BOOM: Enabling Mobile Memory Based Low-Power Server DIMMs

Doe Hyun Yoon Jichuan Chang Naveen Muralimanohar Parthasarathy Ranganathan
Intelligent Infrastructure Lab, Hewlett-Packard Labs
{doe-hyun.yoon, jichuan.chang, naveen.muralimanohar, partha.ranganathan}@hp.com

Abstract

To address the real-time processing needs of large and growing amounts of data, modern software increasingly uses main memory as the primary data store for critical information. This trend creates a new emphasis on high-capacity, high-bandwidth, and *high-reliability main memory systems. Conventional and recently-proposed server memory techniques can satisfy these requirements, but at the cost of significantly increased memory power, a key constraint for future memory systems. In this paper, we exploit the low-power nature of another high volume memory component—mobile DRAM—while improving its bandwidth and reliability shortcomings with a new DIMM architecture. We propose* Buffered Output On Module *(BOOM) that buffers the data outputs from multiple ranks of low-frequency mobile DRAM devices, which in aggregation provide high bandwidth and achieve chipkill-correct or even stronger reliability. Our evaluation shws that BOOM can reduce main memory power by more than* 73% *relative to the baseline chipkill system, while improving average performance by* 5% *and providing strong reliability. For memory-intensive applications, BOOM can improve performance by* 30–40%*.*

1. Introduction

The amount of data we collect, process, and store is growing exponentially, at a higher rate than even Moore's law [20]. To maximize the value of such big data, real-time analytics and search systems are expected to digest, index, and answer queries at the speed of interactive business transactions. In order to address the needs for low-latency analytics on large and growing amounts of data, modern software increasingly exploits main memory (as opposed to persistent storage) as the primary data store for critical business and scientific information. In the Internet and social network realm, Google's web search infrastructure services queries entirely out of in-memory indices, while Facebook, Zynga and others rely on *memcached* servers to cut the latency of key-value searches. In the enterprise space, SAP HANA, Oracle TimesTen and VoltDB are just a few recent examples of emerging databases that host the whole dataset in memory. With multicore processing and large-memory hardware, these systems often can provide orders of magnitude better performance and interactive user experiences. To support such emerging workloads, a new emphasis is placed on high-capacity, high-bandwidth, and high-reliability main memory systems.

Main memory, on the other hand, has become a significant contributor to the total power in modern servers. For example, to support a 2 terabyte in-memory database, DRAM power can account for 30–57% of total server power when populating 128 DDR3 DIMMs in an 8-socket server [2]. In such large-memory configurations, reducing main memory power is as important as, if not more important than, reducing processor power. The challenges with memory power are likely to be exacerbated when entering the exascale era, where DRAM power will become dominant as the processor's energy efficiency continues to improve. Indeed, DRAM power, coupled with limited memory capacity and resilience support, is one of the top challenges for exascale computing [17].

The combination of these two trends motivates new architectures for high-performance, high-capacity, high-reliability, *and* low-power server DIMMs. Today's solutions are inadequate as they cannot satisfy *all* of these conflicting requirements. For example, *chipkill-correct* can dramatically reduce the uncorrectable error rate [30] by mandating a certain minimum number of ECC (error checking and correcting) chips per rank. But in current DIMM organizations, chipkill-correct limits the use of wide-datapath DRAM chips, which can lower power [10], increases ECC storage overhead, and often requires lock-step transfers across memory channels, reducing channel-level parallelism. Capacity expansion techniques such as load-reduced DIMM (LR-DIMM) and buffer-on-board (BoB) do not directly reduce DRAM power, and recent low-power proposals [38, 7] incur increased ECC storage overhead, especially for chipkill-correct.

In this paper, we exploit the low-power nature of another volume component—mobile memory—analogous to the recent demonstrations of mobile processors in scale-out servers [18, 12]. While optimized for low power, mobile memory devices lack critical server memory features: high bandwidth and high reliability. The key open question is how to architecturally enable *server* DIMMs built from mobile memory.

We propose *Buffered Output On Module* (BOOM) to answer this question. BOOM buffers the outputs from multiple ranks of low-frequency mobile DRAM to provide the data and ECC bits for an entire cache block. Similar to LRDIMM, the on-module buffer can support a larger number of DRAM chips within a DIMM, yielding higher capacity. Buffering the outputs from multiple ranks allows the use of low-frequency, hence low-power, DRAM chips (e.g., 400MHz LPDDR2) that in aggregation match the bandwidth of server memory channels (e.g., 1600MHz DDR3). As a cache block is partitioned and serviced across multiple ranks, the ECC chips in these ranks are grouped together to enable chipkill-correct or even stronger protection with low ECC overheads. By integrating these aspects, BOOM can simultaneously satisfy the multi-dimensional requirements of high capacity, high bandwidth, high reliability and low power for future server memory.

Our paper makes the following contributions:

1. We propose the BOOM architecture with an internally wide data bus. BOOM can meet the bandwidth and reliability demands of server memory with low-frequency mobile memory devices. The technique is simple to incorporate and requires no change to commodity DRAM chips and only minimal changes in the memory controller.

2. We describe novel designs to utilize the multiple ECC symbols across ranks: effective detection and correction of I/O pin failures and chipkill support for wide DRAM (e.g., $\times 16$) at low frequency. Together, they enable mobile memory devices in servers, drastically saving memory power.

3. We evaluate BOOM using cycle-based simulations. Our results demonstrate more than 73% memory power reduction and 5% average speedup, still providing chipkill-correct level reliability.

2. Background

This section briefly reviews a modern main memory architecture, its key parameters, and the implications on bandwidth, capacity, power, and reliability.

2.1. Memory Organization and Parameters

Figure 1 illustrates the organization of a typical DRAM system. Modern processors have multiple on-chip integrated memory controllers (MCs), each of which has one or more physical channels. A physical channel consists of an address/command bus (*ABUS*), a data bus (*DBUS*), and a few control signals (e.g., clocking and power control). A DBUS in a typical physical channel has 64 wires for data and 8 additional wires for ECC. An MC may have multiple physical channels to form a wide logical channel; e.g., a 128-bit wide channel for chipkill-correct (more details later).

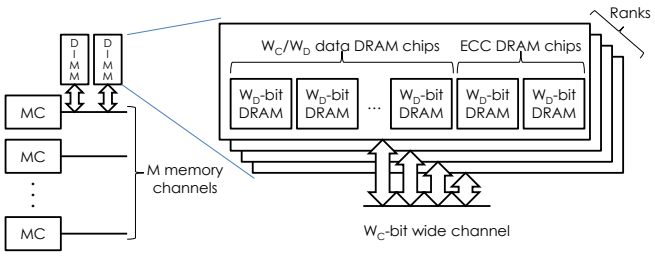


Figure 1: Memory system organization with channels, ranks, and DRAM chips.

Each channel has several *DIMMs* (dual in-line memory modules). A DIMM consists of 1–8 internal *ranks*. A rank is a minimal logical device that an MC can control independently and includes a set of DRAM *chips*.

A DRAM chip is the actual storage device with address/command, control, and data pins. DRAM data width (W_D) can be 4, 8, 16, or 32 bits, commonly referred to as $\times 4$, $\times 8$, $\times 16$, or $\times 32$, respectively. $\times 32$ devices are not available in DDRx, whereas LPDDRx devices are $\times 16$ and $\times 32$ only.

2.2. Implications on the Memory System

Bandwidth: DRAM performance growth has almost entirely come from bandwidth increase, while access latency has reduced very slowly. Three parameters determine the peak bandwidth of a processor: the number of memory channels (M), channel width (W_C), and channel frequency (F_C). Limited scaling in packaging technology makes little room for improving M and W_C. Hence, increasing F_C has been the primary means for high bandwidth.

High channel frequency mandates high-frequency DRAM chips in the current memory systems because a memory channel and the ranks attached to the channel use the same width. Each DRAM generation has increased DRAM frequency: 200–400MHz in DDR, 400–800MHz in DDR2, and 1066–1600MHz in DDR3.

High-frequency DRAM chips employ burst-mode transfers to keep DRAM internal logic at low speed. A DRAM transaction transfers a burst of consecutive data, and *burst length* (BL) is the number of data transfers for a transaction. Long burst transfers enable high-frequency DRAM: BL is 2 in DDR, 4 in DDR2, and 8 in DDR3.

Power: Two major components of DRAM power are background and activate power.

Background power primarily depends on DRAM frequency [6]. As each DRAM chip includes PLL/DLL circuitry, total system background power is also proportional to the number of DRAM chips in a system. Wide DRAM chips can reduce the number of total DRAM chips since each rank has W_C/W_D DRAM chips. If total capacity is the same, wider DRAM configurations use less power than narrower DRAM configurations [10].

An *ACTIVATE* command fetches a row in a bank (1k bits for $\times 4$ and $\times 8$ and 2k bits for $\times 16$) to a row buffer. Recent multicore, multithreaded processors interleave memory accesses from multiple threads and reduce row-buffer locality [7, 32]; the MC activates an entire row but accesses only a small fraction, wasting power – the *overfetch* problem. Wide DRAM configurations and narrow DRAM channels can reduce activate power by mitigating overfetch.

Reliability: High-end servers and datacenters use a large amount of memory devices, increasing the likelihood of memory failures, and business-critical applications require high reliability and availability. Hence, server-class memory systems implement stringent memory protection mechanisms, such as *chipkill-correct* [13] (also referred to as *single-chip sparing*).

Chipkill-correct is a single chip failure correction and double chip failure detection capability, allowing a system to operate continuously even with a DRAM chip failure. Chipkill-correct uses a symbol-based Reed-Solomon (RS) code [28].

A *symbol* is b-bit data. For chipkill-correct, we make a symbol b-bit data out of a DRAM device such that a chip failure appears as a symbol error.

Recent chipkill-correct implementations use a two-ECC-symbol RS code and take advantage of an *erasure* – an erasure is a symbol error whose location is known. Correcting an erasure is much easier than correcting an error. Two ECC symbols can correct an error with unknown location, but one ECC symbol is enough to correct an erasure.

If the MC detects a chip failure, it corrects the chip failure and memorizes the failed chip location, after which the chip failure becomes an erasure [25, 26]. When accessing the memory rank with the failed DRAM chip, the MC uses one ECC symbol for correcting the erasure (the known failed DRAM chip) and the other ECC symbol for further detecting an additional DRAM chip error (but cannot correct it).

The storage overhead of chipkill-correct depends on DRAM data width (W_D). A typical chipkill-correct uses two physical channels in lock-step mode to construct a 128-bit wide logical channel. With two ECC chips, the ECC storage overhead is $2 \times W_D$ bits per 128-bit data; the relative storage overheads are 6.25% for $\times 4$, 12.5% for $\times 8$, 25% for $\times 16$, and 50% for $\times 32$ DRAM. Because overheads above 12.5% are unacceptable, most servers use $\times 4$ or $\times 8$ DRAM configurations.

Though $\times 4$ chipkill-correct allows 64-bit wide channels at 12.5% overhead, recent systems still use 128-bit wide channels (and 16-bit ECC) for $\times 4$ DRAM to enable stronger protection such as DDDC (double device data correction) [9].

Table 1: Key memory system parameters and their implications.

Parameters	Power	Bandwidth	Reliability	Capacity
Channel width ↓	—	—	↓	—
DRAM width ↑	↓	—	↓	↓
DRAM freq. ↓	↓	↓	—	↑

Capacity: High capacity is a key requirement for emerging in-memory databases and other large-memory applications. For a given technology generation, capacity is a function of number of channels, DIMMs per channel, ranks per DIMM, and chips per rank.

While packaging constraints limit the number of channels, signal integrity makes it difficult to increase the number of DIMMs per channel and ranks per DIMM. For example, high-frequency DDR3 channels allow only $1-2$ DIMMs per channel, but lowering the frequency allows one extra DIMM [3]. Registered DIMMs (R-DIMMs) and load-reduced DIMMs (LR-DIMMs) isolate electrical signals (ABUS only in R-DIMMs and both ABUS and DBUS in LR-DIMMs) inside and outside the module, mitigate the signal integrity problem, and allow more DIMMs per channel and more ranks per DIMM, enabling large capacity memory even at high frequencies.

The number of DRAM chips per rank is W_C/W_D. Hence, narrow DRAM chips can increase memory capacity if DRAM chip capacity is the same for different DRAM data widths.

2.3. Inter-Dependence

As discussed, each DRAM system parameter affects bandwidth, power, reliability, and capacity in a different way. For example, high frequency DRAM chips improve bandwidth but negatively impact power and capacity. Wider DRAM chips reduce the number of DRAM chips per rank, lowering DRAM power, whereas narrower DRAM chips help reduce chipkill-correct overhead and increase capacity. Table 1 summarizes the various tradeoffs that exist between the design parameters. Even within this simplified design space, no single combination can meet all of the requirements of server memory systems.

A Case of Mobile Memory: LPDDRx, originally developed for embedded mobile applications, optimizes memory power by using low supply voltage and additional power-saving states. Compared to DDRx, mobile DRAM has lower frequency and wider data path, reducing DRAM power. Mobile memory chips provide also large capacity, and the high volume target market keeps lowering the cost of mobile memory, making LPDDRx an attractive memory device for building high-capacity DIMMs at low cost and low power.

LPDDRx lacks, however, important server memory features: high bandwidth and high reliability. Compared to 1600MHz DDR3, the fastest LPDDR2 memory is

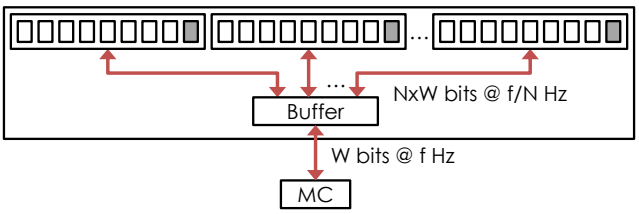


Figure 2: The key idea of the BOOM architecture.

only 1067MHz, while most in-production chips are 667–800MHz. Halving today's server memory bandwidth will cause significant performance degradation and increase the whole system's energy consumption, defeating the purpose of the low power memory system. Furthermore, implementing chipkill-correct with $\times 16$ or $\times 32$ LPDDRx DRAM incurs more than 12.5% ECC overheads, making it impractical for most server designs.

3. The BOOM Architecture

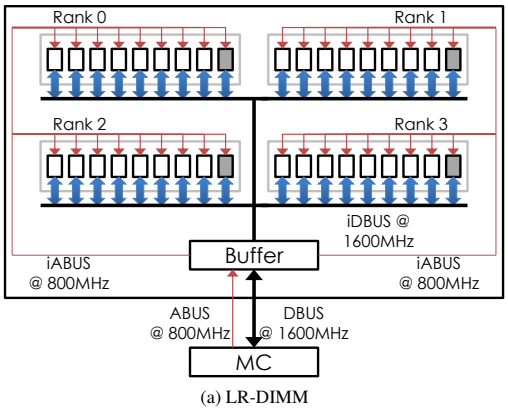
We propose the *BOOM* (Buffered Output On Module) architecture to simultaneously improve power and reliability without negatively impacting capacity and bandwidth. Figure 2 illustrates the high-level design of BOOM. The key innovation is a *wide DIMM-internal data path* that allows low-frequency, wide DRAM (e.g., LPDDRx), while still providing high bandwidth and chipkill-correct level reliability.

3.1. Design of the BOOM Architecture

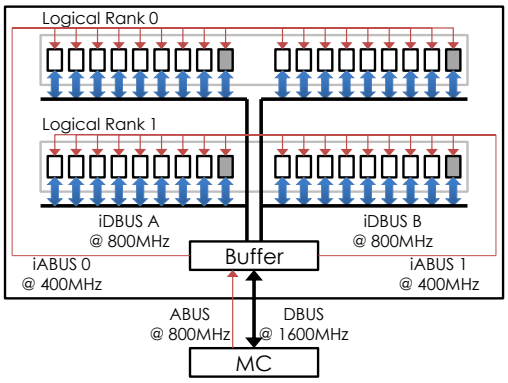
Figure 3 compares our design of the BOOM architecture with buffered DIMM designs such as LR-DIMM. Figure 3(a) shows a typical LR-DIMM design with 1600MHz data rate (the highest DDR3 speed). LR-DIMM's internal data bus (iDBUS) has the same width as the external data bus (DBUS). Two internal address buses (iABUS) are used to reduce electrical load, both conveying the same address/command. LR-DIMMs are functionally identical to unbuffered DIMMs (U-DIMMs), except that the buffer chip adds delays (1 cycle for address/command and 1 cycle for data).

Details of the BOOM Architecture: Figure 3(b) illustrates the BOOM architecture with a $2\times$ wide internal data path. Here, the internal data bus (iDBUS) is twice as wide as the external DBUS, and an $2\times$ wide logical rank consists of 18 $\times 8$ DRAM chips. We can generalize this architecture to support a $N\times$ internal data path (BOOM $N\times$): the wider internal data path runs at slower speed (only $1/N$ of the external bus frequency) and allows a larger number of ECC chips at the same relative storage overhead. The BOOM $N\times$ architecture has the following advantages:

• BOOM $N\times$ uses $N\times$ slower DRAM and dramatically reduces DRAM power. Slow DRAM is relatively cheaper, further reducing the component cost for large memory



(a) LR-DIMM



(b) BOOM with a $2\times$ internal bus

Figure 3: LR-DIMM vs. BOOM at 1600MHz with $\times 8$ DRAMs. Rectangles in a rank are DRAM chips (gray for ECC chips).

servers. The power and cost benefits can significantly reduce the total cost of ownership (TCO).

• BOOM can keep up with the fast external bus even with slow DRAM, guaranteeing high performance.

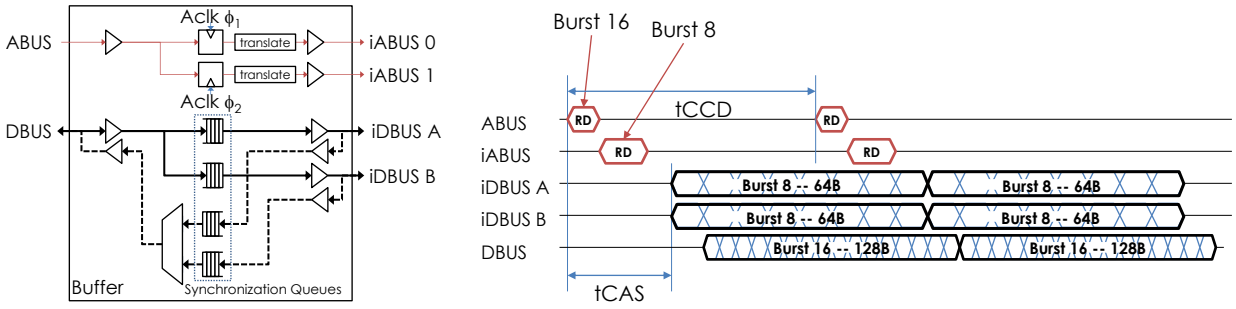
• BOOM's buffer chip enables large memory capacity, similar to LR-DIMMs. Further, BOOM uses low-frequency DRAM and mitigates signal integrity issues within the DIMM; hence, BOOM potentially allows a larger number of ranks per DIMM than LR-DIMM.

• BOOM's wide internal bus enables high reliability, enabling chipkill-correct with a single 64-bit wide channel and wide DRAM chips, e.g., $\times 16$ DRAM (more details in Section 4.3).

Putting these in the context of mobile DRAM based DIMMs, BOOM $4\times$ can use $\times 16$ 400MHz LPDDRx chips, allowing it to retain mobile DRAM's low power advantage, while providing the same bandwidth as 1600MHz DDR3 and chipkill protection at 12.5% ECC overhead.

3.2. Buffer Chip

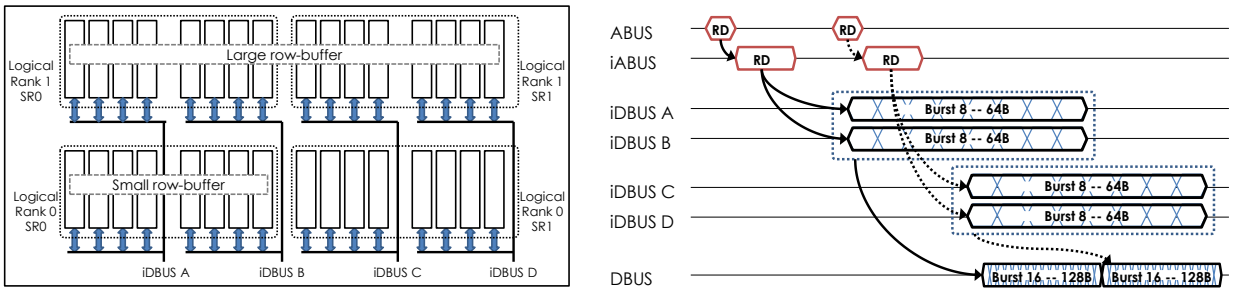
The buffer chip is an essential part of the BOOM architecture, illustrated in Figure 4(a). The buffer chip's primary role is to relay signals between the fast external bus and the



(a) The buffer chip

(b) Read timing (only one iABUS is depicted for the simplicity)

Figure 4: The buffer chip implementation and an example read operation's timing in the $2\times$ BOOM architecture.



(a) Sub-ranked $4\times$ BOOM using $\times 16$ DRAM

(b) Read timing (only one iABUS is depicted for the simplicity)

Figure 5: The BOOM architecture with sub-ranking.

slow internal buses. It also includes a set of synchronization queues to handle the burst length differences between the external and internal buses. Figure 4(b) shows the timing of an example read operation in BOOM $2\times$: (1) a read command on ABUS is relayed to iABUS; (2) after tCAS (column access latency), DRAM transmits two concurrent burst-8 transfers on the 800MHz iDBUS A and B (each 64-bit wide); and (3) the buffer chip relays the two data blocks to the DBUS running at 1600MHz, as one 128B data block over a burst-16 transfer. In this example, data from iDBUS A and B are immediately passed on to the fast external DBUS so the synchronization queues in the buffer chip can be simply flip-flops.

Similarly, the buffer chip splits the fast external ABUS into multiple slow internal buses to provide sufficient address/command bandwidth. It also provides address/command *translation*, when the memory controller and DRAM chips use different burst lengths or address/command formats (e.g., the external bus uses DDR3 protocol, while DRAM chips are LPDDR2). iABUS 0 and 1 in the BOOM architecture can carry two different commands in parallel. For design simplicity, we restrict iABUS 0 and iABUS 1 to each manage half of the total ranks in a DIMM (Figure 3(b)). The number of internal ABUS scales as the internal DBUS width increases; e.g., 4 iABUSes in a BOOM $4\times$ configuration.

3.3. BOOM with Sub-Ranking

While the BOOM architecture enables low-frequency devices to reduce DRAM background power, its wide internal data path has several shortcomings: high activate power, exacerbating overfetch; increased access granularity; and performance degradation due to reduced rank-level parallelism. To address these issues, we apply the sub-ranking technique [33, 37, 7, 36] to BOOM.

Figure 5(a) illustrates sub-ranking with $4\times$ BOOM as an example. A wide logical rank (4×64 bit-wide) is divided into two sub-ranks. The MC can independently access each sub-rank, halving the row buffer size and rank interface width and enabling sub-rank level parallelism.

Figure 5(b) shows an example read operation with sub-ranking: (1) a read from sub-rank 0 is served via iDBUS A and B, and relayed to the external bus; (2) meanwhile, another read from sub-rank 1 is issued to keep the external bus busy. Note that back-to-back requests should be routed to different sub-ranks in order to fully utilize the external bus. Unlike Figure 4(b), a block transfer on the iDBUS takes longer than the transfer on the DBUS, and sub-rank level parallelism is essential to fully utilize the external DBUS. Consequently, the synchronization queues in the buffer chip should be large enough to buffer up to a cache block on each data path.

4. Evaluation Methodology

4.1. Performance and Power Models

We use a multicore simulator [4] built on top of PIN [19]. The event-driven simulator models many in-order cores

Table 2: SPEC CPU 2006 workload mixes.

SPEC INT	CINT-HIGH	mcf, libquantum, omnetpp, astar
	CINT-MED	gcc, gobmk, h264ref, xalancbmk
SPEC FP	CFP-HIGH	milc, soplex, GemsFDTD, lbm
	CFP-MED	bwaves, zeusmp, leslie3d, wrf

with simultaneous multithreading, caches, coherence directories, and MCs. The MC model includes request buffering in the queue, DRAM scheduling (read, write, activate, precharge, etc), power control, as well as conflict/contention of banks, ranks, and address/data buses.

We use Micron's DRAM power calculator [6] to estimate memory power. We calculate DDR3 power directly using the power model with I_{DD} values extracted from datasheets [21, 22, 23]. Similarly, we estimate LPDDR2 power using the same power model, further incorporating LPDDR2's multiple V_{DD} planes (V_{DD1}, V_{DD2}, V_{DDCA}, and V_{DDQ}) and using I_{DD} values from the LPDDR2 datasheets [24].

The buffer chip is a relatively simple device that only relays address/command and data. We assume that I/O power dominates the buffer chip power and estimate the I/O power using the DDR3 I/O power model [6]. In addition, we conservatively estimate 500mW as non-I/O power including static and dynamic power in logic and DLL/PLL.[1] Our buffer chip power model is conservative, and we expect an actual implementation will use less power.

4.2. Workloads

We simulate a wide range of workloads, using subsets of SPLASH2 [34] and PARSEC [11] benchmark suites as well as multiprogrammed workload mixes from SPEC CPU 2006 [31]. Although we mainly focus on memory-intensive workloads to study performance and power impacts, we also present the results of non-memory-intensive workloads. We use the `simlarge` input set for PARSEC. The input sizes for SPLASH2 applications are 1024k points for `FFT`, 8M integers for `RADIX`, and car for `RAYTRACE`. Table 2 lists the SPEC CPU 2006 multiprogrammming workloads used in our evaluation. We skip the initialization phase for multithreaded applications. For multiprogrammed workloads, we use SimPoint [14] to find each application's representative regions and their weights. The number of simulated instances per region is set proportional to its weight. We simulate 200 million memory instructions per workload unless it finishes earlier.

[1] A similar on-DIMM buffer chip [37] running at 1066MHz is estimated to be 804mW I/O power and 150 mW non-I/O power. Since the buffer chip in BOOM has more pins for the wide internal data path and runs at 1600MHz, we scale the non-I/O power result in [37] (conservatively) to 500mW. We obtained 17–27mW/Gb/s in our evaluation (Section 5), which is in line with the survey from circuit research [27].

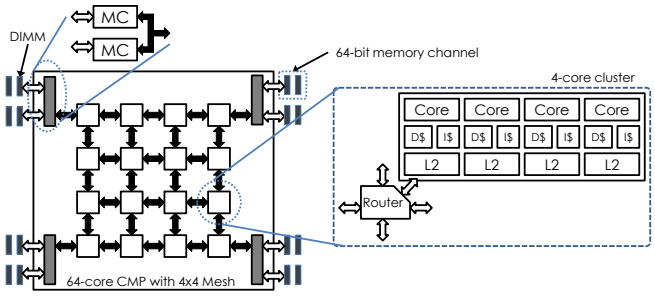


Figure 6: Base system configuration

4.3. System Configurations

Base System Configuration: We model an aggressively scaled 64-core CMP (chip multiprocessor) targeting 22nm technology. As depicted in Figure 6, the CMP consists of a 4×4 mesh network of clusters. Each cluster has 4 processor cores sharing a 1MB 16-way multibanked L2 cache and a directory for maintaining cache coherence. Each core is in-order, running at 3.5GHz, and supports up to 4 concurrent threads, totaling 256 threads per chip.

The CMP includes eight 64-bit wide physical memory channels (72-bit including ECC). Two physical channels are placed at each of the 4 corners of the chip so that the MC can combine them to form a 128-bit wide channel in lock-step mode, if needed. We assume 2 LR-DIMMs per channel, 8 ranks per DIMM, and 2GB per rank, yielding a total of 256GB main memory. Each physical memory channel has a data rate of 12.8GB/s (64 bits $\times$ 1600MHz), leading to 102.4GB/s total off-chip bandwidth.

The MC has a 64-entry queue and uses the FR-FCFS [29] scheduling with closed page policy (a DRAM row is closed when there is no row-buffer-hit request pending in the queue). We implement a power control mechanism that aggressively puts a rank to low-power mode when there is no pending request to the rank. We only use fast-exit power-down mode for high performance.

Baseline Chipkill-Correct: The baseline CMP ties two physical channels to form a 128-bit wide logical channel. The wide channels do not degrade the peak bandwidth but reduce channel-level parallelism – only 4 logical channels. The 128-bit wide channel has 16 bit ECC, supporting DDDC with $\times 4$ and chipkill-correct with $\times 8$ DRAM (chipkill-correct for $\times 16$ is not possible in the baseline). The minimum access granularity is 128B due to burst 8 in DDR3, which sets the cache line size to 128B.

BOOM Configurations: A BOOM configuration is denoted as BOOM-N*n*-*X*-*Y*-S*z*, where n is the number of iDBUS, X is DRAM type (D is DDR3, and L is LPDDR2), Y is DRAM frequency (typically $1600/n$MHz), and z is the number of sub-ranks. Table 3 lists the evaluated baseline and BOOM configurations.

Table 3: Evaluated baseline and BOOM configurations (chipkill-correct requires 2 or more ECC DRAMs per rank/sub-rank).

	# channels	# ranks per DIMM	Burst length		# ECC DRAMs per rank			ECC overhead	row-buffer size		
			DBUS	iDBUS	×4	×8	×16		×4	×8	×16
Baseline	4	8	8	8	4	2	N/A	12.5%	32kB	16kB	N/A
BOOM-N2-D-800-S1	8	4	16	8	4	2	N/A	12.5%	32kB	16kB	N/A
BOOM-N4-L-400-S1	8	4	16	4	N/A	N/A	2	12.5%	N/A	N/A	32kB
BOOM-N4-L-400-S2	8	4	16	8	N/A	N/A	4	25%	N/A	N/A	16kB

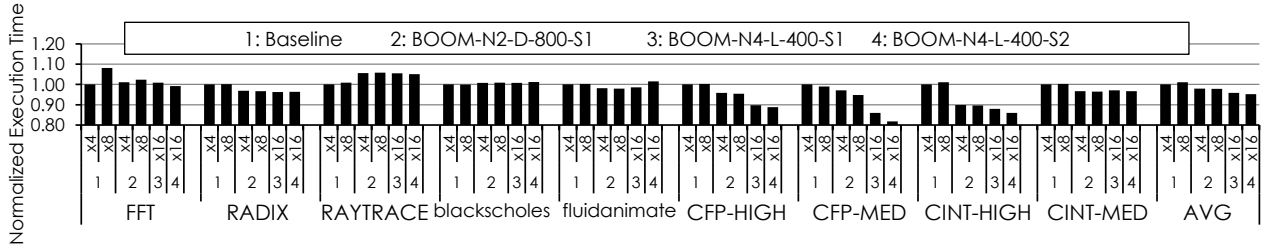


Figure 7: Execution time normalized to baseline ×4 (the lower, the better).

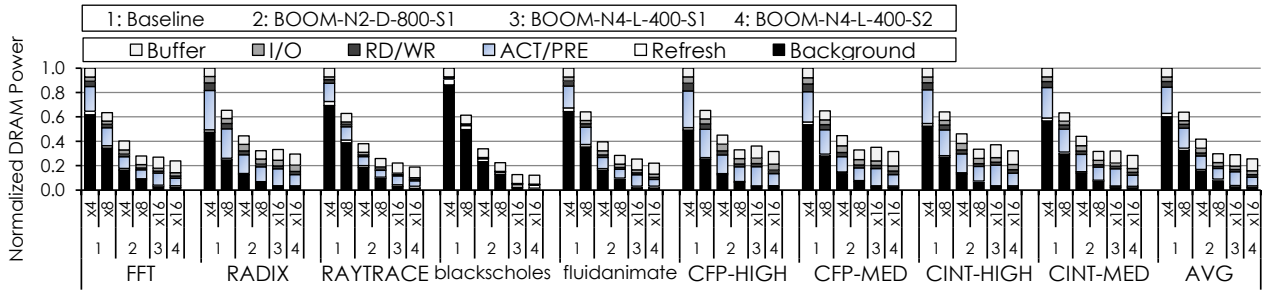


Figure 8: Breakdown of DRAM power consumption (the lower, the better).

While numerous BOOM designs are possible, we restrict our evaluation only to the designs with 128B access granularity and less than 12.5% ECC storage overhead (except for BOOM-N4-L-400-S2, which has 25% ECC overhead). Note that the number of logical ranks is $8/n$ except in LPDDR configurations, which need twice the number of ranks per DIMM to reach 256GB total capacity. Even though the BOOM configurations use 128B line size, a cache line is transferred via a 64-bit physical channel with a longer burst of 16. BL 16 access increases latency but has an advantage of increased channel-level parallelism (total 8 logical channels). BOOM-N4-L-400-S2 can reduce access granularity to as small as 64B using LPDDR2's BL 4 access. Note that it is difficult to use 64B lines in baseline systems due to chipkill-correct constraints. We evaluate the effects of smaller cache line size in Section 5.

5. Evaluation Results

Performance: Figure 7 compares the execution time of the baselines and BOOM configurations (normalized to that of the baseline ×4 configuration) and shows that BOOM improves performance by 5% on average. In the baseline chipkill, using ×8 slightly degrades performance (by 1% on average) due to its smaller row buffer size. BOOM configurations (BOOM-N2-D-800-S1, BOOM-N4-L-400-S1, and BOOM-N4-L-400-S2, marked as configurations 2, 3, and 4 in Figure 7, respectively) perform very close to the baseline ×4 in `FFT`, `RADIX`, `blackscholes`, `fluidanimate`, and `CINT-MED`. The BOOM architecture degrades `RAYTRACE` by 5%. The performance loss is because the buffer chip adds non-trivial delay when it relays data and command to/from the low-frequency internal buses, even though low frequency DDR3 and LPDDR2 have comparable DRAM latencies to those of 1600MHz DDR3. For memory intensive workloads (e.g., `CINT-HIGH`, `CINT-MED`, and `CFP-HIGH`), BOOM improves performance by 5–18% mainly due to its higher channel-level parallelism (detailed analysis presented later in Figure 9).

DRAM Power: Figure 8 presents normalized DRAM power; BOOM with LPDDR uses only less than 30% compared to the baseline ×4 with 1600MHz DDR3.

Wide DRAM configurations (×8 and ×16) and low frequencies (800MHz and 400MHz) reduce DRAM background power. Compared to DDR3 configurations, LPDDR2, optimized to low power, is even more effective in minimizing background power. BOOM-N4-L-400-S1, however, increases dynamic power because the 4× wider internal data bus activates more DRAM chips per access and use more DRAM power than BOOM-N2-D-800-S1 in memory intensive applications (`RADIX`, `CFP-HIGH`, `CFP-MED`, `CINT-HIGH`). With sub-ranking, BOOM-N4-L-400-S2 reduces activate/precharge power and achieves the lowest overall power (only 24% of the baseline ×4 on average).

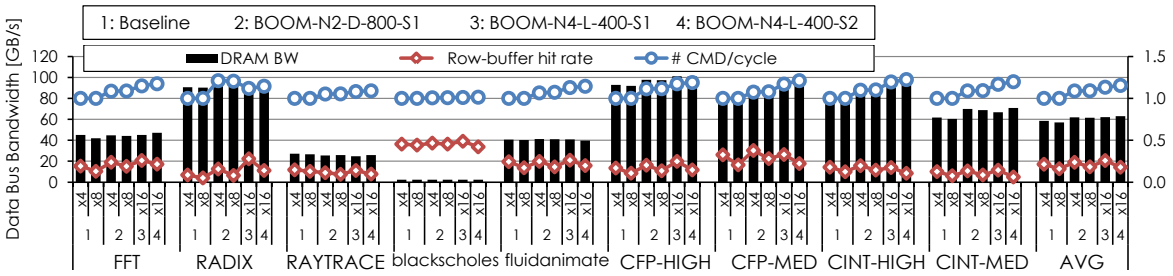


Figure 9: DRAM channel utilization (peak BW: 104.2GB/s), page hit rate, and command issue statistics.

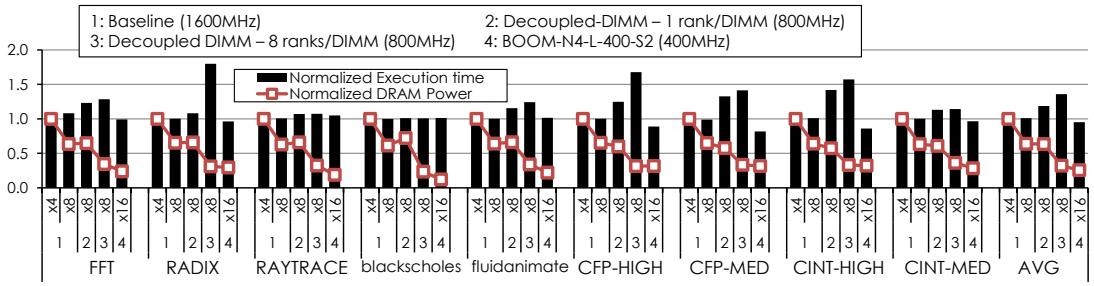


Figure 10: Comparing Decoupled DIMM with BOOM.

Because DRAM power occupies a large fraction of server power (30–57%) in high-end large memory servers, such dramatic reduction in DRAM power can leads to significant TCO savings.

`Blackscholes` is a compute-intensive workload with light off-chip traffic, so the majority of memory power is spent on static power (DRAM background and refresh and static power in the buffer chip). For high performance, we use only fast-exit power-down mode. Although the baseline power for `blackscholes` could have been lowered with deep power-down mode, the current results demonstrate the main advantage of LPDDR without requiring sophisticated power control policies.

DBUS/ABUS Utilization and Row-Buffer Locality: Figure 9 shows DBUS utilization, row-buffer hit rate, and address/command issue statistics.

Memory-intensive applications heavily utilize the external DBUS. `RADIX`, `CFP-HIGH`, `CFP-MED`, and `CINT-HIGH` use 80GB/s or higher bandwidth out of the 104.2GB/s peak bandwidth. `RAYTRACE`, on the other hand, lightly utilizes bandwidth (about 30GB/s) but is more sensitive to memory latency, leading to slight performance degradation in BOOM (never more than 5%). In all applications except `blackscholes`, DRAM row-buffer hit rate is low (only around 20%) as the many-core application's concurrent threads execution disrupts row-buffer locality [32]. Low row-buffer hit rate increases activate/precharge power with 4× configurations, although sub-ranking can mitigate this problem (Figure 8).

To understand ABUS bandwidth impact, we profile ABUS statistics and plot the number of commands issued per cycle in Figure 9. The BOOM architecture issues 1.1 commands per cycle on average, where support for high ABUS bandwidth is not critically needed.

Decoupled DIMM: *Decoupled DIMM* [38] is a recent proposal that allows low-frequency DRAM to save DRAM power. Decoupled DIMM uses the same data path in both internal and external buses – only frequencies are different. To keep the fast external bus busy, Decoupled DIMM relies on DIMM switching – every back-to-back request should be routed to different DIMMs. This scheduling constraint, however, incurs rank-to-rank switching penalty and can potentially degrade channel utilization.

To better understand the difference between BOOM and Decoupled DIMM, we compare the baseline chipkill-correct, BOOM-N4-L-400-S2, and Decoupled DIMM with 800MHz DRAM in Figure 10. We use two configurations of Decoupled DIMM (1 rank per DIMM and 8 ranks per DIMM) to see how the Decoupled DIMM's scheduling constraint affects performance.

As shown in Figure 10, Decoupled DIMM performs worse than both the baselines and BOOM due to the rank-to-rank-switching penalty for every back-to-back request. Decoupled DIMM with 1 rank/DIMM has relatively low performance degradation but requires one buffer chip per rank, limiting power saving with Decoupled DIMM. The BOOM architecture uses wide internal data path to avoid such rank switching overhead and achieves reduction in both execution time and power over the baseline and Decoupled DIMM.

BOOM with 64B Cache Line: BOOM-N4-L-400-S2 enables both 128B and 64B cache lines. To see the effects of smaller cache line size enabled by the BOOM architecture, Figure 11 compares BOOM-N4-L-400-S2 with 64B cache lines to the baseline ×4 (only the applications with noticeable differences are shown). BOOM with 64B achieves

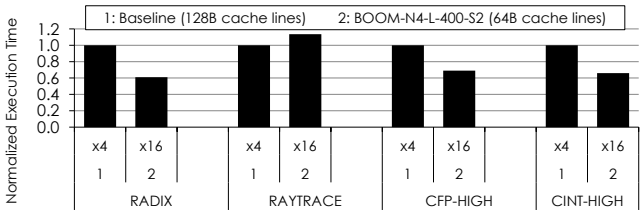


Figure 11: BOOM with 64B cache lines.

significant reduction in execution time in RADIX (40%), CFP−HIGH (31%), and CINT−HIGH (35%) . Such performance gains are primarily due to the use of a smaller cache line size (as discussed in adaptive granularity study by Yoon *et al*. [36]), especially for applications with low spatial locality. For other workloads, BOOM-N4-L-400-S2 with 64B cache line sometimes degrades performance (by 12% in RAYTRACE). Such results suggest an opportunity for workload-based adaptation of cache line size, statically or dynamically; and we leave it for future research.

6. Discussion

This section further discusses the specific ECC schemes for the BOOM architecture (Section 6.1), support for stronger memory protection (Section 6.2), and future memory system designs (Section 6.3).

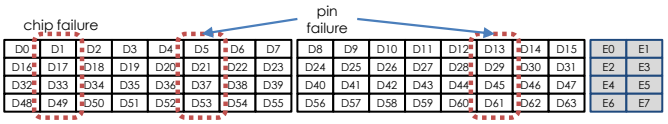
6.1. ECC for the BOOM Architecture

We use an RS code to implement chipkill-correct in the BOOM architecture. For simplicity, the example shown in Figure 12 uses 2× BOOM configuration with ×8 DRAM chips and burst 4 access, but the principles described here can be applicable to any BOOM configuration.

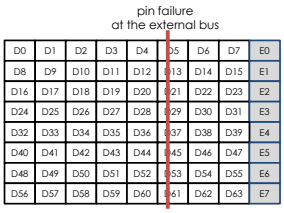
Figure 12(a) shows the conventional data and ECC layout for chipkill-correct. We construct a horizontal code word with 16 data symbols and 2 ECC symbols (e.g., D0–D15 and E0–E1). If a chip fails, it corrupts 4 symbols (e.g., D1, D17, D33, and D49) and the two-ECC-symbol RS code can correct the failure.

Unlike conventional systems, BOOM transfers a data block through the narrow external bus. If in any case a failure occurs at an I/O pin of the buffer chip or the external bus, it can affect many more symbols than a chip failure. Figure 12(b) illustrates how the symbols of Figure 12(a) are interleaved on the external bus in the BOOM architecture and which symbols are corrupted due to a pin failure on the external bus. As shown in Figure 12(a), a pin failure corrupts 8 symbols and the two-ECC-symbol RS code cannot correct this error.

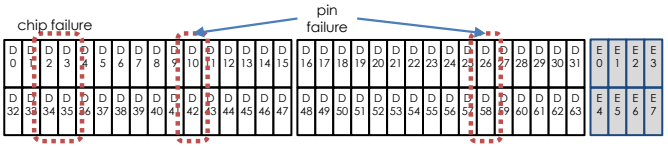
To tolerate both chip and pin failures in the BOOM architecture, we propose a new data / ECC layout (Figure 12(c)). We construct an 8-bit symbol out of 4 bits of burst 2, which does not change DRAM behavior since it conforms to DRAM burst length specifications: 4 in LPDDR2 and 8



(a) Data and ECC layout for conventional chipkill (in memory)



(b) Data transfer in BOOM (on the external bus)



(c) New ECC scheme that tolerates both chip and pin failures (in memory)

Figure 12: Data/ECC layout and data transfers on the external bus.

in DDR3. With the new data / ECC layout, a chip failure contaminates 4 symbols (e.g., D2, D3, D34, and D35), and a pin failure also corrupts 4 symbols (e.g., D10, D26, D42, and D58). Both appear as two symbol failures and can be tolerated using 4-ECC-symbol RS code per horizontal strip.

Furthermore, pin failures and chip failures corrupt different symbol combinations in this layout so that the MC can analyze corrupted symbol patterns (after correcting a failure) to take different actions for pin failures (e.g., to disable the failed physical channel) and chip failures (e.g., to request a DIMM replacement).

6.2. Strong reliability

Although we have only focused on low-power memory systems providing chipkill-correct so far, the BOOM architecture can enable even stronger reliability implementations. For example, a 4× BOOM configuration using ×4 and ×8 DDR3 DRAM provides 8 and 4 ECC chips per rank, respectively.

The 4× data path, however, increases the minimum access granularity to 256B in DDR3. Such a large cache block is undesirable in most systems. We can enable 128B data blocks by leveraging burst chop 4 (BC4) transfer, one of DDR3's less-known features. The BC4 in DDR3 allows BL 4 transfers, reducing memory access granularity but incurs a *dead* time period after a BL 4 transfer, leading to poor DBUS utilization. To further compensate for BC4's performance penalty, we use 800MHz data rates for the 4× DDR3 architecture. We do not show the detailed evaluation results, but our evaluation shows that the 4× BOOM architecture with ×4 and ×8 DDR3 DRAM has performance and power comparable to those of the baseline chipkill-correct.

Combining an RS code and the erasure technique, this architecture can tolerate up to 7 ×4 DRAM failures or 3 ×8

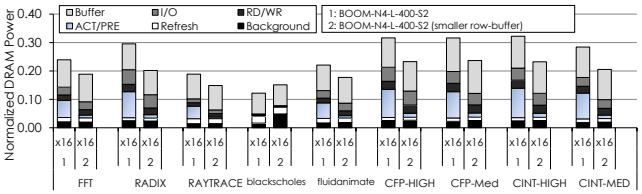


Figure 13: Power consumption using smaller row buffers.

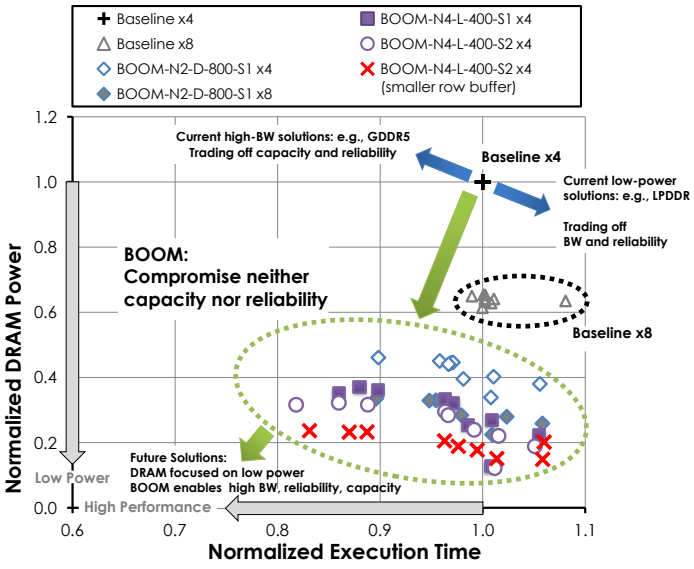


Figure 14: Performance and DRAM power.

DRAM failures, still at 12.5% ECC overhead. Note that today such stringent memory protection is supported only in high-end systems at the cost of much higher ECC overhead: spare memory and mirrored memory in HP servers [1] have up to 50% and 100% storage overhead, respectively; spare chips in Power 7 [16] incurs 25%; and parity channels in zEnterprise [15] requires more than 40% overhead. In comparison, BOOM can provide strong reliability level at low cost and low power, a unique advantage that is becoming particularly relevant as future systems integrate more and more components.

6.3. Opportunities and Implications for Future Memory Systems

Up to this point, we have focused only on commodity DRAM based memory system designs without changing the internals of the DRAM chip. Relaxing this limitation, we now discuss the potential direction and benefits of future server memory when DRAM chip redesigns are also considered.

The mainstream DRAM (e.g., DDRx and LPDDRx) has evolved by doubling per-pin data rates per generation. This direction, however, comes at the cost of increased design complexity, signal integrity issues, and higher power consumption (even after applying many low-power circuit techniques).

With BOOM as the enabling architecture, we outline a new direction in designing future memory *systems*. Rather than continually increasing DRAM frequency, DRAM devices should instead focus on power optimization. Using such low-power DRAM chips, the *system architecture* should focus on bandwidth, reliability and other system-level properties to satisfy user requirements. Such a combined, system-based approach can achieve much higher efficiency at lower cost. Below we first investigate the potential gains of changing DRAM internal designs and then present the directional change and efficiency improvement led by the new architecture.

The proposed BOOM architecture uses a wide internal data path to reduce DRAM clock frequency. While our evaluation shows significant background power reduction with low-speed DDR3 and LPDDR2, a naïve design suffers from overfetch-induced dynamic power increase, which is addressed by combining BOOM with sub-ranking (section 3.3). Following this direction, future designs should benefit from DRAM chip with reduced row-buffer size (also suggested in [32]).

We approximate the first-order effects of this change by reducing the baseline LPDDR2 DRAM's row-buffer size by a factor of 8, while keeping the other parameters unchanged. We scale I_{DD0} to 20% of the current LPDDR2 chip (adding a conservative margin over an 8× reduction). Enabled by BOOM, lowering device frequency saves the die area and design complexity previously devoted to support high-speed signaling (to implement complex transceivers and receivers), and such savings can now be used to implement smaller row-buffer size.

Applying the row-buffer size reduction to the BOOM-N4-L-400-S2 configuration, the performance remains almost unchanged. As shown in Figure 13, DRAM power for activate and precharge, the most power-consuming activities in BOOM, is reduced significantly. Combining the background and activate/precharge power reductions, the buffer chip power now dominates total DRAM power. Under this trend, low-power signaling techniques and low-power buffer chip designs will become more important.

`Blackscholes` is the only benchmark, where the power with smaller row-buffer configuration is slightly higher (only by 3%). This exception is mainly because `blackscholes`'s light memory traffic. With larger row-buffers, most accesses are routed to a few ranks, allowing other ranks to stay in power-down mode. With smaller row-buffers, however, accesses are spread across more ranks, reducing the opportunity to exploit power-down mode.

Figure 14 plots the normalized power and performance of various configurations. This chart clearly shows that the BOOM approach effectively reduces power, while maintaining performance. The figure suggests key directions for future designs: (1) use low frequency DRAM (e.g.,

200MHz) to save device power; (2) achieve high throughput with the proposed wide-internal data path; (3) use short burst (e.g., burst 2) to avoid increasing access granularity; and (4) reduce row-buffer size to lower dynamic power. Compared to current solutions that push on DRAM clock frequency using longer burst length and complex transceiver/receiver circuits, future memory can take a completely different direction to simultaneously improve performance and reliability while reducing power and cost.

7. Related Work

At a broader level, our proposal to enable low-power mobile components is similar to recent research on Atom/ARM-based serers [18, 12, 5, 8]. Our work is complementary and addresses server memory design, a subsystem of significant and growing importance. Our proposal is unique in that it can boost power efficiency, performance, reliability and capacity, all at the same time. In the memory system domain, a large body of work exists on designs and optimizations for power, performance and reliability.

R-DIMM and LR-DIMM: R-DIMMs and LR-DIMMs are two industry solutions to overcome the signal integrity issues and their limitations on capacity expansion. R-DIMMs buffer only ABUS signals, and LR-DIMMs completely isolate signals (both ABUS and DBUS), allowing an even larger number of DIMMs per channel than R-DIMMs.

Decoupled DIMM: Decoupled DIMM [38] allows low-frequency DRAM and reduces DRAM power. Decoupled DIMM is more effective when a DIMM has one or two ranks, but the performance degrades due to rank-to-rank-switching penalty.

Sub-Ranking: We employ sub-ranking technique for reducing dynamic power. Prior work in this category includes MC-DIMM [7], Mini-rank [37], module threading [33], and AGMS [36]. Unlike the prior work, a sub-rank in BOOM is very wide (128 bits), and chipkill-correct ECC overhead is relatively lower than prior proposal [7].

Systems with a Wide Slow Interface: Buffer on board (BoB) bridges 64-bit wide 1600MHz DRAM channels (relatively wide and slow) and a 16-bit wide 6.4GHz bus (narrow and fast) to the on-chip MC. Recently proposed wide I/O DRAM (a 512-bit wide bus at 200MHz) is another example that uses a wide slow interface to reduce power. None of these, however, consider power efficiency, performance, reliability, and capacity at the same time as in our work.

Chipkill-Correct: The wide channel interface with lock-step mode (commonly used in commercial chipkill implementations), together with long burst access in modern DRAM, increases access granularity and mandates large cache lines. Although traditional server applications have high spatial locality, many-core accesses can disrupt locality, and emerging applications (e.g., graph structures) demand high-throughput in fine-grained random access. Under such trends, large cache lines can become a critical hurdle in future servers.

Stronger Protection than Chipkill-Correct: Industry has already started to support even more stringent protection. A few examples are DDDC in Intel Itanium [9] (supports only $\times 4$), a spare DRAM chip in HP and IBM servers [1, 16], a parity channel in IBM zEnterprise mainframe [15], and mirrored memory in HP servers [1].

Chipkill for Wide DRAM Chips: Virtualized ECC [35] is a technique that stores ECC within the memory namespace and also enables wide $\times 8$ and $\times 16$ DRAM chips for chipkill-correct. This technique is orthogonal to our work, and the BOOM architecture can potentially also include the concept of virtualized ECC.

8. Conclusions

Emerging real-time data-centric applications represent a large and fast growing server market that demands high-capacity, high-bandwidth and high-reliability memory systems. The key challenge, however, is to deliver such solutions at low power and low cost.

In this paper, we propose and evaluate BOOM (Buffered Outputs on Module), a new memory architecture that can achieve the performance and reliability of server memory using low-power mobile DRAM devices.

We first identify the key contributors to low-power DRAM—wide data path and low frequency—and their implications on bandwidth, capacity and reliability. We then address these issues at the architectural level by using a simple buffer chip to bridge between a DIMM's slow wide internal data bus and its fast narrow external data bus. BOOM with sub-ranking mitigates the dynamic power increase due to overfetch, and the BOOM-specific data/ECC layout provides chipkill-correct or stronger protections at low ECC overhead. Overall, our evaluation of BOOM demonstrates on average 73% reduction of memory power and 5% performance improvement.

Furthermore, BOOM enables a new direction of memory system designs. With BOOM-optimized DRAM, which has row-buffer, low frequency, and short burst length, we envision a *device/system cooperative approach* to building future memory systems that simultaneously improve performance, power, cost, capacity and reliability. Future work will explore and evaluate such new designs using new data-centric workloads.

9. Acknowledgment

We thank the anonymous reviewers for their comments and suggestions. This research was partially supported by the Department of Energy under Award Number DE - SC0005026. See http://www.hpl.hp.com/DoE-Disclaimer.html for additional information.

References

[1] HP advanced memory protection technologies. ftp://ftp.hp.com/pub/c-products/servers/options/c00256943.pdf.

[2] HP power advisor. http://h18000.www1.hp.com/products/solutions/power/index.html.

[3] HP server memory. http://h18004.www1.hp.com/products/servers/options/memory-description.html.

[4] McSim. http://cal.snu.ac.kr/mediawiki/index.php/McSim.

[5] Project Moonshot. http://h17007.www1.hp.com/us/en/iss/110111.aspx.

[6] Calculating memory system power for DDR3. Technical Report TN-41-01, Micron Technology, 2007.

[7] J. H. Ahn, N. P. Jouppi, C. Kozyrakis, J. Leverich, and R. S. Schreiber. Future scaling of processor-memmory interfaces. In *Proc. the Int'l Conf. High Performance Computing, Networking, Storage and Analysis (SC)*, Nov. 2009.

[8] D. Anderson, J. Franklin, M. Kaminsky, A. Phanishayee, L. Tan, and V. Vasudevan. FAWN: A fast array of wimpy nodes. In *Proc. the 22nd ACM Symp. Operating Systems Principles (SOSP)*, Oct. 2090.

[9] R. Angy, E. DeLano, and M. Kumar. The Intel®Itanium®processor 9300 series: A technical overview for IT decision-makers. http://download.intel.com/pressroom/archive/reference/Tukwila_Whitepaper.pdf.

[10] S. Ankireddi and T. Chen. Challenges in thermal management of memory modules. http://electronics-cooling.com/html/2008_feb_a3.php.

[11] C. Bienia, S. Kumar, J. P. Singh, and K. Li. The PARSEC benchmark suite: Characterization and architectural implications. Technical Report TR-811-08, Princeton Univ., Jan. 2008.

[12] A. M. Caulfield, L. M. Grupp, and S. Swanson. Gordon: using flash memory to build fast, power-efficient clusters for data-intensive applications. In *Proc. the 14th Int'l Conf. Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, Mar. 2009.

[13] T. J. Dell. A white paper on the benefits of chipkill-correct ECC for PC server main memory. IBM Microelectronics Division, Nov. 1997.

[14] G. Hamerly, E. Perelman, J. Lau, and B. Calder. SimPoint 3.0: Faster and more flexible program analysis. In *Proc. the Workshop on Modeling, Benchmarking and Simulation (MoBS)*, Jun. 2005.

[15] D. Hayslett. System z redundant array of independent memory. http://www.thebrainhouse.ch/gse/doku/74_GSE/Silvios_Jukebox/System\%20z\%20Redundant\%20Array\%20of\%20Independent\%20Memory.pdf, 2011.

[16] D. Henderson, J. Mitchel, and G. Ahrens. Power7®system RAS: Key aspects of power systems^{TM} reliability, availability, and serviceability. http://www.essextec.com/assets/pdfs/powerraswp.pdf, 2010.

[17] P. Kogge et al. Exascale computing study: Technology challenges in achieving exascale systems.

[18] K. Lim, P. Ranganathan, J. Chang, C. Patel, T. Mudge, and S. Reinhardt. Understanding and designing new server architectures for emerging warehouse-computing environments. In *Proc. the 35th Ann. Int'l Symp. Computer Architecture (ISCA)*, Jun. 2008.

[19] C.-K. Luk, R. Cohn, R. Muth, H. Patil, A. Klauser, G. Lowney, S. Wallace, V. J. Reddi, and K. Hazelwood. PIN: Building customized program analysis tools with dynamic instrumentation. In *Proc. the ACM Conf. Programming Language Design and Implementation (PLDI)*, Jun. 2005.

[20] M. Meyer. The physics of data. http://www.parc.com/event/936/innovation-at-google.html.

[21] Micron Corp. *Micron 1 Gb* $\times$4, $\times$8, $\times$16, *DDR3 SDRAM*, 2006.

[22] Micron Corp. *Micron 2 Gb* $\times$4, $\times$8, $\times$16, *DDR3 SDRAM*, 2006.

[23] Micron Corp. *Micron 4 Gb* $\times$4, $\times$8, $\times$16, *DDR3 SDRAM*, 2009.

[24] Micron Corp. *Micron 2Gb* $\times$16, $\times$32, *Mobile LPDDR2 SDRAM*, 2010.

[25] J. A. Nerl, K. Pomaranski, G. Gostin, A. Walton, and D. Soper. System and method for controlling application of an error correction code. US Patent, US 7,437,651, Oct. 2004.

[26] J. A. Nerl, K. Pomaranski, G. Gostin, A. Walton, and D. Soper. System and method for applying error correction code (ECC) erasure mode and clearing recorded information from a page deallocation page. US Patent, US 7,313,749, Dec. 2007.

[27] J. Pouton, R. Palmer, A. M. Fuller, T. Greer, J. Eyles, W. J. Dally, and M. Horowitz. A 14-mW 6.25-Gb/s transceiver in 90-nm CMOS. *IEEE Journal of Solid-State Circuits*, 42(12), Dec. 2007.

[28] I. S. Reed and G. Solomon. Polynomial codes over certain finite fields. *J. Soc. for Industrial and Applied Math.*, 8:300–304, Jun. 1960.

[29] S. Rixner, W. J. Dally, U. J. Kapasi, P. R. Mattson, and J. D. Owens. Memory access scheduling. In *Proc. the 27th Ann. Int'l Symp. Computer Architecture (ISCA)*, Jun. 2000.

[30] B. Schroeder, E. Pinheiro, and W.-D. Weber. DRAM errors in the wild: A large-scale field study. In *Proc. the 11th Int'l Joint Conf. Measurement and Modeling of Computer Systems (SIGMETRICS)*, Jun. 2009.

[31] Standard Performance Evaluation Corporation. SPEC CPU 2006. http://www.spec.org/cpu2006/, 2006.

[32] A. N. Udipi, N. Muralimanohar, N. Chatterjee, R. Balasubramonian, A. Davis, and N. P. Jouppi. Rethinking DRAM design and organization for energy-constrined multi-cores. In *Proc. the Ann. Int'l Symp. Computer Architecture (ISCA)*, Jun. 2010.

[33] F. A. Ware and C. Hampel. Improving power and data efficiency with threaded memory modules. In *Proc. the Int'l Conf. Computer Design (ICCD)*, 2006.

[34] S. C. Woo, M. Ohara, E. Torrie, J. P. Singh, and A. Gupta. The SPLASH-2 programs: Characterization and methodological considerations. In *Proc. the 22nd Ann. Int'l Symp. Computer Architecture (ISCA)*, Jun. 1995.

[35] D. H. Yoon and M. Erez. Virtualized and flexible ECC for main memory. In *Proc. the 15th Int'l. Conf. Architectural Support for Programming Language and Operating Systems (ASPLOS)*, Mar. 2010.

[36] D. H. Yoon, M. K. Jeong, and M. Erez. Adaptive granularity memory systems: A tradeoff between storage efficiency and throughput. In *Proc. the Ann. Int'l Symp. Computer Architecture (ISCA)*, Jun. 2011.

[37] H. Zheng, J. Lin, Z. Zhang, E. Gorbatov, H. David, and Z. Zhu. Mini-rank: Adaptive DRAM architecture for improving memory power efficiency. In *Proc. the 41st IEEE/ACM Int'l Symp. Microarchitecture (MICRO)*, Nov. 2008.

[38] H. Zheng, J. Lin, Z. Zhang, and Z. Zhu. Decoupled DIMM: Building high-bandwidth memory systems using low-speed DRAM devices. In *Proc. the 36th Ann. Int'l Symp. Computer Architecture (ISCA)*, Jun. 2009.