Reducing Memory Access Latency with Asymmetric DRAM Bank Organizations

Young Hoon Son† Seongil O† Yuhwan Ro† Jae W. Lee‡ Jung Ho Ahn†

†Seoul National University
Seoul, Korea
{yhson96, swdfish, yuhwanro, gajh}@snu.ac.kr

‡Sungkyunkwan University
Suwon, Korea
jaewlee@skku.edu

ABSTRACT

DRAM has been a de facto standard for main memory, and advances in process technology have led to a rapid increase in its capacity and bandwidth. In contrast, its random access latency has remained relatively stagnant, as it is still around 100 CPU clock cycles. Modern computer systems rely on caches or other latency tolerance techniques to lower the average access latency. However, not all applications have ample parallelism or locality that would help hide or reduce the latency. Moreover, applications' demands for memory space continue to grow, while the capacity gap between last-level caches and main memory is unlikely to shrink. Consequently, reducing the main-memory latency is important for application performance. Unfortunately, previous proposals have not adequately addressed this problem, as they have focused only on improving the bandwidth and capacity or reduced the latency at the cost of significant area overhead.

We propose asymmetric DRAM bank organizations to reduce the average main-memory access latency. We first analyze the access and cycle times of a modern DRAM device to identify key delay components for latency reduction. Then we reorganize a subset of DRAM banks to reduce their access and cycle times by half with low area overhead. By synergistically combining these reorganized DRAM banks with support for non-uniform bank accesses, we introduce a novel DRAM bank organization with center high-aspect-ratio mats called CHARM. Experiments on a simulated chip-multiprocessor system show that CHARM improves both the instructions per cycle and system-wide energy-delay product up to 21% and 32%, respectively, with only a 3% increase in die area.

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

This material is based on work supported by Samsung Electronics. J. Ahn was supported in part by the Center for Integrated Smart Sensors funded by the Ministry of Education, Science and Technology of Korea as Global Frontier Project and by IDEC (EDA Tool). Jae W. Lee was supported in part by the Korean IT R&D program of MKE/KEIT KI001810041244.

ISCA'13 Tel-Aviv, Israel

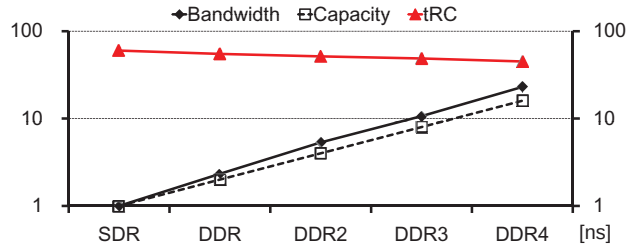


Figure 1: The capacity and bandwidth of DRAM devices have increased rapidly over time, but their latency has decreased much more slowly [41].

General Terms: Design, Experimentation, Performance

Keywords: DRAM, microarchitecture, asymmetric bank organizations, high-aspect-ratio mats

1. INTRODUCTION

DRAM has been a de facto standard for main memory for decades thanks to its high density and performance. DRAM has more than ten times higher storage density than SRAM and is orders of magnitude faster than NAND flash devices. With continued technology scaling, DRAM devices have evolved to exploit these smaller and faster transistors to increase mainly their capacity and bandwidth under tight cost constraints. To increase capacity, the DRAM cell size has been scaled down aggressively, and more cells share control and datapath wires. Meanwhile, DRAM arrays are divided into many subarrays, or *mats*, not to slow down those wires, and more bits are prefetched to improve bandwidth. However, the latency of DRAM devices, especially their random access time, has been reduced much more slowly. It is still around 50ns, which translates to approximately 100 CPU clock cycles (Figure 1).

Modern computer systems try to address this memory wall problem [51] with either latency tolerance techniques, such as out-of-order speculative execution [37], vector [36], stream [18], and massive multithreading [6, 28], or multi-level caches that lower the average memory access time. However, not all applications can be made insensitive to the main-memory latency because they often have insufficient parallelism or locality. Also, the memory footprints of popular applications [15] keep growing, and emerging applications, such as in-memory databases [1] and genome assemblies [3], demand even higher capacity. Besides, the gap in size between last-level caches and main memory has not been narrowed. As a result, lowering the main-memory la-

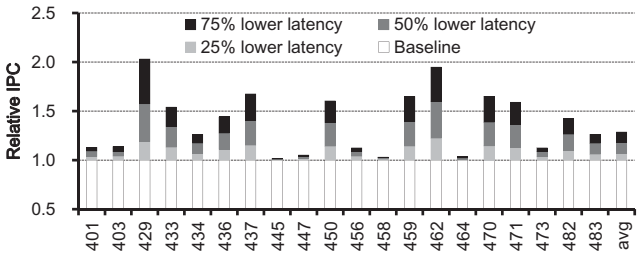


Figure 2: Many SPEC CPU2006 applications benefit from the reduction in memory access latency.

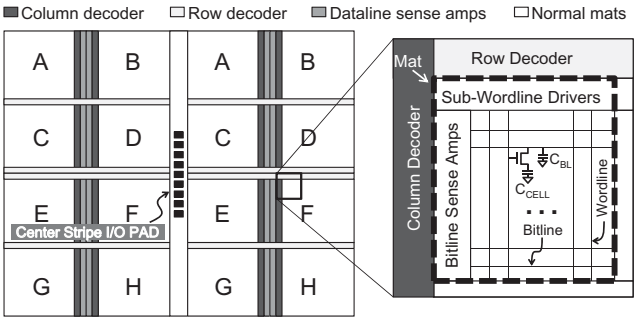


Figure 3: A typical modern main-memory DRAM organization with multiple banks and hierarchical structures.

tency would benefit many such applications. Figure 2 shows the relative instructions per cycle (IPC) of SPEC CPU2006 applications [15] when we lower the 28ns access time and the 48ns cycle time of DRAM by 25%, 50%, and 75% [1]. The degree of IPC improvement varies across applications, but an IPC improvement of more than 50% is observed for memory-intensive applications when we cut the DRAM latency by half without changing the bandwidth.

There have been several proposals for lowering the DRAM latency, including Reduced Latency DRAM (RLDRAM) [34], MoSys 1T-SRAM [13], and Fast Cycle DRAM (FCRAM) [42], but only with significantly increased die area. For example, an RLDRAM die is reported to be 40-80% larger than a comparable DDR2 DRAM die [20]. Alternatively, the ideas of embedding DRAM into processor dies [7], embedding SRAM into DRAM dies [56], or providing multiple row buffers per DRAM bank [14, 29] have been proposed, but they are more suitable for caches. Stacking DRAM dies on top of the processor die [46] can reduce main-memory access latency and power, as the physical distances and impedance between the dies are greatly reduced. However, this technique is mostly applied to embedded systems due to high heat density and limited scalability in capacity.

This paper proposes asymmetric DRAM bank organizations to reduce the average main-memory access latency. Through detailed analysis of the access and cycle times of a contemporary DRAM device, we identify that it is critical to reduce both datapath capacitance within mats and data transfer distance from I/Os to the mats. We reduce the former by making fewer DRAM cells share a datapath wire, or equivalently, by increasing the aspect ratio of a mat. We add extra datapath and control wires to enable non-uniform bank accesses, which shorten the data transfer distance to banks located close to I/Os at the center of a device. By synergistically combining these two aforementioned techniques, we devise a novel asymmetric DRAM bank organization with center high-aspect-ratio mats (CHARM). Exploiting the observation that a relatively small subset of memory pages are performance-critical [9, 39], CHARM places the low-latency high-aspect-ratio mats only for a small subset of banks at the center of the device and hence maintains low area overhead. Our evaluation demonstrates that CHARM improves system performance and energy efficiency for a variety of workloads with minimal area overhead. When CHARM DRAM devices with 2× higher aspect ratio mats are placed on a quarter of the DRAM banks at the center, our simulation results on SPEC CPU2006 benchmarks [15] show improvements in both the IPC and energy-delay product (EDP) by 4.4% and 7.6% on average, and up to 21% and 32%, respectively, with 3% area overhead. Multiprogrammed and multithreaded workloads also benefit from CHARM, depending on their bandwidth demands and latency sensitivities.

Our key findings and contributions regarding the asymmetric DRAM bank organizations are as follows:

- We present detailed breakdowns of the DRAM access and cycle times, through which we identify the key structures to reorganize within and outside of DRAM mats.
- We reduce the access and cycle times by increasing the aspect ratio of the mats and further improve the access time for local banks with better-than-worst-case delays.
- We propose CHARM, a practical solution to the latency problem of DRAM with minimal area overhead, which exploits the non-uniform criticality of memory accesses across the entire memory footprint.
- We quantify the system-wide benefits of CHARM in performance (IPC) and energy efficiency (EDP) using various workloads.

2. BACKGROUND

We first review the pertinent details of the organization and operations of contemporary DRAM devices to understand how various DRAM timing parameters influence the main-memory access latency. In so doing, we emphasize the importance of reducing both DRAM access time and cycle time to improve application performance.

2.1 Modern DRAM Device Organization

The continuing evolution of DRAM device organization has steadily increased its capacity and bandwidth even under tight cost constraints. A DRAM cell, which stores a bit of data, consists of a transistor and a capacitor. Multiple DRAM cells share control and datapath wires, called wordlines and bitlines, respectively, to improve area efficiency. In this 2D array structure, a row decoder specifies the wordline to drive, and a column decoder chooses one or more bitlines to transfer data to and from I/O pads. The wordlines and bitlines are made of metallic or polysilicon stripes to minimize the area overhead due to wiring. As the capacity of a device increases, the resistance and capacitance of these wordlines and bitlines also increase rapidly, leading to high

[1]We used the setup specified in Section 4 except that we used one memory controller.

Table 1: DDR3-1600 (1.25ns tCK) timing parameters [41]

Parameter	Symbol	Min (ns)
Activate to read delay	tRCD	13.75
Read to first data delay (= tCK×CL)	tAA	13.75
Access time (= tRCD + tAA)	tAC	27.5
Activate to precharge delay	tRAS	35
Precharge command period	tRP	13.75
Cycle time (= tRAS + tRP)	tRC	48.75
Read to read command delay	tCCD	5
Activate to activate command delay	tRRD	6
Four bank activate window	tFAW	30

access and cycle times. To address this problem, hierarchical structures [49] have been applied to the control and datapath wires such that an array is divided into multiple mats, where each mat has sub-wordline drivers and local bitline sense amplifiers. Global datalines connect the local sense amplifiers to the data I/Os. The global datalines also have sense amplifiers to increase transfer speed.

DRAM devices have adopted prefetching and multi-bank architectures [24] to improve the sequential and random-access bandwidth. Instead of increasing the internal operating frequency through deep pipelining (i.e., reducing tCCD), the DRAM mats transfer more bits in parallel through a wide datapath to keep up with ever-surging bandwidth demands. The transfer rate of a data I/O (2/tCK) is 8 times higher than the operating frequency of DDR3 [41] and DDR4 DRAM arrays. Hence, a DRAM array has 8 times more global datalines than the data I/O width (×N), which is called the prefetch size. This prefetching increases DRAM access granularity. Because a row in an array must be latched in the sense amplifiers before transferring data, it takes time to latch another row in the same array, which is defined as the cycle time (tRC). In modern DRAM devices, the cycle time (∼50ns in Table 1) is much longer than the reciprocal of the array's operating frequency (5ns in Table 1). For random accesses, the number of accesses to a specific row is at most a few; therefore, a mismatch in the tRC and tCCD leads to a poor performance. This problem is alleviated by having multiple banks where each bank is essentially an independent array, while sharing resources with others, such as I/Os, DLLs, and charge pumps. The multi-bank devices have inter-bank datalines to connect the global datalines of each bank to the data I/Os. Depending on target applications (e.g. graphics [8] and mobile [22, 33]), the designs of the I/O interfaces, the widths of the global datalines, or the transistor characteristics are modified, whereas the internal organization is mostly unchanged, as illustrated in Figure 3.

The capacity and bandwidth of DRAM devices have increased rapidly for years, but the latency, especially the cycle time has improved much more slowly. Figure 1 shows the relative capacity, bandwidth, and cycle time of multiple generations of DRAM devices. The DRAM cell size has been reduced from $8F^2$ to $6F^2$ and $4F^2$ [17], where F is the minimum wire pitch. The reduction in cell size, continuing evolution of process technology, and prefetching techniques all lead to an over 10× improvement in the capacity and bandwidth when we compare SDR and DDR3 DRAM devices. Meanwhile, the cycle time has improved much more slowly; tRC of DDR3 DRAM is still more than half that of SDR DRAM.

2.2 How DRAM Devices Operate

A memory controller manages DRAM devices under various timing constraints imposed by resource sharing, limitations on the operation speed and power, and the volatile nature of DRAM cells. The memory controller receives requests from various components of a processor, stores them into a request queue, and generates a sequence of commands to the attached DRAM devices to service the requests. The devices are grouped into one or more ranks, where all of the devices in a rank receive the same commands and operate in tandem. The ranks that share control and datapath I/Os compose a memory channel. A memory controller controls one or more channels.

Figure 4 shows a sequence of commands injected into a DRAM bank and the corresponding changes in the voltage of a bitline over a DRAM cycle time. When a bank is idle, bitlines and other datalines remain precharged and no data is latched in the bitline sense amplifiers. First, an activate command (ACT) arrives at the I/O pads and proceeds to the row decoder of the specified bank, where the decoder drives the specified wordline. On the mats that include the selected row, the access transistors controlled by the wordline are turned on and connect the DRAM cells to the bitlines in the mats. This charge sharing develops a voltage difference between the shared bitlines and the reference non-shared bitlines. Because the voltage difference is small due to the limited cell capacitance, sense amplifiers (often called row buffers) are used to speed up the voltage sensing for the cells whose voltage level is either VSS (zero) or VCC (one). Once the values are latched, column-level commands, such as read (RD) and write (WR), are applied to the open (activated) row and the specified address regions are accessed. The access time of a device (tAC) is the sum of the minimum ACT to RD/WR time (tRCD) and the RD command to the first data in the I/O time (tAA [2]).

Once the sense amplifiers latch the values, the bitlines are fully charged or discharged over time to store the latched data back into the cells in the open row. This process is known as the restore process. Then, the bank can accept a precharge command (PRE), which changes the voltage of the datapath wires back to (VCC + VSS)/2, the original voltage level, to access data in another row. The cycle time of a device (tRC) is the sum of the ACT to PRE time (tRAS) and the precharge time (tRP). A bank with an open row can read or write a batch of data (8×N bits for DDR3) on every tCCD, but only one bank can occupy the data I/Os of the device at any given time, which determines the device bandwidth. The capability of internal power delivery networks limits the number of the ACT commands that a device can process during the period of tFAW and tRRD. Table 1 summarizes the major timing constraints on a state-of-the-art DDR3 device.

2.3 The Impact of DRAM Timing Parameters on Memory Access Latency

All the DRAM timing parameters reviewed thus far affect how much time it takes for a memory controller to process an arriving request, while their relative impact depends on memory access patterns and access scheduling policies [40]. When a request arrives at the controller, if it has other pending requests and the scheduling policy determines to process

[2]tAA is the product of tCK and CAS latency (CL).

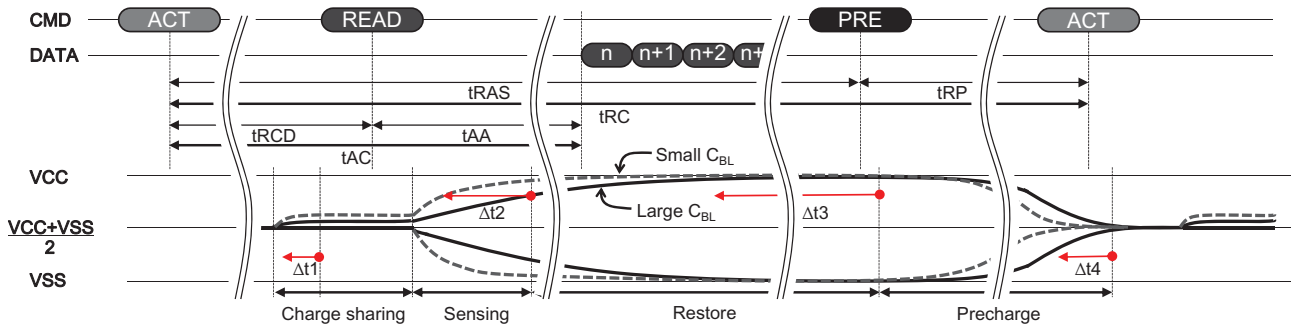


Figure 4: A sequence of DRAM commands and the corresponding changes in voltage of a bitline over a DRAM cycle.

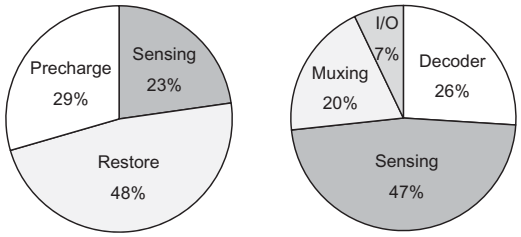
them first, the request experiences a queuing delay. The major parameters that influence the amount of this delay are tCCD and tRC. If the pending requests are well distributed across the banks or concentrated into a certain row in a bank, each request can be serviced at an interval of tCCD. Otherwise, tRC determines the minimum number of cycles needed to service any two requests that access the same bank but different rows in the bank. Besides, tFAW and tRRD may impose additional constraints for a controller operating with fewer DRAM ranks.

Once a request is ready to be serviced after experiencing an optional queuing delay, the memory controller generates one or more commands to process the request. If the bank targeted by the request has a row open and the row is different from the target of the request, the controller initially needs to wait until the bitline voltages are fully developed to either VCC or VSS (governed by tRAS), after which it precharges the corresponding bitlines (tRP), activates the target row (tRCD), and accesses the specified column (tAA). If the bitlines of the bank are already precharged or if the target row is already activated, the time to service the request can further be reduced. All of these facts show that both the access time and the cycle time of DRAM devices heavily influence main-memory access latency.

Each application has a different degree of memory-level parallelism (MLP), which determines its level of performance sensitivity on the DRAM timing parameters. The performance of the applications that have higher MLP, such as STREAM [32], typically depends less on main-memory access latency and more on the bandwidth (tCCD), while that of applications with lower MLP is often sensitive to the access latency, as evidenced in Figure 2, which shows the performance sensitivity of SPEC CPU2006 applications. Therefore, it is of great importance to devise DRAM microarchitectures that reduce both the access time and the cycle time of DRAM devices.

3. ASYMMETRIC DRAM BANK ORGANIZATIONS

In this section, we first identify the key contributors of the cycle and access times of a modern DRAM device. Then, we reduce its cycle time by decreasing the number of wordlines per mat, which also lowers the access time and power dissipation at the cost of an area increase. Based on the observation that the main-memory accesses of many applications are not uniformly distributed over their memory footprints, we alleviate the area overhead by locating mats with different aspect ratios together and further reduce the access time on a portion of the device by placing mats with high aspect ratios close to the I/Os.



(a) tRC (cycle time) (b) tAC (access time)

Figure 5: The DRAM timing breakdown of (a) tRC and (b) tAC.

3.1 DRAM Cycle and Access Time Analysis

To devise techniques to lower memory access latency, we first analyze the cycle and access times of a contemporary DRAM device to understand their key delay components. We use a 4Gb Samsung DDR3 device in a 28nm process technology [41] as a reference chip with the following assumptions: (1) each bitline is connected to 512 cells; (2) each mat has 512 bitlines; (3) the cell array width of a mat is 9 times the width of a bitline sense amplifier; (4) the bitline capacitance is 6 times higher than the cell capacitance based on recently reported values [16, 52]. The simulated 8Gb device, which serves as baseline organization, has 8 banks and follows the DDR3 specification. We use the PTM low power model [57] for SPICE simulation.

The simulation results show that the critical path of the cycle time (Figure 5(a)) is composed of sensing, restore, and precharge processes, all of which only depend on the structures within a mat. Moreover, all the processes involve manipulation of bitline voltages so that managing the capacitance and resistance of the bitline heavily influences the cycle time. The access time (Figure 5(b)) is affected not only by the bitline sensing time (tRCD) but also by the address decoding process, transfer and rate conversion times through datalines and multiplexers (muxing), and I/O driving time. Both the address decoding process and dataline transfer time depend on the physical distance of control and datapath wires between individual mats and I/Os. As a result, it is necessary to reduce the time taken both within and outside of mats.

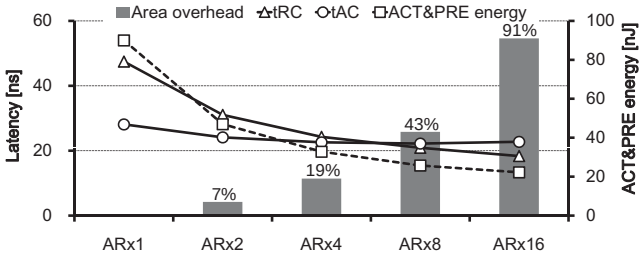


Figure 6: The area overhead, tRC, tAC, and activate and precharge energy of a DRAM device as the number of wordlines per mat increases.

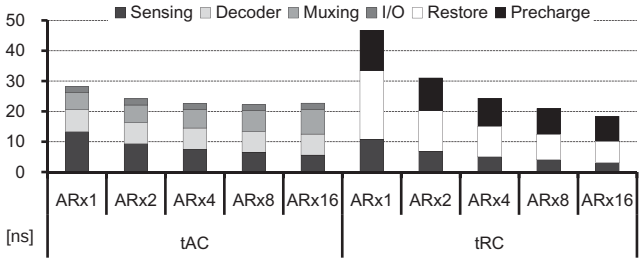


Figure 7: The breakdowns of tAC and tRC over various numbers of wordlines per mat. Fewer wordlines per mat lower sensing and precharge delays significantly first, but we then see diminishing returns.

3.2 Low-Latency Mats with High Aspect Ratios

We first focus on reducing the time to load and store data within mats. The analysis in Section 3.1 shows that the sensing, restore, and precharge processes are all sensitive to the local bitline capacitance. Hence, we propose to decrease the number of wordlines per mat, which makes fewer DRAM cells attached to a bitline and reduces the bitline capacitance. Figure 4 illustrates the effects of the reduced bitline capacitance. The amount of voltage developed by the charge sharing between a bitline and a cell increases as the bitline capacitance decreases, thus reducing the sensing time (Δt1 and Δt2). The bitline capacitance is a significant portion of the load capacitance of the sense amplifiers and the precharge drivers in a mat, so both the cell restore time (Δt3) and the precharge time (Δt4) decrease. Note that decreasing the number of bitlines per mat has little influence on the sensing, restore, and precharge time and is not further explored in the paper.

There are two main sources of overhead when decreasing the number of wordlines per mat. First, a bank needs more mats to keep its capacity constant as each mat has fewer wordlines. Because a mat has the same number of sense amplifiers regardless of the number of wordlines, having more mats incurs area overhead. Second, a global dataline within a bank becomes longer and is connected to more mats. This increases the fanin and junction capacitance of the global dataline. However, the impact of this increase in capacitance on the access latency is limited because global datalines, which are made of metal, have much lower capacitance and resistance than bitline wires.

Figures 6 and 7 present the SPICE simulation results of the area overhead, tRC, tAC, and the activate and precharge energy of the device with varying numbers of wordlines per mat. The reference mat has 512 bitlines and 512 wordlines. Therefore, decreasing the number of wordlines is equivalent to increasing the aspect ratio of the mat. We use the notation AR×N to denote a mat with aspect ratio N. A mat with 128 wordlines (AR×4) has half the cycle time of the reference mat, while further quadrupling the aspect ratio results in additional 10% reduction. Increasing the aspect ratio also reduces the activate and precharge energy of the mat. Figure 6 shows that a mat with 128 wordlines requires 33% less activate and precharge energy compared to the reference mat. In contrast, increasing the aspect ratio does not improve tAC much because the sensing process becomes faster but the decoding speed is mostly unchanged while the datapath delay between mats and I/Os becomes worse, as shown in Figure 7. Also, the area increases rapidly with more mats per bank (Figure 6). Using four times more mats incurs an area overhead of 19%, and the bank becomes almost twice as large with 16 times more mats. Because the cost of DRAM devices is very sensitive to the area, we need to devise microarchitectures that limit the area overhead and further decrease the DRAM access time.

3.3 Banks with a Non-Uniform Access Time

Because decreasing the number of wordlines per mat has a limited impact on the DRAM access time, it is necessary to decrease the physical distance between the banks and I/Os to reduce the access time further. We leverage the idea of non-uniform cache architecture [21] and propose a multi-bank organization with a non-uniform access time in which the control and data transfer time between a bank and I/Os depends on the location of the bank within a device. Here we assume that the I/Os are located at the center of the device without a loss of generality.

To improve the random access performance, many modern main-memory DRAM devices have 8 or 16 banks. A bank is typically implemented with one or more blocks. Each block has a row decoder and a column decoder to process the assigned request. Existing DRAM specifications, such as DDR3 and DDR4, assume a single, uniform access time to all the banks. Therefore, devices complying with the specifications are designed to improve the area efficiency under the constraint that all of the banks have the same transfer time regardless of their locations. For example, a split-bank architecture is applied to the baseline DDR3 DRAM organization shown in Figure 3 and Figure 8(a). Either side of a device has 8 blocks, each of which corresponds to half of a bank. Each pair of blocks operates in tandem such that one of the blocks takes charge of half of the data being transferred. This can reduce the number of inter-bank datalines by half compared to the organization that places an entire bank on a single side.

We reduce the access time to the blocks located at the center of a device by relocating the column decoders to the center stripe and by making a single block, not a pair of blocks, take charge of a data transfer instance, as shown in Figure 8(b). This enables the center blocks (e.g., Block **D'**) to start decoding column addresses much earlier than the corner blocks (e.g., Block **B**). We also group the blocks of a bank together on the same side such that both blocks share the same dataline sense amplifiers, which keeps the number of dataline sense amplifiers needed in a device unchanged. Still, the number of inter-bank datalines is doubled com-

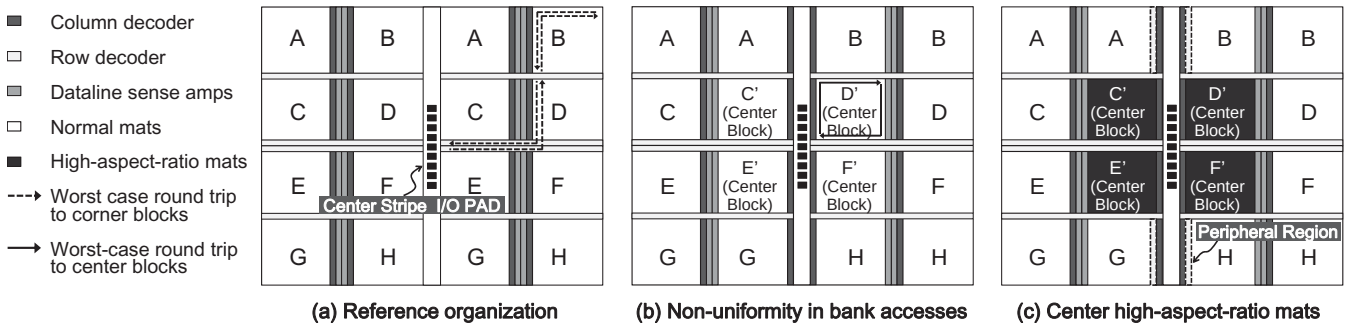


Figure 8: A DRAM device with (a) the reference structure is reorganized to (b) support non-uniformity in bank accesses and (c) replace the mats in the center blocks with high-aspect-ratio mats (CHARM).

pared to the split-bank architecture. These changes make the command path and inter-bank datalines of the 4 center blocks (the solid arrows in Figure 8(b)) become shorter than those of the 4 blocks at the corner (the dotted arrows in Figure 8(a)) due to the perimeter of a block in a round trip. The SPICE simulation results show that the access time of 4 center blocks is 6ns lower than that of the other blocks. The layout overhead of the relocated column decoders and the additional inter-bank dataline wires needed for data transfers increase the device area by 1%. In this paper, we assume that the corner blocks and the remaining blocks at the edges have the same access latency and leave the exploration of three-tier or more bank organizations for future work.

3.4 CHARM: Center High-Aspect-Ratio Mats

We synergistically combine the ideas of increasing the aspect ratio of the mats and introducing non-uniformity in bank accesses to further decrease average access and cycle times of DRAM devices with low area overhead. Replacing all of the mats in a device with high-aspect-ratio (HAR) mats incurs high area overhead, as explained in Section 3.2. Instead, we only use HAR mats in the center blocks of the device as shown in Figure 8(c). We call this DRAM microarchitecture CHARM, Center High-Aspect-Ratio Mats. The CHARM microarchitecture can cut both the access time and the cycle time of the center HAR mats down to at least half of the values of the remaining mats with a normal aspect ratio while limiting the area increase rate to a single digit percentage. Because a DRAM block with HAR mats is larger than a block with only normal aspect ratio mats when both blocks have the same capacity, the blocks in the middle columns of the device are misaligned. This spare area can be easily filled with peripheral resources, such as charge-pump circuits, decoupling capacitors, and repeaters for inter-bank datalines, which further reduces the area overhead.

We compare the access time and the relative area of the CHARM and other organizations in Figure 9. The organizations that have the HAR mats in all of the blocks and uniformly access them (AR×N in the figure) has a single access time. The access time of an AR×N (N > 1) organization is lower than that of the reference organization (AR×1) mainly due to the decrease in the activate time (tRCD). The read-to-first-data delay (tAA) actually increases as the aspect ratio (N) increases because the device gets larger and more mats are connected to global datalines. We use the notation CHARM[×N,/M] for a CHARM DRAM that po-

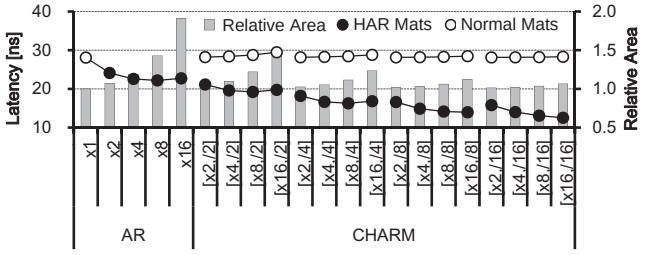


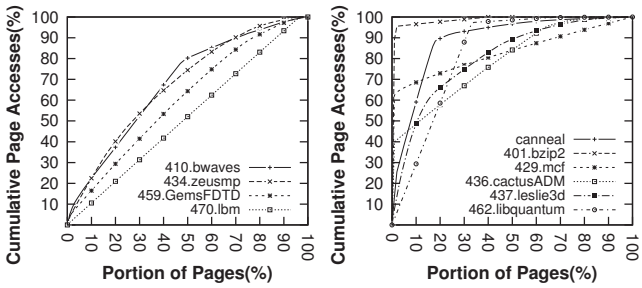
Figure 9: The access time to the center HAR mats and the remaining normal mats and the area overhead of various CHARM DRAM organizations. The reference organization has no HAR mats and accesses all the banks with uniform latency.

sitions the HAR mats with the aspect ratio N only in the center blocks that encompass the one M-th of the device capacity. It alleviates the increase of tAA of the normal mats. As M increases, the tAA value of the normal mats becomes closer to that of the AR×1 and tAA of the HAR mats improves further. The access time of the HAR mats for CHARM[×4,/8] becomes 15ns, almost half the access time of the reference organization. Further increasing M improves the tAA of the HAR mats further, but with diminishing returns. The area overhead of CHARM[×N,/M] is lower than that of AR×N as well because the HAR mats exist only at the center blocks. For example, the area overhead of AR×4 is 19%, while the corresponding percentages of CHARM[×4,/4] and CHARM[×2,/4] are only 6% and 3%, respectively. Note that the cycle time of a mat only depends on the aspect ratio of the mat. This is not shown in the figure.

The analysis thus far shows that CHARM DRAM organizations with center high-aspect-ratio mats can significantly improve the average cycle and access time with minimal area overhead. We quantitatively compare the system-level performance and energy efficiency of CHARM and the other DRAM organizations using popular single- and multi-threaded benchmark suites in Section 5.

3.5 OS Page Allocation

The non-uniform bank organization becomes more effective if we can assign performance-critical data to lower-latency blocks. Previous work suggests that a relatively small subset of pages is performance-critical [9, 39]. The performance criticality of an OS page can be estimated by



(a) Accesses more uniformly distributed over pages **(b) Accesses less uniformly distributed over pages**

Figure 10: Cumulative page accesses over the portion of touched pages on SPEC CPU2006 applications with high L2 cache MPKI and canneal from PARSEC where pages are sorted by the access frequency.

(a combination of) various metrics, such as the access frequency and recency [25], the cache miss rate [9], and the TLB miss rate [48]. Note that this problem is nothing new; similar problems appear in other non-uniform memory systems such as distributed non-uniform memory access (NUMA) machines [48], hybrid memory systems [39], adaptive granularity memory systems [55], as well as conventional demand paging systems.

For the rest of this paper, we use access frequency as page criticality predictor due to its ease of implementation. Exploiting the non-uniformity of access frequencies across the entire memory footprint, we assign frequently-accessed pages to low-latency blocks in CHARM DRAM devices. The page access frequency can be easily measured via profiling. Note that this work primarily focuses on novel DRAM bank organizations and their performance potentials and that we leave more sophisticated page-allocation mechanisms for future work.

Figure 10 offers an evidence of the existence of the aforementioned access non-uniformity in SPEC CPU2006 applications [15]. The figure shows the cumulative memory access frequency over the portion of the accessed pages sorted according to the access frequency. Some applications access pages relatively evenly such that their cumulative curves closely resemble a diagonal line (Figure 10(a)). They often repeat sweeping large blocks of memory rapidly (470.lbm and 459.GemsFDTD, for example) or access random locations (RADIX in SPLASH-2 [50]). In contrast, for other applications such as 429.mcf, 437.leslie3d, and 462.libquantum, relatively small portions of pages account for most of the page accesses (Figure 10(b)). These applications can benefit greatly from the proposed non-uniform bank organization approach.

The page placement decision can be made statically (e.g., compiler profiling, programmer annotations) or dynamically (e.g., OS tracking per-page memory access frequencies), possibly assisted by hardware support. A new memory allocation system call that maintains a separate free list for low-latency page frames is necessary, which may also migrate data between low- and high-latency blocks as needed. In this paper, we take a static approach via off-line profiling to identify the most-frequently accessed data regions. We envision this process to be automated using compiler passes targeting the new system call. Automating profile-guided page allocation and evaluating OS-managed dynamic allocation are outside the scope of this paper. We evaluate the performance impact of allocating only a portion of frequently accessed pages to low-latency blocks in Section 5.

Table 2: Power and timing parameters of the representative DRAM organizations.

Parameter	Reference (AR×1)	CHARM[×2,/4] HAR mats	CHARM[×2,/4] Normal mats
tRCD	14ns	10ns	14ns
tAA	14ns	8ns	14ns
tRAS	35ns	20ns	35ns
tRP	14ns	11ns	14ns
ACT+PRE energy	90nJ	47nJ	90nJ
RD energy	15.5nJ	13.8nJ	15.7nJ
WR energy	16.5nJ	14.7nJ	16.7nJ

4. EXPERIMENTAL SETUP

We modeled a chip-multiprocessor system with multiple memory channels to evaluate the system-level impact of the DRAM devices with HAR mats and asymmetry in bank accesses on performance and energy efficiency. The system has 16 out-of-order cores. Each core operates at 3 GHz, issues and commits up to 4 instructions per cycle, has 64 reorder buffer entries, and has separate L1 instruction and data caches and a combined L2 cache. The size and associativity of each L1 cache and L2 cache are 16 KB and 4, and 512 KB and 16, respectively. Each cache has 4 banks with a line size of 64 B. A MESI protocol is used for cache coherency and a reverse directory is associated with each memory controller. The system has 8 memory controllers unless mentioned otherwise, while each controller has one memory channel. There are two dual-inline memory modules (DIMMs) per channel and each DIMM consists of 2 ranks. Each rank has 9 8Gb ×8 DDR3-2000 DRAM devices, 8 for data and 1 for ECC, making the peak memory bandwidth of the system with 8 memory controllers 128 GB/s. Each memory controller uses the PAR-BS [35] and the open-row [19] policy for access scheduling and has 64 request queue entries.

We slightly modified the memory access scheduler to give equal priority to the requests targeting both the center HAR mats and the other normal mats. Row-level commands, such as ACT and PRE, change the status within the mats and only share the command path, which is sufficient to make the scheduler apply proper timing constraints depending on the mat type. Column-level commands share the command and datapath I/Os. The scheduler first assumes that even the center HAR mats have the tAA of the normal mats, after which it finds the request that has the highest priority and meets all the timing and resource constraints. When a RD or WR command to a center HAR mat is generated by the scheduler, it also checks whether the data can be transferred into the slot specified by the original tAA of the center HAR mat and uses the slot if available.

McSimA+ [5] was used for simulation. We obtained the power, area, and timing models of out-of-order cores and caches using McPAT [27]. The power and timing values of the physical interfaces between the processor and DRAM devices and their internal components are based on the DDR3 specifications [41] and the SPICE modeling results described

Table 3: We categorized the SPEC CPU2006 applications into 3 groups depending on their L2 cache MPKI values.

Group	SPEC CPU2006 applications
spec-high	429.mcf, 433.milc, 437.leslie3d, 450.soplex, 459.GemsFDTD, 462.libquantum, 470.lbm, 471.omnetpp, 482.sphinx3
spec-med	401.bzip2, 403.gcc, 410.bwaves, 434.zeusmp, 435.gromacs, 436.cactusADM, 464.h264ref, 473.astar, 481.wrf, 483.xalancbmk
spec-low	400.perlbench, 416.gamess, 444.namd, 445.gobmk, 447.dealII, 453.povray, 454.calculix, 456.hmmer, 458.sjeng, 465.tonto

in Section 3.1. The key timing and power parameters we obtained from the modeling in Section 3 on the representative DRAM organizations are summarized in Table 2. Note that if the increase in tAA on normal mats is smaller than tCK, we increase the size of drivers in inter-bank datalines to negate the increase in tAA. This increases the energy for read and write operations slightly, which is properly modeled and shown in the table.

We used the SPEC CPU2006 [15], SPLASH-2 [50], and PARSEC [10] benchmark suites to evaluate the efficiency of the CHARM DRAM-based main memory system. We ran Simpoint [43] to find the representative phases of the SPEC CPU2006 applications, and selected the 4 slices with the highest weights per application. Each slice consists of 100 million instructions. We simulated regions of interest for SPLASH-2 and PARSEC. We used reference datasets for SPEC CPU2006, simlarge datasets for PARSEC, and the datasets listed in [27] for SPLASH-2. We classified SPEC CPU2006 applications into three groups based on L2 cache misses per kilo-instructions (MPKI) [15]. These are called spec-high, spec-med, and spec-low, as shown in Table 3. We created four mixtures of multiprogrammed workloads, one from each group and one from all three groups. The latter is called spec-blend and consists of five applications from spec-high, six from spec-med, and five from spec-low. For each multiprogrammed mixture, a simulation point is assigned to each core, and one or two highest weight points are used per application. We use a weighted speedup approach [44] to compare the performance of multiprogrammed workloads. We compute the weighted speedup by initially determining the relative performance of an application, which is the ratio of its IPC in a multiprogrammed environment to IPC in a standalone execution environment, and then aggregating the relative performance of all applications in the mixture.

5. EVALUATION

We evaluate the system-level impacts of CHARM on the performance and energy efficiency of a contemporary multicore system with a variety of workloads. We first execute SPEC CPU2006 programs to show how much improvement in instructions per cycle (IPC) and energy-delay product (EDP) is achievable with CHARM on single-threaded applications. Then, we run multiprogrammed and multithreaded workloads to evaluate the efficacy of CHARM DRAMs in scenarios in which main-memory devices serve accesses from multiple requesters with and without correlation.

5.1 Performance Impact on Single-Threaded Applications

Figure 11 shows scatter plots of the relative area (x-axis) versus the relative IPC or EDP (y-axis) for various DRAM organizations on single-threaded SPEC CPU2006 benchmark programs. The organization with normal mats and uniform accesses for all of the banks (AR×1) is the reference. Due to limited space, we only present the average values over all the programs (Figure 11(b) and Figure 11(d)) and the average values over those with high L2 cache MPKI, categorized as spec-high in Table 3 (Figure 11(a) and Figure 11(c)). For this experiment, we use only one memory controller to stress memory system bandwidth.

We make the following observations from the experimental results. First, increasing the aspect ratio of all the mats in a device improves the IPC (higher is better), but increasing it by more than twice the original value yields only marginal improvements in IPC. We define the return on investment (ROI) of IPC as the gain in IPC over the area overhead such that the gradients of the virtual lines between the points and the origin in the figures become the ROIs. Figure 11(b) shows that AR×2 improves the average IPC of all the SPEC CPU2006 applications by 4.0%, while AR×4 and AR×8 improve it by 4.5% and 5.0%, respectively. For applications with low intra-page spatial locality (i.e., with high ratio of row-level commands to column-level commands), such as 429.mcf and 436.cactusADM, increasing the aspect ratio improves performance because they benefit from lower tRC. However, for applications with high spatial locality, such as 437.leslie3d and 482.sphinx3, tAA influences more on performance than tRC. Increasing the aspect ratio in fact increases tAA, making those applications perform worse. Considering that AR×4 (19%) and AR×8 (43%) have much higher area overhead than AR×2 (7%), AR×2 has a higher ROI and can be regarded as a more effective configuration.

Second, CHARM DRAM devices have lower area overhead but its performance gain is affected by how effectively an application utilizes the center HAR mats. CHARM organizations with main-memory access frequency oblivious mapping (configurations with suffix O in Figure 11) result in smaller performance improvement than AR×N. For the CHARM[×N,/M]O, only one M-th of memory accesses head to the center HAR mats on average when the oblivious mapping is used. When M is small, a large portion of data are allocated to the HAR mats, but the access time to the center HAR mats also increases and becomes closer to that of AR×N. As M increases, the access time to the center HAR mats becomes much lower than that of AR×N, but there is a fixed area overhead of implementing asymmetric bank accesses and the fewer memory accesses utilize the HAR mats. These trends make the ROI of CHARM[×N,/M]O comparable to that of AR×N. The organizations with access frequency aware mapping (ones with suffix A in Figure 11) yield substantially higher performance gain on average without any further area overhead. The IPCs of CHARM[×2,/2]A, CHARM[×2,/4]A, CHARM[×4,/2]A, and CHARM[×4,/4]A are all higher than the IPC of AR×8, which provides the highest performance with uniform bank accesses, on spec-high applications. The observation made above also holds here such that CHARM[×2,/M]A has a higher ROI than CHARM[×4,/M]A. Among those, CHARM[×2,/4]A gives the highest ROI with an 11% IPC improvement and a 3% area increase. Hereafter, we use it as default configuration.

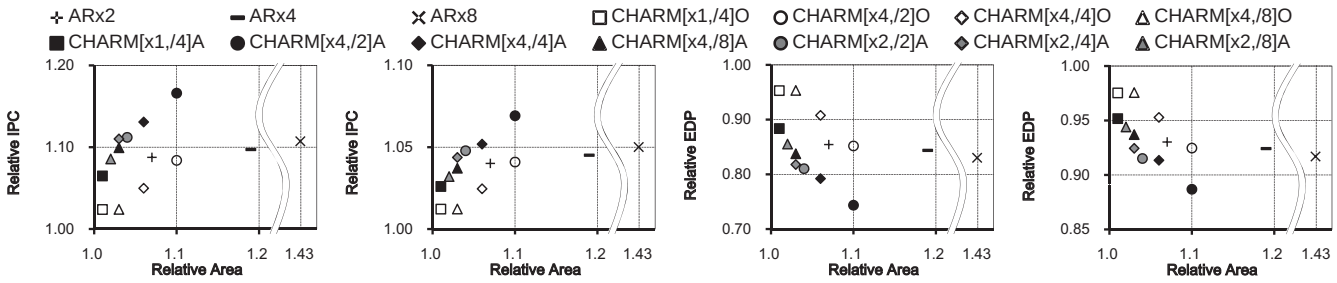


Figure 11: Scatter plots of the relative area and the relative IPC of the various DRAM organizations on (a) the average of SPEC CPU2006 applications with high main-memory bandwidth (spec-high in Table 3) and (b) the average of all SPEC CPU2006 applications (spec-all), and the relative area and the relative EDP on (c) the average of spec-high and (d) the average of spec-all.

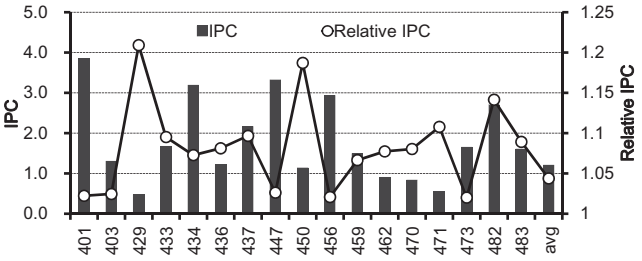


Figure 12: The absolute and relative IPCs of SPEC CPU2006 applications for CHARM[×2,/4]A. The applications with low memory bandwidth demand are omitted in the graph, but included when the average values are computed.

Third, more performance gains are generally observed for applications with higher main-memory bandwidth demands. 429.mcf and 450.soplex, two of the top three bandwidth-demanding applications, provide the highest performance gains with CHARM[×2,/4]A, showing 21% and 19% increases in IPC, respectively. If we take the average IPC of the 9 SPEC CPU2006 applications with high L2 MPKI values, the IPC gain for CHARM[×2,/4]A is 11% higher than the average IPC gain for the same configuration over the entire CPU2006 applications. Comparison of Figure 11(a) and Figure 11(b) shows that the relative order of the ROI values among the DRAM organizations is mostly preserved even though we only take the average of spec-high applications. Figure 12 shows the absolute and relative IPCs of SPEC CPU2006 applications for CHARM[×2,/4]A.

Fourth, the EDP (lower is better) and IPC values of the various DRAM organizations show similar trends. The relative order of the ROI values is preserved. Because the CHARM DRAM devices improve the performance of the applications and consume less activate and precharge energy, the absolute ROI value in EDP is higher than that in IPC on the same configuration. On CHARM[×2,/4]A, the EDP is improved by an average of 7.6% and 18% on all the SPEC applications and the spec-high applications, respectively.

5.2 Performance Impact on Multiprogrammed Workloads

The CHARM organization also improves the performance of systems running multiprogrammed workloads. Figure 13 shows the weighted speedups [44] of the three mixtures described in Section 4 on systems using CHARM[×2,/4] and CHARM[×4,/4]. We do not present the spec-low results because the aggregate main-memory bandwidth from the applications is too low to make a noticeable difference in the weighted speedups. We test three interesting memory-provisioning schemes for each mixture. These are characterized as follows: 1) each application is provisioned with the same amount of memory space, which equals the maximum footprint of the application in the mixture, with a quarter of that allocated to CHARM (Scheme 1); 2) each application is provisioned with the same footprint, which equals the average footprint of all applications in the mixture, with a quarter of that allocated to CHARM (Scheme 2); and 3) a quarter of the memory footprint of each application is allocated to CHARM (Scheme 3). We use the weighted speedups of the AR×1 configuration as references and apply the access-frequency-aware memory allocation scheme discussed in Section 3.5.

We make the following observations from the experiment. First, spec-high benefits more (i.e., better relative weighted speedups) from the CHARM DRAM devices compared to the other mixtures because spec-high applications are more sensitive to the main-memory access latency than spec-med and spec-blend applications. However, spec-high yields the smallest absolute weighted speedup values, as memory requests from the applications often contend with each other in the memory controllers and because each application experiences a higher average memory access time than when executed in isolation. The weighted speedup values are improved by 7.1% and 12% for spec-high on CHARM[×2,/4] and CHARM[×4,/4], respectively.

Second, among the three memory-provisioning schemes, the one that assigns the maximum memory footprint among all of the applications in the mixture to each application (Scheme 1) provides the greatest weighted speedup improvement, as a larger memory footprint is allocated to the center HAR mats in Scheme 1 compared to Schemes 2 and 3. Nonetheless, all the scenarios provide noticeable improvements in the weighted speedup and show similar trends.

Third, the relative weighted speedup improves as more main-memory bandwidth is provided in the system for mixtures with bandwidth-demanding applications. We change the number of memory controllers and present the weighted speedups of Scheme 1 in Figure 13(b). Because memory

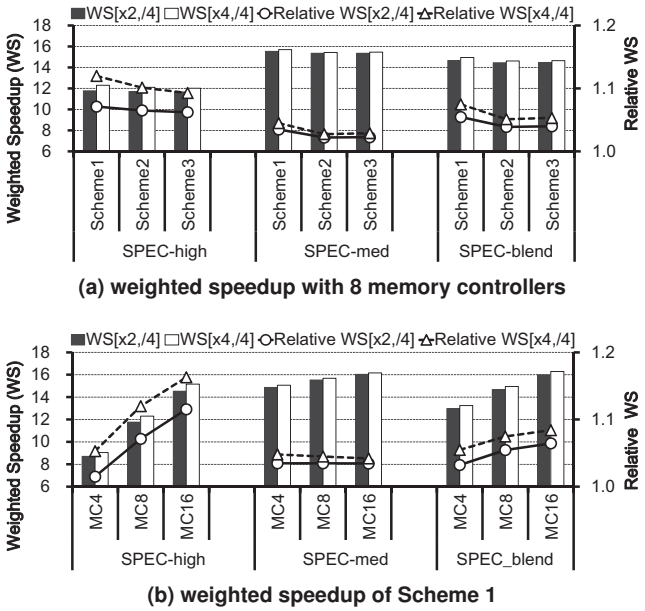


Figure 13: The absolute and relative weighted speedups of spec-high, spec-med, and spec-blend on the systems using CHARM[×2,/4] and CHARM[×4,/4] with (a) 8 memory controllers. We vary the number of controllers and present the weight speedups of Scheme 1 in (b).

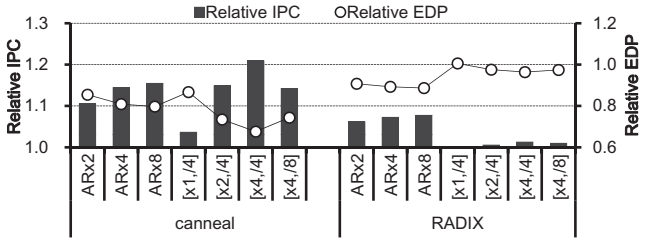


Figure 14: The relative IPC and EDP of the various DRAM organizations on canneal and RADIX applications. AR×1 is the reference organization.

requests are almost always piled up in the request queues of the spec-high memory controllers, the bandwidth of the main-memory system is the most prominent factor affecting system performance. As a result, both the absolute and relative weighted speedups of spec-high are more sensitive to the main-memory bandwidth than the others. These findings demonstrate the effectiveness of CHARM for multiprogrammed workloads.

5.3 Performance Impact on Multithreaded Workloads

The CHARM DRAM devices also improve the performance and energy efficiency of the simulated system on multithreaded applications, while the degree of improvement depends on the characteristics of the applications. Similar to the single-threaded applications, multithreaded applications with low L2 cache MPKI are mostly insensitive to the DRAM organizations. Among the ones with higher L2 cache MPKI, we present the relative IPC and EDP of canneal from PARSEC and RADIX from SPLASH-2 in Figure 14. Again, AR×1 is the reference DRAM organization. RADIX is an integer radix sort application that randomly accesses main memory, while canneal has memory regions that are accessed more frequently. Therefore, we use the access frequency aware memory mapping only for canneal.

The performance of canneal is substantially improved both by increasing the aspect ratio of the mats and by decreasing the read to data delay (tAA) on the center blocks, but the latter has higher influence. AR×2, AR×4, and AR×8 improve the IPC of canneal by 10%, 14%, and 15%, respectively. Exploiting the low tAA of the center blocks enables more of a speedup; CHARM[×2,/4] and CHARM[×4,/4] increase the IPC by 15% and 21%, respectively, as the center blocks, which correspond to only 25% of the DRAM capacity, service 89% of main-memory accesses (Figure 10(b)) with the access frequency aware memory allocation scheme. Note that CHARM[×4,/8] increases the IPC by 14% compared to the reference organization, which is slightly worse than CHARM[×2,/4]. This occurs because a significant portion of the frequently accessed memory footprints are not allocated to the center HAR mats due to their limited capacity, which negates the benefits of lower tAA on the HAR mats.

The reduction in the DRAM cycle time is the most influential factor for RADIX. As opposed to the case of canneal, CHARM[×4,/4] increases the IPC only by 1.6%. AR×2, AR×4, and AR×8 improve the IPC by 6.2%, 7.3%, and 7.8%, respectively. Because the capacity of the center HAR mats is only one eighth and one fourth of the total capacity for CHARM[×4,/8] and CHARM[×2,/4], the performance gains are 0.6% and 1.5%, respectively.

6. RELATED WORK

There is a large body of research that aims to improve the performance and energy efficiency of main-memory systems. Besides core-side multi-level caches and fast-page modes in DRAMs, there have been many proposals sharing this goal.

High-Performance DRAM Bank Structures: There have been several proposals to lower the average access time by reducing the number of cells attached to a bitline, including Reduced Latency DRAM (RLDRAM) [34], MoSys 1T-SRAM [13], and Fast Cycle DRAM (FCRAM) [42]. This can be achieved by either fragmenting a cell array into smaller subarrays [34, 42] or by increasing the bank count [13]. However, these strategies mainly involve low-cost SRAM replacements with much higher area overhead than CHARM [16]. For example, an RLDRAM memory array and associated circuits are reported to be 40-80% larger than those of a comparable DDR2 device [20]. Single Subarray Access (SSA) [47] and Fine-Grained Activation [11] microarchitectures read or write data to an entirely activated row in a DRAM mat. This activates fewer mats per access and reduces energy consumption. However, it requires as many data lines as the row size of a mat, thus requiring much higher area and static power overhead than CHARM.

Kim et al. [23] propose the subarray-level parallelism system, which overlaps the latencies of different requests that head to the same bank. Sudan et al. [45] propose micropages which allow chunks from different pages to be colocated in a row buffer to improve both the access latency and energy efficiency by better exploiting locality. CHARM is complementary to both proposals, as we focus on modifying the microarchitecture of DRAM banks to lower the access and cycle times.

DRAM-Side Caching: Both Enhanced DRAM [56] and Virtual Channel DRAM [2] add an SRAM buffer to the DRAM device to cache frequently accessed DRAM data. Any hit in the cache obviates the need for a row access and hence improves the access time. Because an SRAM cell is much larger than a DRAM cell, the cache incurs significant area overhead. In contrast, CHARM adds only minimal area overhead (a 3% increase) to conventional DRAM devices and is suitable for multi-gigabyte main memory. Tiered-latency DRAM [26] also advocates an idea to make a portion of a mat to have lower access time, but it is targeted to be used as a cache.

3D Die Stacking: Madan et al. [31] and Loh and Hill [30] propose the idea of stacking a DRAM die on top of a processor die, connecting them using through-silicon vias (TSVs), and using the DRAM die as last-level caches, while wide I/O DRAMs [22] are used for main memory. Hybrid Memory Cube [38] (HMC) packages a logic die below multiple DRAM dies to improve the capacity per package and bandwidth for a processor package. Udipi et al. [46] also propose to offload part of the memory controller's function to the logic die of an HMC-style structure, and Loh [29] puts row-buffer caches onto the logic die. It is feasible to stack multiple CHARM DRAM dies to further reduce the access latency while increasing access bandwidth.

DRAM Module-Level Solutions: To enhance the energy efficiency of accessing main memory, multiple groups [4, 58] have proposed rank subsetting, the idea of utilizing not all but a subset of DRAM chips in a DRAM rank to activate fewer DRAM cells per access. These methods save energy at the cost of additional latency, while CHARM lowers both access latency and energy consumption. Rank subsetting has been applied to improve the reliability and energy efficiency of main-memory systems as well [47, 53, 54]. The Fully-Buffered DIMM (FBDIMM) architecture [12] is characterized by non-uniform access time to DRAM banks, which is similar to CHARM from the viewpoint of a memory controller. However, an increase in latency and power consumption incurred by Advanced Memory Buffers in FB-DIMM limits its applicability compared to CHARM.

7. CONCLUSION

Recent research and development of DRAM microarchitectures for main-memory systems have mostly focused on increasing the capacity, bandwidth, and energy efficiency of the DRAM devices while retaining or even sacrificing the access latency. However, application performance is often sensitive not only to bandwidth but also to latency and applications access small portions of the memory footprints more frequently than the remainder. This observation has motivated us to propose asymmetric DRAM bank organizations that reduce the *average* access and cycle times and analyze their system-level impacts on performance and energy efficiency.

The cycle time analysis of contemporary DRAM devices shows that the sensing, restore, and precharge processes are slow due to high bitline capacitance within DRAM mats. Increasing the aspect ratio of the mats cuts the cycle time more than half and also reduces the access time and activate energy at the cost of increased area. To minimize the area overhead, we synergistically combine the high-aspect-ratio mats with support for non-uniform bank accesses to devise a novel DRAM organization called CHARM. A CHARM DRAM device places the high-aspect-ratio mats only at the center blocks, which are physically closer to I/Os, and reorganizes the column decoders and inter-bank datalines so that the access time to the center high-aspect-ratio mats is about half the access time to the other normal mats. Simulation results on a chip-multiprocessor system demonstrate that applications with higher bandwidth demands on main memory typically benefit more from CHARM. For example, CHARM DRAM devices with 2× higher aspect-ratio mats placed on a quarter of the DRAM banks increase the area only by 3%, but improve the IPC and the EDP of the system up to 21% and 32%, respectively, on single-threaded applications compared to the reference uniform organization. Similar degrees of performance and energy efficiency gains are also realized on the multithreaded and multiprogrammed workloads that frequently access main memory.

8. ACKNOWLEDGMENTS

The authors thank Uksong Kang, Hak-soo Yu, Churoo Park, Jung-Bae Lee, and Joo Sun Choi from Samsung Electronics for their helpful comments.

9. REFERENCES

[1] "The SAP HANA Database," http://www.sap.com.
[2] "Virtual Channel DRAM. Elpida Memory, Inc." http://www.elpida.com/en/products/eol/vcdram.html.
[3] J. Ahn, "ccTSA: A Coverage-Centric Threaded Sequence Assembler," *PLoS ONE*, vol. 7, no. 6, 2012.
[4] J. Ahn *et al.*, "Improving System Energy Efficiency with Memory Rank Subsetting," *ACM TACO*, vol. 9, no. 1, 2012.
[5] J. Ahn *et al.*, "McSimA+: A Manycore Simulator with Application-level+ Simulation and Detailed Microarchitecture Modeling," in *ISPASS*, Apr 2013.
[6] R. Alverson *et al.*, "The Tera Computer System," in *ICS*, Jun 1990.
[7] D. L. Anand *et al.*, "Embedded DRAM in 45-nm Technology and Beyond," *Design Test of Computers, IEEE*, vol. 28, no. 1, 2011.
[8] S.-J. Bae *et al.*, "A 40nm 2Gb 7Gb/s/pin GDDR5 SDRAM with a Programmable DQ Ordering Crosstalk Equalizer and Adjustable Clock-tracking BW," in *ISSCC*, Feb 2011.
[9] A. Bhattacharjee and M. Martonosi, "Thread Criticality Predictors for Dynamic Performance, Power, and Resource Management in Chip Multiprocessors," in *ISCA*, Jun 2009.
[10] C. Bienia *et al.*, "The PARSEC Benchmark Suite: Characterization and Architectural Implications," in *PACT*, Oct 2008.
[11] E. Cooper-Balis and B. Jacob, "Fine-Grained Activation for Power Reduction in DRAM," *IEEE Micro*, vol. 30, no. 3, 2010.
[12] B. Ganesh *et al.*, "Fully-Buffered DIMM Memory Architectures: Understanding Mechanisms, Overheads and Scaling," in *HPCA*, Feb 2007.
[13] P. N. Glaskowsky, "MoSys Explains 1T-SRAM Technology," *Microprocessor Report*, Sep. 1999.
[14] M. Hashimoto *et al.*, "An Embedded DRAM Module using a Dual Sense Amplifier Architecture in a Logic Process," in *ISSCC*, Feb 1997.

[15] J. L. Henning, "SPEC CPU2006 Memory Footprint," *Computer Architecture News*, vol. 35, no. 1, 2007.
[16] B. Jacob *et al.*, *Memory Systems: Cache, DRAM, Disk*. Morgan Kaufmann Publishers Inc., 2007.
[17] D. James, "Recent Innovations in DRAM Manufacturing," in *Advanced Semiconductor Manufacturing Conference*, Jul 2010.
[18] U. J. Kapasi *et al.*, "Programmable Stream Processors," *IEEE Computer*, vol. 36, no. 8, 2003.
[19] D. Kaseridis *et al.*, "Minimalist Open-page: a DRAM Page-mode Scheduling Policy for the Many-core Era," in *MICRO*, Dec 2011.
[20] B. Keeth *et al.*, *DRAM Circuit Design*, 2nd ed. IEEE, 2008.
[21] C. Kim *et al.*, "An Adaptive, Non-Uniform Cache Structure for Wire-Delay Dominated On-Chip Caches," in *ASPLOS*, Oct 2002.
[22] J.-S. Kim *et al.*, "A 1.2V 12.8GB/s 2Gb mobile Wide-I/O DRAM with 4x 128 I/Os using TSV-based stacking," in *ISSCC*, Feb 2011.
[23] Y. Kim *et al.*, "A Case for Exploiting Subarray-Level Parallelism (SALP) in DRAM," in *ISCA*, Jun 2012.
[24] C. Kozyrakis, "Scalable Vector Media-processors for Embedded Systems," Ph.D. dissertation, University of California at Berkeley, 2002.
[25] D. Lee *et al.*, "LRFU: A Spectrum of Policies that Subsumes the Least Recently Used and Least Frequently Used Policies," *IEEE TC*, vol. 50, no. 12, 2001.
[26] D. Lee *et al.*, "Tiered-Latency DRAM: A Low Latency and Low Cost DRAM Architecture," in *HPCA*, Feb 2013.
[27] S. Li *et al.*, "The McPAT Framework for Multicore and Manycore Architectures: Simultaneously Modeling Power, Area, and Timing," *ACM TACO*, vol. 10, no. 1, 2013.
[28] E. Lindholm *et al.*, "NVIDIA Tesla: A Unified Graphics and Computing Architecture," *IEEE Micro*, vol. 28, no. 2, 2008.
[29] G. H. Loh, "A Register-file Approach for Row Buffer Caches in Die-stacked DRAMs," in *MICRO*, Dec 2011.
[30] G. H. Loh and M. D. Hill, "Efficiently Enabling Conventional Block Sizes for Very Large Die-stacked DRAM Caches," in *MICRO*, Dec 2011.
[31] N. Madan *et al.*, "Optimizing Communication and Capacity in a 3D Stacked Reconfigurable Cache Hierarchy," in *HPCA*, Feb 2009.
[32] J. D. McCalpin, "STREAM: Sustainable Memory Bandwidth in High Performance Computers," University of Virginia, Tech. Rep., 1991.
[33] Micron Technology Inc., *LPDDR2 SDRAM Datasheet*, 2010.
[34] Micron Technology Inc., *RLDRAM3 Datasheet*, 2011.
[35] O. Mutlu and T. Moscibroda, "Parallelism-Aware Batch Scheduling: Enhancing both Performance and Fairness of Shared DRAM Systems," in *ISCA*, Jun 2008.
[36] D. Patterson *et al.*, "A Case for Intelligent RAM," *Micro, IEEE*, vol. 17, no. 2, 1997.
[37] D. A. Patterson and J. L. Hennessy, *Computer Architecture: A Quantitative Approach*, 5th ed. Morgan Kaufmann Publishers Inc., 2012.
[38] J. T. Pawlowski, "Hybrid Memory Cube," in *Hot Chips*, Aug 2011.
[39] L. E. Ramos *et al.*, "Page Placement in Hybrid Memory Systems," in *ICS*, Jun 2011.
[40] S. Rixner *et al.*, "Memory Access Scheduling," in *ISCA*, Jun 2000.
[41] Samsung Electronics, *DDR3 SDRAM Datasheet*, 2012.
[42] Y. Sato *et al.*, "Fast Cycle RAM (FCRAM); a 20-ns Random Row Access, Pipelined Operating DRAM," in *VLSI*, Jun 1998.
[43] T. Sherwood *et al.*, "Automatically Characterizing Large Scale Program Behavior," in *ASPLOS*, Oct 2002.
[44] A. Snavely and D. Tullsen, "Symbiotic Job Scheduling for a Simultaneous Mutlithreading Processor," in *ASPLOS*, Nov 2000.
[45] K. Sudan *et al.*, "Micro-pages: Increasing DRAM Efficiency with Locality-aware Data Placement," in *ASPLOS*, Oct 2010.
[46] A. N. Udipi *et al.*, "Combining Memory and a Controller with Photonics through 3D-stacking to Enable Scalable and Energy-efficient Systems," in *ISCA*, Jun 2011.
[47] A. N. Udipi *et al.*, "Rethinking DRAM Design and Organization for Energy-constrained Multi-cores," in *ISCA*, Jun 2010.
[48] B. Verghese *et al.*, "Operating System Support for Improving Data Locality on cc-NUMA Compute Servers," in *ASPLOS*, Oct 1996.
[49] T. Vogelsang, "Understanding the Energy Consumption of Dynamic Random Access Memories," in *MICRO*, Dec 2010.
[50] S. C. Woo *et al.*, "The SPLASH-2 Programs: Characterization and Methodological Considerations," in *ISCA*, Jun 1995.
[51] W. A. Wulf and S. A. McKee, "Hitting the Memory Wall: Implications of the Obvious," *Computer Architecture News*, vol. 23, no. 1, 1995.
[52] Y. Yanagawa *et al.*, "In-substrate-bitline Sense Amplifier with Array-noise-gating Scheme for Low-noise $4F^2$ DRAM Array Operable at 10-fF Cell Capacitance," in *VLSI*, Jun 2011.
[53] D. H. Yoon *et al.*, "BOOM: Enabling Mobile Memory Based Low-Power Server DIMMs," in *ISCA*, Jun 2012.
[54] D. H. Yoon and M. Erez, "Virtualized ECC: Flexible Reliability in Main Memory," *IEEE Micro*, vol. 31, no. 1, 2011.
[55] D. H. Yoon *et al.*, "Adaptive Granularity Memory Systems: a Tradeoff Between Storage Efficiency and Throughput," in *ISCA*, Jun 2011.
[56] Z. Zhang *et al.*, "Cached DRAM for ILP Processor Memory Access Latency Reduction," *IEEE Micro*, vol. 21, no. 4, 2001.
[57] W. Zhao and Y. Cao, "New Generation of Predictive Technology Model for Sub-45nm Design Exploration," in *ISQED*, Mar 2006.
[58] H. Zheng *et al.*, "Mini-Rank: Adaptive DRAM Architecture for Improving Memory Power Efficiency," in *MICRO*, Nov 2008.