark. We assume three LRDIMMs per channel.

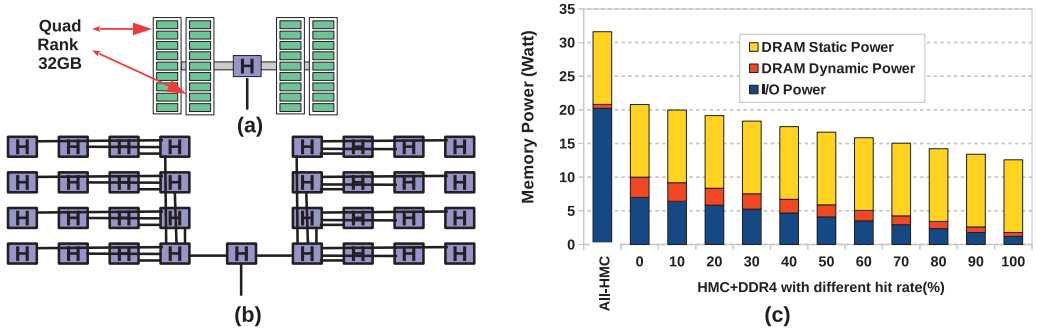


Fig. 7. (a) A memory organization with one HMC "cache" that is backed by two DDR4 channels and four quad-rank LRDIMMs. (b) An HMC-only organization with one HMC "cache" backed by 32 other 4GB HMCs. (c) Memory power for the two organizations, as a function of the hit rate in the HMC "cache."

influenced by the following five factors in decreasing order of importance: DIMM type, technology, frequency, read/write intensity, and DPCs.

To better understand the differences between DDR3 and DDR4, we show the power breakdown for various benchmarks and frequencies in Figure 6. These graphs assume three LRDIMMs per channel. DDR4 consumes less power within DRAM and within interconnects, primarily because of its lower voltage and its lower idle termination power. This is true despite DDR4 operating at a significantly higher frequency than DDR3. But as a percentage of total memory power, the average contribution of I/O increases from 21% in DDR3 to 24% in DDR4.

Next, to show the impact of I/O power in future large memory systems, including those that incorporate new memory devices such as Micron's Hybrid Memory Cube (HMC) [Jeddeloh and Keeth 2012], we evaluate the memory organizations described in Figure 7. Both organizations connect the processor to a low-latency HMC device that caches the most popular pages. The remaining pages are either scattered uniformly

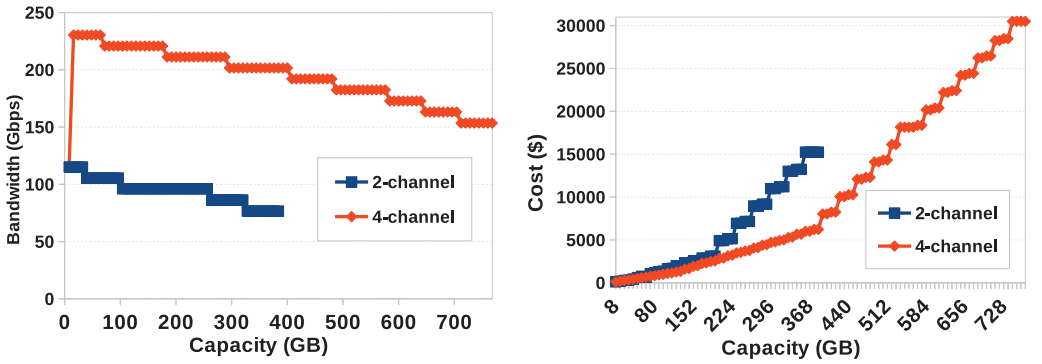


Fig. 8. Identifying the best bandwidth (left) and cost (right) design points for each memory capacity requirement.

across four LRDIMMs on two DDR4 channels (a) or across an iso-capacity network of 32 HMCs (b). HMC memory access energy is based on the empirical data of Jeddeloh and Keeth [2012]. We assume the same amount of background DRAM power per bit for both cases. The graph shows memory power dissipation as the hit rate in the HMC cache is varied. Given the many HMC SERDES links that are always active, we see that the HMC-only organization consumes significantly more power. This analysis showcases (i) the importance of I/O power and (ii) the relevance of DDR channels in future memory ecosystems.

3.2. Motivation for Cascaded Channels and Narrow Channels

Next, to show the power of our tool's design space explorations, and to motivate the ideas in the second half of the article, we identify the best cost and bandwidth design points for a range of memory capacities. This is shown in Figure 8. We consider processor models that support two channels and those that support four channels. The data shows that for a given memory capacity requirement, the use of four channels can improve overall bandwidth by more than the expected 2×. This is because the use of more channels helps spread the DIMMs, thus lowering per-channel load and boosting per-channel bandwidth. Similarly, the use of more channels can also yield DIMM configurations that cost a lot less. This is because with more channels, we can populate each channel with different DIMM types and frequency, thus providing a richer design space and avoiding the need for expensive high-capacity DIMMs.

With these observations in mind, we set out to create new memory channel architectures that can grow the number of memory channels without growing the pin count on the processor. Our first approach creates a daisy chain of channels with a lightweight buffer chip. Our second approach partitions a DDR channel into narrow subchannels. By decoupling memory capacity and processor pin count, we can further reduce cost by enabling lower-end processors and motherboards for certain platforms/workloads. In Figure 9, we show that Intel Xeon processors with fewer channels are available for a much lower price range.

4. THE CASCADED CHANNEL ARCHITECTURE

The cascaded channel architecture can simultaneously improve bandwidth and cost while having a minimal impact on server volume or complexity or power. It can also be leveraged to implement a hybrid memory system, thus serving as an important enabling technology for future systems that may integrate DRAM and NVM.

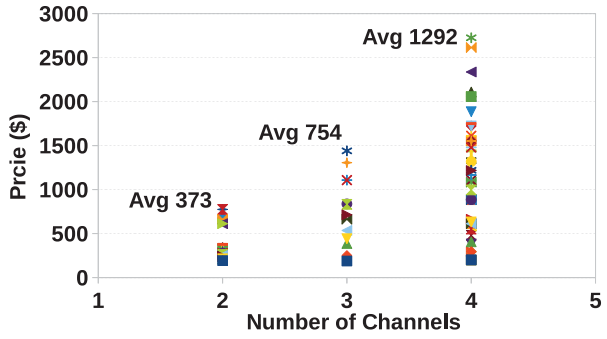


Fig. 9. Prices for Intel Xeon (E3 and E5) processors as a function of channel count [Intel 2016].

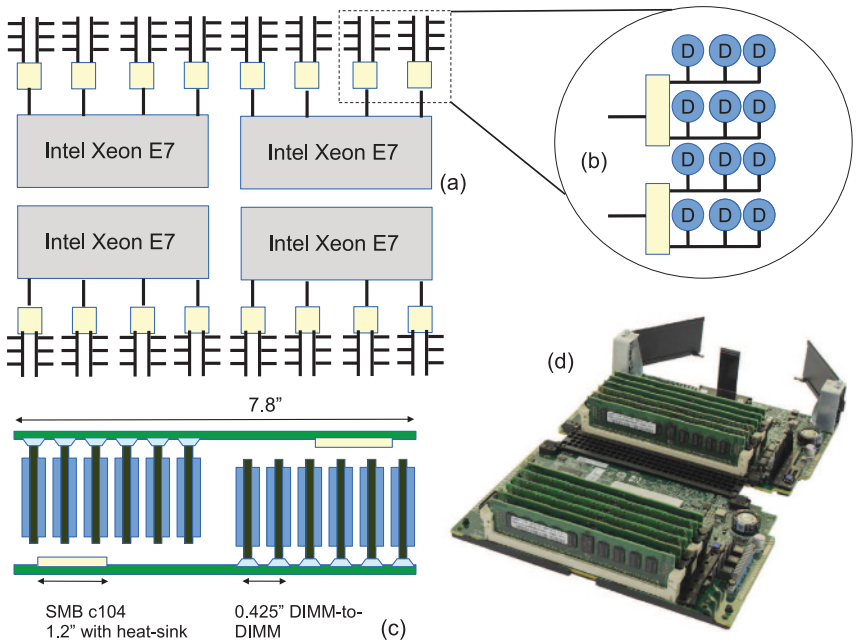


Fig. 10. Organization of the HP ProLiant DL580 Gen8 server (a). The memory cartridge is represented in (b), (c), and (d).

The key tool insights that motivate this proposal are the following: (i) low load on a parallel bus leads to higher frequency, and (ii) cost is lowered by using many low-capacity DIMMs. We therefore partition a single DDR channel into multiple shorter DDR channels with a simple on-board relay chip.

4.1. Proposed Design

4.1.1. Baseline Memory Cartridge. We use a state-of-the-art memory cartridge as the evaluation platform and baseline in this work. Both HP and Dell have servers that can accommodate eight 12-DIMM memory cartridges to yield servers with 6TB memory capacity [Myslewski 2014; HP 2014; Dell 2014].

Figure 10(a) shows the overall configuration of an HP ProLiant DL580 Gen8 Server [HP 2014]. Four processor sockets in the 4U server connect to eight memory cartridges. Each processor socket has four memory links that connect to four BoB chips (Intel C104

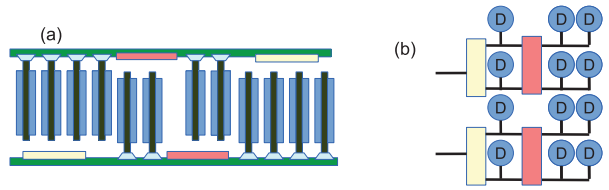


Fig. 11. The cascaded channel architecture. RoB chips are used to partition each channel.

Scalable Memory Buffers [Intel 2014]). A memory cartridge is composed of two BoBs and their four DDR memory channels. Each memory channel can support up to three 64GB LRDIMMs at a channel frequency of 533MHz. Figure 10(b) shows a logical view of a single memory cartridge; Figure 10(d) shows an actual image of an open cartridge populated with 12 DIMMs. The two PCBs close up to form a dense enclosure, as shown in Figure 10(c) (a side view). Each BoB chip is on a separate PCB, and the two emerging DDR channels are connected to the six interleaved DIMMs. We refer to the six DIMMs on the bottom PCB as *stalagmites* and the six DIMMs on the top PCB as *stalactites*. The DIMMs are arranged so that stalactites lie above a BoB and do not interfere with stalagmites (and vice versa).

4.1.2. Cascaded Channel Architecture. Our proposal uses a new relay-on-board (RoB) chip to create a daisy chain of DDR channels. As shown in Figure 11, the DDR channel emerging from the (yellow) BoB chip terminates in a (pink) RoB chip. Another DDR channel emerges from the other end of the RoB chip. What distinguishes this RoB chip from all other known BoB chips is that it has traditional DDR channels on either end that are populated with DIMMs.

From the memory channel's perspective, the RoB chip appears similar to an LRDIMM buffer chip. For instance, in terms of handling ODT and rank-to-rank switching delays, the RoB chip is simply handled as another rank. The RoB chip has a minimal amount of logic to receive data and (if necessary) drive the same signal on the next channel in the daisy chain (with appropriate retiming and skew); the area of this chip is largely determined by its pin count. If we assume that the channels connected to the RoB operate at the same frequency, no data buffering is required on the RoB, and the signals on one channel are propagated to the next channel with a delay of one cycle. At boot-up time, the memory controller must go through a few additional steps for system initialization.

On a memory read or write, depending on the address, the request is serviced by DIMMs on the first channel or by DIMMs on the second cascaded channel. In the baseline (Figure 10), a single channel supports three LRDIMMs at a frequency of 533MHz. In the cascaded channel design (Figure 11), the first channel segment supports the memory controller, a single LRDIMM, and a RoB chip. Such a channel is equivalent to a DDR channel populated with two LRDIMMs. Based on server datasheets [ESG Memory Engineering 2012], such a channel can safely operate at a frequency of 667MHz. The second channel segment is similar—it supports the RoB chip (equivalent to a memory controller) and two LRDIMMs. Therefore, it too operates at a frequency of 667MHz. The introduction of a RoB chip in a memory channel is similar to the introduction of a latch in the middle of a CMOS circuit to create a pipeline. The primary benefit is a boost in frequency and parallelism.

Figure 11(a) also shows how the cartridge can be redesigned in a volume-neutral way. The LRDIMMs are now interspersed with 1-inch-wide RoB packages, again ensuring noncolliding stalactites and stalagmites. Assuming that cartridges can be designed

with longer dimensions, we can continue to grow the daisy chain to further boost the number of DIMMs per cartridge.

Although the proposed design bears a similarity to the FBDIMM approach of daisy-chained buffer chips, it has been carefully designed to not suffer from the pitfalls that doomed FBDIMM. First, we continue to use standard DDR channels and the RoB chips are on the cartridge rather than on the DIMM. This enables the use of commodity DIMMs. Second, the RoBs simply propagate signals and do not include power-hungry circuits for buffering, protocol conversion, and SERDES. Third, as described next, we introduce collision avoidance logic to simplify the memory controller scheduler.

4.1.3. A Scalable Memory Controller Scheduler. A many-rank system accessed by a cascaded channel requires an adapted memory controller scheduler. In essence, the memory controller must be careful to avoid collisions on the cascaded memory channels. Depending on the ranks involved in already scheduled data transfers, the timing for the next column-read/write must be adjusted. This is done by maintaining a small table with latency constraints. We have synthesized this collision avoidance circuit to timing-correct gate-level netlists and confirmed that the circuit adds area and power overheads under 10% to the memory controller. Since the design of this scheduler is not central to memory interconnect analysis, we do not delve into more details here.

4.1.4. Hybrid DRAM/NVM Hierarchy. It is expected that future off-chip memory systems will support a combination of DRAM and NVM. Several works have explored caching policies for such hybrid DRAM/NVM systems (e.g., Qureshi et al. [2009], Ramos et al. [2011], and Yoon et al. [2012b]). Most of these studies assume that the processor socket has a channel populated with DRAM DIMMs and a separate channel populated with NVM DIMMs. However, such an organization suffers from three weaknesses. First, the processor pins are statically partitioned between DRAM and NVM; as a result, the entire processor-memory bandwidth is not available for the faster DRAM region. Second, some of these studies operate the NVM pins at a frequency lower than that of the DRAM pins, further lowering the overall processor-memory bandwidth. Third, data migration between DRAM and NVM involves the processor and consumes bandwidth on both the DRAM and NVM pins.

Given the expected popularity of future DRAM/NVM hierarchies, it is important to define memory interconnects that can address the preceding problems. Ham et al. [2013] address the third problem by introducing BoB chips on the DRAM and NVM channels, and a new link between these BoB chips; pages can therefore be copied between DRAM and NVM without involving the processor pins. We note that the cascaded channel architecture addresses all three weaknesses listed previously.

The RoB chip decouples the characteristics of cascaded channels. Not only can each channel support different voltages and frequencies, each can also support different memory technologies. Figure 12 describes several possible DRAM/NVM configurations for baseline channels, as well as for RoB-based cascaded channels where the first channel represents a lower-capacity lower-latency region of memory, and the second channel represents a higher-capacity higher-latency region. Whereas the baseline cases allocate some processor pins for DRAM channels and some for slower NVM channels, the RoB-based designs boost processor pin bandwidth by connecting all processor memory pins to fast DRAM that can contain hot pages. The proposed design also allows the copying of data between DRAM and NVM without occupying the processor pins twice. For example, when copying data from NVM to DRAM, the processor first issues a read to the distant NVM; the data transfer on the distant channel is buffered on the RoB; the processor then issues a write to the nearby DRAM; the data for that write is driven on the near channel by the RoB; and the write into DRAM thus leverages the already scheduled data transfer from the distant NVM.

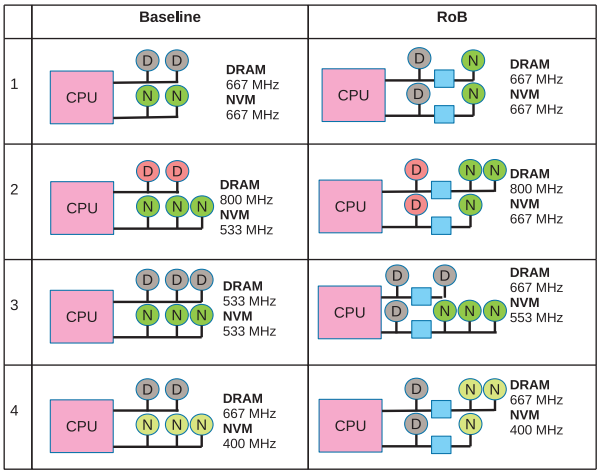


Fig. 12. Four different cases that use DRAM and NVM. The baseline organizations use separate channels for DRAM and NVM. The corresponding RoB architecture implements NVM on a distant cascaded channel while supporting the same DRAM and NVM capacities as the baseline. The 800MHz and 400MHz channels work at 1.5V and 1.2V, respectively, whereas other channels operate at 1.35V.

4.2. Cascaded Channel Evaluation Methodology

To model the power and cost of RoB-based designs, we modified our tool's outer *for* loops to not only partition the required capacity across channels but also across cascaded channels. Every time a cascaded channel is used, we estimate I/O power correctly, treating each RoB chip similar to an LRDIMM buffer. We assume that the second cascaded channel has half the utilization of the first channel. Since the RoB chip partitions a channel into two subchannels, a larger design space is explored since we consider different options for every channel.

For our architectural evaluation, we consider several memory-intensive workloads from SPEC2k6 (libquantum, omnetpp, xalancbmk, milc, GemsFDTD, mcf, leslieeD, and soplex) and NPB [Bailey et al. 1994] (cg, mg, and bt). We generate memory access traces for these workloads with Wind River Simics [Wind 2007]. Two (staggered) copies of these 8-core traces are then fed to the USIMM cycle-accurate memory system simulator [Chatterjee et al. 2012] to create a 16-core workload sharing a single memory channel. This enables tractable simulations of future throughput-oriented architectures. Simics and USIMM parameters are summarized in Table V.

4.3. Cascaded Channel Results

For our DRAM-only analysis, we compare against a baseline memory cartridge with two 533MHz DDR3 channels per BoB with three DDR3L quad-rank LRDIMMs per channel. The RoB-based cartridge has the same capacity as the baseline, but each DDR channel can now operate at 667MHz.

4.3.1. Power Analysis. We use the modified version of our tool to estimate the power of the memory cartridge under high utilization. The baseline has a DDR channel utilization of 70%, whereas with RoB-based cascaded channels, the first channel has a 70% utilization and the second has a 35% utilization. We also assume a read/write ratio of 2.0 and a row buffer hit rate of 50%. The power breakdown is summarized in Table VI. At higher frequencies, the DRAM power is higher because of higher background power. I/O power in the cascaded channel design is also higher because of the increase in channel segments and the increase in frequency. The result is an 8.1% increase in

Table V. Simulation Parameters

Processor	
ISA	UltraSPARC III ISA
Size and frequency	8-core, 3.2GHz
ROB	64 entry
Fetch, dispatch, execute, and retire	Maximum 4 per cycle
Cache Hierarchy	
L1 I-cache	32KB/2-way, private, 1-cycle
L1 D-cache	32KB/2-way, private, 1-cycle
L2 cache protocol	8MB/8-way, shared, 10-cycle Snooping MESI
DRAM Parameters	
DDR3	Micron DDR3-1600 [Micron 2006]
DRAM configuration	Single BoB with 2 channels 3 ECC DIMMs/channel LRDIMM 4 ranks/DIMM
Memory capacity	192GB (half a cartridge)
Memory frequency	533MHz
Memory read queue	48 entries per channel
Memory write queue size	48 entries per channel

Table VI. Power Breakdown for Baseline and Cascaded Designs

	DIMM Power (W)	BoB Power (W)	I/O Power (W)	Total Power (W)	Power/ BW (nJ/B)
Baseline	23.2	5.5	9.4	38.1	7.94
Cascaded	22.6	6.4	12.2	41.2	6.86

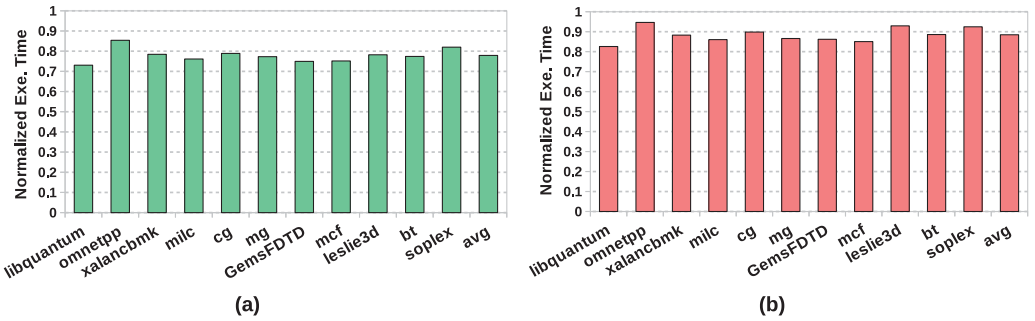
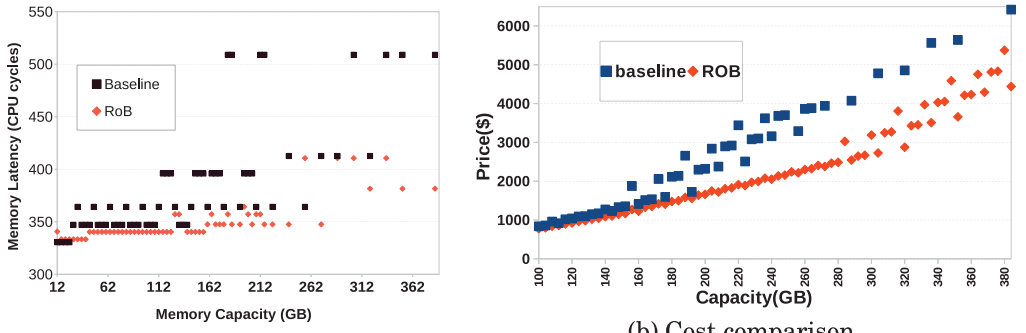


Fig. 13. Execution times for the cascaded channel architecture, normalized against the baseline cartridge (DDR3 (a) and DDR4 (b)).

total cartridge power but lower energy per bit (power/bandwidth). The I/O power is influenced by bus utilization. If we assume 50% and 80% utilization, the cascaded model consumes 4.6% and 7.1% more power than the baseline, respectively.

4.3.2. Performance Analysis. Figure 13 compares execution times for the cascaded design, relative to the baseline cartridge with the same memory capacity. For DDR3 design points, even though RoB traversal adds a few cycles to the memory latency, it



(a) Memory latency comparison.

(b) Cost comparison.

Fig. 14. Memory latency and cost comparison with baseline and RoB approaches, as the memory capacity requirement is varied.

enables a 25% higher bandwidth and lower queuing delays. The net effect is a 22% performance improvement. For the DDR4 design point, the cascaded channels enable a bandwidth increase of 13% and a performance improvement of 12%.

To further boost memory capacity, additional RoB chips can be used to extend the daisy chain and add more DIMMs, without impacting the bandwidth into the processor.

4.3.3. Design Space Explorations. With our extended tool, we evaluate performance as we sweep the design space. Figure 14(a) shows memory latency for all considered design points. Again, the RoB-based designs are superior in nearly every case. Note that the baseline shows lower latency in a few cases—in these cases, the baseline and RoB-based designs end up with similar channel bandwidths and the RoB designs suffer from the extra hop latency.

Similarly, Figure 14(b) shows the cost of constructing the cheapest memory system for a range of memory capacity requirements. RoB-based designs offer more options when configuring a memory system. As a result, for example, a 256GB memory system can be configured in the baseline with two 32GB and one 64GB DPCs, whereas a RoB-based design can use two segments per channel, each with two 32GB DIMMs. For this example, by avoiding expensive 64GB DIMMs, the RoB-based approach yields a 48.5% reduction in cost. We see that in all cases the cascaded approach is superior, with the improvements growing as capacity increases.

This analysis introduces the RoB as an additional knob in our design space exploration and helps identify the best way to configure a memory system for a given capacity requirement. Our extended tool makes the case that cost and performance can be improved by partitioning a standard DDR channel into multiple cascaded segments.

4.3.4. DRAM/NVM Hierarchies. We now turn our attention to the RoB chip's ability to implement hybrid DRAM/NVM hierarchies on cascaded channels. We consider iso-capacity comparisons in four cases, depicted in Figure 12 for baseline and cascaded approaches. The baseline in each case isolates the NVMs to a separate channel. The parameters for the NVM are based on PCM parameters assumed by Lee et al. [2009b]. In the first case, the two designs have the same total bandwidth; the cascaded approach has the potential to do better if most requests are steered to DRAM because of hot page caching. The second case is similar but uses lower-capacity dual-rank 1.5V RDIMMs for higher bandwidth, while increasing the number of NVM DIMMs. The third case increases capacity further; the baseline suffers from low memory bandwidth while the RoB-based design isolates the high capacity and low bandwidth to a single distant channel. The fourth case is a low-power version of the second case.

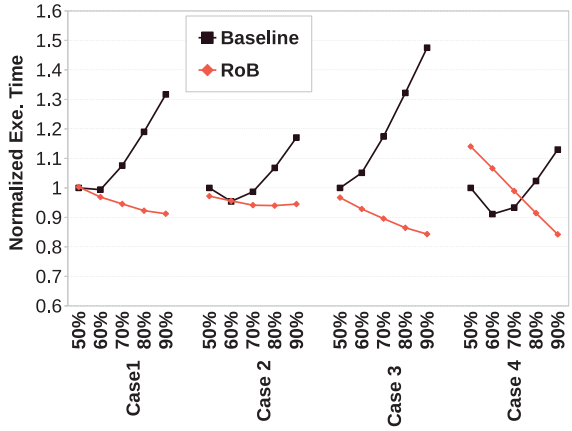


Fig. 15. Normalized execution time (averaged across all benchmarks) for baseline and RoB cases, as the fraction of DRAM accesses is varied from 50% to 90%.

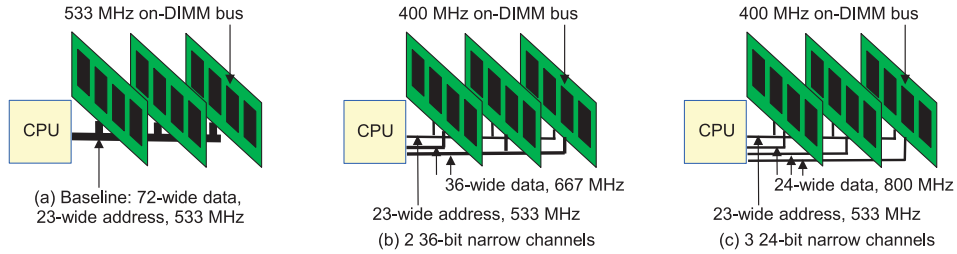


Fig. 16. The baseline (a) and two narrow channel organizations (b and c).

A DRAM/NVM hierarchy must be complemented by OS/hw policies that can move hot pages to low-latency DRAM (e.g., Yoon et al. [2012b], Ramos et al. [2011], and Lim et al. [2009]). We view such policies as an orthogonal effort that is beyond the scope of this work. Here we show results for several assumptions, ranging from 50% to 90% of all accesses being serviced by DRAM. Figure 15 shows the normalized execution time averaged across all benchmarks for the four cases and for varying levels of DRAM activity.

A common theme in these results is that when the DRAM is responsible for more than 60% of all memory traffic, performance starts to degrade in the baselines. This is because a single DRAM channel is being oversubscribed. The cascaded approaches allocate more processor pins for DRAM and can better handle the higher load on DRAM. In nearly all cases, the cascaded approach is a better physical match for the logical hierarchy in DRAM/NVM access. In the fourth power-optimized case, the RoB-based NVM is disadvantaged, relative to the baseline NVM, so it performs worse than the baseline at high NVM traffic rates. The energy trends are similar to the performance trends.

5. THE NARROW CHANNEL ARCHITECTURE

Finally, we introduce a new interconnect topology that is not constrained by the DDR standard. This is also an example of how our tool is useful for on-DIMM evaluations.

5.1. Proposal

5.1.1. Example Narrow Channels. Figure 16(a) shows a standard 72-bit DDR channel supporting three DIMMs at 533MHz. To support high memory capacity without

growing the load on the data bus, we propose to use narrow parallel data channels—two options are shown in Figure 16(b) and (c). The first option in Figure 16(b) implements two 36-bit-wide parallel channels, both operating at 667MHz. For an iso-capacity comparison, we assume that one channel supports two DIMMs and the other supports a single DIMM. These DIMMs use a buffer chip for address and data, similar to the design style of an LRDIMM. The buses between the buffer chip and the DRAM chips on the DIMM are similar to that of a baseline DIMM, but they can operate at a lower frequency and still keep up with the bandwidth demands of the external link. In the example in Figure 16(b), the on-DIMM 72-bit bus conservatively operates at 400MHz and supports an external 36-bit link at 667MHz.

In the example in Figure 16(c), the channel is split into three narrow channels, each 24 bits wide, and each supporting a single DIMM at 800MHz. Again, the on-DIMM 72-bit bus conservatively operates at 400MHz and supports the 24-bit external link at 800MHz.

In both of these designs, we assume that only the data bus is partitioned across the narrow channels. The address/command bus remains the same as before. For instance, a single address/command bus is shared by all three channels and all three DIMMs. This ensures that the new architecture does not result in a large increase in pin count for the processor. Because the address/command bus is driving the same load as the baseline, it continues to operate at the slower 533MHz frequency. Since address/command buses are utilized far less than the data bus, this lower frequency does not make the address/command bus a new bottleneck.

5.1.2. Advantages and Disadvantages. Both of these new narrow channel designs have four primary advantages. First, they support a faster aggregate bandwidth into the processor. Second, they have the potential to reduce DIMM power by operating DIMM components at a lower clock speed. Third, this approach can also be used to grow memory capacity without a corresponding steep penalty in bandwidth. Fourth, just as we saw for the RoB-based cascaded channels, there is a potential to reduce cost by implementing a given memory capacity with many low-capacity DIMMs instead of a few high-capacity DIMMs.

There are three disadvantages as well. The primary disadvantage of course is that nonstandard non-DDR DIMMs will likely be more expensive because they are produced at lower volume. Despite this, we believe that this approach is worth exploring in the era of memory specialization (e.g., similar to how IBM produces custom DIMMs for their Power8 line of processors) [Stuecheli 2014]. The second disadvantage is that a longer transfer time per cache line is incurred. And the third disadvantage is that there is limited rank-level parallelism within each narrow channel. The second and third disadvantages are captured in our simulations and turn out to be relatively minor.

5.1.3. A Two-Tier Error Protection Approach. Since cache lines are aggregated on the buffer chip before returning to the processor, we take this opportunity to also improve error protection. We assume that the DIMM supports some form of error protection (say, SECDED or chipkill), but the processor is kept oblivious of this protection. The buffer chip inspects the bits read from DRAM, performs error detection and correction, and sends just the cache line back to the processor. Since ECC typically introduces a 12.5% overhead on bandwidth, this strategy eliminates or reduces that overhead. To deal with potential link transmission errors, we add a few CRC bits to every data packet returned to the processor. The CRC is used only for error detection. When an error is detected, the buffer chip simply retransmits the data packet. To keep the protocol relatively unchanged, the retransmission can be triggered by the processor requesting the same cache line again. With this two-tier protection (SECDED or chipkill within DRAM and CRC for the link), we are still maintaining data in DRAM with strong

protection, but the processor and link are subject to the overheads of a lightweight error detection scheme.

The reverse is done on a write, where the buffer chip on the DIMM receives a CRC-protected packet, computes the SECDED or chipkill code, and performs the write into memory chips.

If we assume the 36-bit narrow channel in Figure 16(b), a minimum of 15 transfers are required to communicate a 512-bit cache line. This leaves room for a 28-bit CRC code. For a well-constructed 28-bit CRC code, the probability of a multibit transmission error going undetected is very low (2^{-28}, the probability that a random 28-bit string matches the CRC for a new data block). By comparison, DDR4 has support for an 8-bit CRC [Micron 2014] and HMC has support for a 32-bit CRC [Hybrid Memory Cube 2013]. The CRC code can be made even stronger by sharing a code across multiple cache lines. For example, a group of four cache lines can share a 112-bit CRC code. As the size of the CRC code grows, the probability of multibit errors going undetected is exponentially lower.

Our two-tier coding approach further reduces bandwidth requirements by reducing the ECC bits sent back to the processor. In the preceding example, we are sending 540 bits on every cache line transfer instead of the usual 576 bits.

This error handling strategy moves the memory system toward an abstracted memory interface [Pawlowski 2014], where the processor simply asks for and receives data, whereas the details of memory access and memory errors are entirely handled by the DIMM or memory product.

5.2. Evaluation

To evaluate the proposed narrow channel architecture, we use the same simulation infrastructure as in Section 4.2.

Modeling this architecture with our tool is relatively trivial because of the convenient API provided. For the model in Figure 16(c), power was estimated with the following queries:

(1) *DDR(long-range, 800, 1, ddr3, 24, on-main-board)*: The 24-wide DDR3 data bus on the board connecting to 1 DIMM at 800MHz.
(2) *DDR(short-range, 400, 4, ddr3, 72, on-dimm)*: The 72-wide DDR3 data bus on the DIMM connecting to four ranks at 400MHz.
(3) *SDR(long-range, 533, 3, ddr3, 23, on-main-board)*: The 23-wide DDR3 address/command bus on the board connecting to three DIMMs at 533MHz.
(4) *SDR(short-range, 400, 4, ddr3, 23, on-dimm)*: The 23-wide DDR3 address/command bus on the DIMM connecting the buffer chip to four DRAM dies on its left at 400MHz.
(5) *SDR(short-range, 400, 5, ddr3, 23, on-dimm)*: The 23-wide DDR3 address/command bus on the DIMM connecting the buffer chip to five DRAM dies on its right at 400MHz.

Figure 17 shows normalized execution time for the two narrow channel architectures shown in Figure 16. The longer data transfer time and the reduced rank level parallelism per channel introduce second-order negative impacts on performance. But to a large extent, performance is impacted by the higher bandwidth enabled in the narrow channel designs. The new error handling strategy also contributes to the improvement in bandwidth, so the 24-bit and 36-bit channels increase peak bandwidth by 64% and 33%, respectively (without the new error handling strategy, the bandwidth increase would have been 50% and 25%). Overall, the 24-bit and 36-bit channels yield performance that respectively is 17% and 18% higher than the baseline.

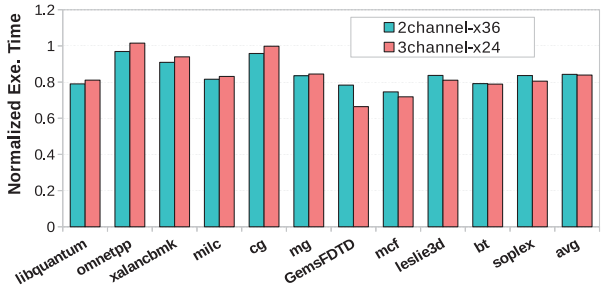


Fig. 17. Execution time for the two narrow channel designs in Figure 16, normalized against the baseline.

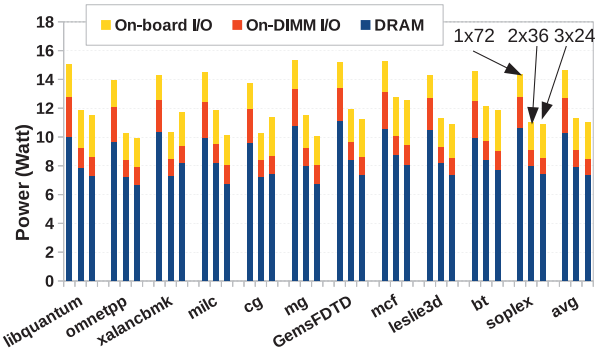


Fig. 18. Memory power for the two narrow channel designs in Figure 16, normalized against the baseline (left).

The power results are shown in Figure 18, which also shows a breakdown across the different components. The on-board I/O power is higher because of the higher frequency for the on-board interconnects; meanwhile, the on-DIMM interconnects and DRAM chips consume less power than the baseline because of their lower frequency. The net result is an overall memory power reduction of 23%. Again, we see that I/O power is a significant contributor to overall memory power, highlighting the importance of precise I/O models as we explore specialized memory architectures.

6. RELATED WORK

Research on memory systems has gained significant traction in the past few years. This is attributed to both interest in emerging nonvolatile memories and relatively modest evolution of memory architecture in recent decades. As a result, there has been a flurry of work in the area of memory simulators and tools to facilitate research on disruptive memories with nontraditional fabric. CACTI-IO [Jouppi et al. 2015] is an I/O modeling tool with its standard model limited to DDR3 configurations with unbuffered DIMMs. Although our tool also focuses on I/O, it is a comprehensive framework that considers cost, power, and noise (jitter and skew), and performs exhaustive search within the tool to find an optimal DIMM configuration for a given memory capacity and bandwidth. In addition to providing support for both on-DIMM and main-board buffers for both DDR3 and DDR4, it supports a wide range of frequencies without any modification to the tool. CACTI 7 is also the first tool to support serial-io with different data rates.

Modeling tools such as the Micron power model [Micron 2005], DRAMPower [Chandrasekar et al. 2012], NVSIM [Dong et al. 2012], and DRAM energy models

by Vogelsang [2010] are primarily targeted at either microarchitecture of memory or DRAM die.

NVSIM is based on the CACTI tool that focuses on emerging nonvolatile memories such as STTRAM, PCRAM, ReRAM, NAND Flash, and Floating Body Dynamic RAM.

Memory simulators such as DRAMSim [Wang et al. 2005], USIMM [Chatterjee et al. 2012], and Ramulator [Kim et al. 2015] are performance simulators that model DRAM timing in a cycle-accurate manner. These tools take input from power models described earlier to calculate memory system power and can benefit from the proposed tool.

Memory DIMMs have been optimized for bandwidth, capacity, and power. The fully buffered DIMM was Intel's solution for extending DDR2 memory capacity [Ganesh et al. 2007]. FBDIMM uses narrow serial links to connect many buffer chips in a daisy chain. HC-DIMM optimized LRDIMM by distributing LRDIMM's memory buffer functionality across multiple smaller buffer chips that are closer to the corresponding DRAM chips on the DIMM [Netlist 2012]. IBM has a custom DIMM for its AMB BoB that supports up to 160 devices [Van Huben et al. 2012]. Apart from these industrial solutions, recent academic works also suggest better DIMM and network organizations. Ham et al. [2013] design a hierarchical tree topology for the memory network to support DRAM and NVMs, and Kim et al. [2013] design a network of HMCs. Decoupled-DIMM decouples channel frequency from DIMM frequency using a custom buffer on DIMM [Zheng et al. 2009]. BOOM [Yoon et al. 2012a] and Malladi et al. [2012] use different approaches to integrate mobile LPDDR chips to reduce idle power. These related works not only highlight the importance of special nonstandard DIMMs but also the need for a proper I/O modeling framework to speed up research in this area.

7. CONCLUSIONS

In the future, we expect that parts of the memory system will move toward specialization. Much of that specialization will likely revolve around new interconnect topologies to connect different types of memory products. This article makes the case that I/O power is a significant fraction of memory power. We therefore develop a tool that models a variety of memory interconnects and provides a framework for design space exploration. With the insights gained from the tool, we devise two new memory network architectures that improve several metrics. We show that partitioning a memory channel into multiple channels (either cascaded or narrow) has a first-order effect on bandwidth and cost. The CACTI 7 tool was also modified and used to characterize power and cost for the proposed architectures. We observed performance improvements of 18% and energy reduction of 23% for the narrow channel architecture, and cost reductions of up to 65% for the cascaded channel architecture.

ACKNOWLEDGMENTS

We thank the anonymous reviewers for many helpful suggestions.

REFERENCES

AMP. 2014. TE DDR2 Connector Model. Retrieved May 6, 2017, from http://www.te.com/documentation/electrical-models/files/slm/DDR2_DIMM_240-Solder_tail.pdf.

D. H. Bailey, E. Barszcz, J. T. Barton, D. S. Browning, R. L. Carter, D. Dagum, R. A. Fatoohi, et al. 1994. The NAS parallel benchmarks. *International Journal of Supercomputer Applications* 5, 3, 63–73. http://www.nas.nasa.gov/Software/NPB/.

G. W. Burr, M. J. Breitwisch, M. Franceschini, D. Garetto, K. Gopalakrishnan, B. Jackson, B. Kurdi, et al. 2010. Phase Change Memory Technology. Retrieved May 6, 2017, from http://arxiv.org/abs/1001.1164v1.

K. Chandrasekar, C. Weis, Y. Li, S. Goossens, M. Jung, O. Naji, B. Akesson, N. Wehn, and K. Goossens. 2012. *DRAMPower: Open Source DRAM Power and Energy Estimation Tool*. Technical Report. DRAMPower.

N. Chatterjee, R. Balasubramonian, M. Shevgoor, S. Pugsley, A. Udipi, A. Shafiee, K. Sudan, M. Awasthi, and Z. Chishti. 2012. *USIMM: The Utah Simulated Memory Module*. Technical Report UUCS-12-002. University of Utah.

Dell. 2010. Dell PowerEdge 11th Generation Servers: R810, R910, and M910 Memory Guidance. Retrieved May 6, 2017, from http://goo.gl/30QkU.

Dell. 2014. Dell PowerEdge R910 Technical Guide. Retrieved May 6, 2017, from http://www.avsys.mx/es/hosting/docs/PowerEdge-R910-Technical-Guide.pdf.

X. Dong, C. Xu, Y. Xie, and N. P. Jouppi. 2012. *NVSim: A Circuit-Level Performance, Energy, and Area Model for Emerging Nonvolatile Memory*. Technical Report. Pennsylvania State University.

ESG Memory Engineering. 2012. Memory for Dell PowerEdge 12th Generation Servers. Retrieved May 6, 2017, from https://www.dell.com/downloads/global/products/pledge/poweredge_12th_generation_server_memory.pdf.

B. Ganesh, A. Jaleel, D. Wang, and B. Jacob. 2007. Fully-buffered DIMM memory architectures: Understanding mechanisms, overheads, and scaling. In *Proceedings of the 2007 HPCA Conference (HPCA'07)*.

T. Ham, B. Chelepalli, N. Xue, and B. Lee. 2013. Disintegrated control for energy-efficient and heterogeneous memory systems. In *Proceedings of the 2013 HPCA Conference (HPCA'13)*.

HP. 2014. HP ProLiant DL580 Gen8 Server Technology. Retrieved May 6, 2017, from http://www.ikthandel.no/pdf/C5AF9260-A142-4A1C-A4F4-BD9063EBE19A.pdf.

HP. 2015. HP ProLiant DL580 G7 Server Technology. Retrieved May 6, 2017, from https://www.hpe.com/h20195/V2/getpdf.aspx/c04128284.pdf?ver=45.

Hybrid Memory Cube. 2013. Hybrid Memory Cube Specification 1.0. Retrieved May 6, 2017, from http://hybridmemorycube.org/files/SiteDownloads/HMC_Specification%201_0.pdf.

IBIS. 2014. IBIS. Retrieved May 6, 2017, from https://ibis.org/.

Intel. 2014. Intel C102/C104 Scalable Memory Buffer Datasheet. Retrieved May 6, 2017, from http://www.intel.com/content/dam/www/public/us/en/documents/datasheets/c102-c104-scalable-memory-buffer-datasheet.pdf.

Intel. 2016. Product Specifications. Retrieved May 6, 2017, from http://ark.intel.com/.

J. Jeddeloh and B. Keeth. 2012. Hybrid memory cube—new DRAM architecture increases density and performance. In *Proceedings of the 2012 Symposium on VLSI Technology*.

N. P. Jouppi, A. B. Kahng, N. Muralimanohar, and V. Srinivas. 2015. CACTI-IO: CACTI with off-chip power-area-timing models. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 23, 7 (2015), 1254–1267.

Perry Keller. 2012. Understanding the New Bit Error Rate Based DRAM Timing Specifications. Retrieved May 6, 2017, from https://www.jedec.org/sites/default/files/Perry_keller.pdf.

G. Kim, J. Kim, J. Ahn, and J. Kim. 2013. Memory-centric system interconnect design with hybrid memory cubes. In *Proceedings of the 2013 PACT Conference (PACT'13)*.

Y. Kim, W. Yang, and O. Mutlu. 2015. *Ramulator: A Fast and Extensible DRAM Simulator*. Technical Report.

B. Lee, E. Ipek, O. Mutlu, and D. Burger. 2009b. Architecting phase change memory as a scalable DRAM alternative. In *Proceedings of the 2009 ISCA Conference (ISCA'09)*.

H. Lee, K.-Y. K. Chang, J.-H. Chun, T. Wu, Y. Frans, B. Leibowitz, N. Nguyen, et al. 2009a. A 16 Gb/s/Link, 64 GB/s bidirectional asymmetric memory interface. *IEEE Journal of Solid-State Circuits* 44, 4, 1235–1247.

K. Lim, J. Chang, T. Mudge, P. Ranganathan, S. K. Reinhardt, and T. F. Wenisch. 2009. Disaggregated memory for expansion and sharing in blade servers. In *Proceedings of the 2009 ISCA Conference (ISCA'09)*.

K. T. Malladi, F. A. Nothaft, K. Periyathambi, B. C. Lee, C. Kozyrakis, and M. Horowitz. 2012. Towards energy-proportional datacenter memory with mobile DRAM. In *Proceedings of the 2012 ISCA Conference (ISCA'12)*.

Micron. 2005. *Calculating Memory System Power for DDR2*. Technical Note TN-47-07. Micron.

Micron. 2006. Micron DDR3 SDRAM Part MT41J256M8. Retrieved from https://www.micron.com/~/media/documents/products/data-sheet/dram/ddr3/2gb_ddr3_sdram.pdf.

Micron. 2014. TN-40-03: DDR4 Networking Design Guide. Retrieved May 6, 2017, from https://www.micron.com/~/media/documents/products/technical-not e/dram/tn_4003_ddr4_network_design_guide.pdf.

Micron. 2015a. LRDIMM. Retrieved May 6, 2017, from https://www.micron.com/products/dram-modules/lrdimm.

Micron. 2015b. System Power Calculator Information. Retrieved May 6, 2017, from https://www.micron.com/support/tools-and-utilities/power-calc.

N. Muralimanohar, R. Balasubramonian, and N. Jouppi. 2007. Optimizing NUCA organizations and wiring alternatives for large caches with CACTI 6.0. In *Proceedings of the 2007 MICRO Conference (MICRO'07)*.

R. Myslewski. 2014. HP busts out new ProLiant rack mount based on Intel's new top o'line server chippery. Retrieved May 6, 2017, from http://www.theregister.co.uk/2014/02/19/hp_busts_out_new_proliant_rack_mount_based_on_intels_new_top_o_line_server_chippery/.

Netlist. 2012. HyperCloud Memory Outperforms LRDIMM in Big Data and Big Mem Applications. Retrieved May 6, 2017, from http://www.marketwired.com/press-release/hypercloud-memory-outperforms-lrdimm-in-big-data-big-memory-applications-nasdaq-nlst-1628259.htm.

F. O'Mahony, J. Kennedy, J. E. Jaussi, and B. Casper. 2010. A 47x10Gb/s 1.4mW/(Gb/s) parallel interface in 45nm CMOS. In *Proceedings of the 2010 IEEE ISSCC Conference (ISSCC'10)*.

J. Ousterhout, P. Agrawal, D. Erickson, C. Kozyrakis, J. Leverich, D. Mazieres, S. Mitra, et al. 2009. The case for RAMClouds: Scalable high-performance storage entirely in DRAM. *ACM SIGOPS Operating Systems Review* 43, 4, 92–105.

R. Palmer, J. Poulton, A. Fuller, J. Chen, and J. Zerbe. 2008. Design considerations for low-power high-performance mobile logic and memory interfaces. In *Proceedings of the 2008 IEEE ASSCC Conference (ASSCC'08)*.

T. Pawlowski. 2011. Hybrid memory cube (HMC). In *Proceedings of the 2011 HotChips Conference (HotChips'11)*.

T. Pawlowski. 2014. The future of memory technology. *Keynote presented at the Memory Forum*.

J. Poulton, R. Palmer, A. M. Fuller, T. Greer, J. Eyles, W. J. Dally, and M. Horowitz. 2009. A 14mW 6.25-Gb/s transceiver in 90nm CMOS. *IEEE Journal of Solid State Circuits* 42, 12, 2745–2757.

M. Qureshi, V. Srinivasan, and J. Rivers. 2009. Scalable high performance main memory system using phase-change memory technology. In *Proceedings of the 2009 ISCA Conference (ISCA'09)*.

L. Ramos, E. Gorbatov, and R. Bianchini. 2011. Page placement in hybrid memory systems. In *Proceedings of the 2011 ICS Conference (ICS'11)*.

SanDisk. 2014. SanDisk Announces ULLtraDIMM Design Win with Huawei. Retrieved May 6, 2017, from https://www.sandisk.com/about/media-center/press-releases/2014/sandisk-announces-ulltradimm-design-win-with-huawei.

SAP. 2013. SAP HANA In-Memory Computing Community. Retrieved May 6, 2017, from http://scn.sap.com/ http://scn.sap.com/community/hana-in-memorycommunity/hana-in-memory.

SAS. 2013. In-Memory Analytics. Retrieved May 6, 2017, from http://www.sas.com/en_us/software/in-memory-analytics.html.

D. B. Strukov, G. S. Snider, D. R. Stewart, and R. Williams. 2008. The missing memristor found. *Nature* 453, 80–83.

Jeffrey Stuecheli. 2014. Power Technology for a Smarter Future. Available at https://www.ibm.com.

Supermicro. 2015. Supermicro Solutions. Retrieved May 6, 2017, from http://www.supermicro.com/products/nfo/Xeon_X10_E5.cfm.

G. A. Van Huben, K. D. Lamb, R. B. Tremaine, B. S. Aleman, S. M. Rubow, S. H. Rider, W. E. Maule, and M. E. Wazlowski. 2012. Server-class DDR3 SDRAM memory buffer chip. *IBM Journal of Research and Development* 56, 1, 32–42.

T. Vogelsang. 2010. Understanding the energy consumption of dynamic random access memories. In *Proceedings of the 2010 MICRO Conference (MICRO'10)*.

P. Vogt. 2004. *Fully Buffered DIMM (FB-DIMM) Server Memory Architecture: Capacity, Performance, Reliability, and Longevity*. Intel Developer Forum.

D. Wang, B. Ganesh, N. Tuaycharoen, K. Baynes, A. Jaleel, and B. Jacob. 2005. DRAMsim: A memory-system simulator. *SIGARCH Computer Architecture News* 33, 4, 100–107.

Wikipedia. 2014. DDR4 SDRAM. Retrieved May 6, 2017, from http://en.wikipedia.org/wiki/DDR4_SDRAM.

Wind. 2007. Wind River Simics Full System Simulator. Retrieved May 6, 2017, from http://www.windriver.com/products/simics/.

D. H. Yoon, J. Chang, N. Muralimanohar, and P. Ranganathan. 2012a. BOOM: Enabling mobile memory based low-power server DIMMs. In *Proceedings of the 2012 ISCA Conference (ISCA'12)*.

H. Yoon, J. Meza, R. Ausavarungnirun, R. A. Harding, O. Mutlu. 2012b. Row buffer locality aware caching policies for hybrid memories. In *Proceedings of the 2012 ICCD Conference (ICCD'12)*.

M. Zaharia, M. Chowdhury, M. Franklin, S. Shenker, and I. Stoica. 2010. Spark: Cluster computing with working sets. In *Proceedings of the 2010 HotCloud Conference (HotCloud'10)*.

H. Zheng, J. Lin, Z. Zhang, and Z. Zhu. 2009. Decoupled DIMM: Building high-bandwidth memory system from low-speed DRAM devices. In *Proceedings of the 2009 ISCA Conference (ISCA'09)*.

Received October 2016; revised Febuary 2017; accepted March 2017