공학석사 학위논문

Save Without Sacrifice: Understanding and Exploiting Power-performance Relationship of Energy-efficient Modern DRAM Devices

DRAM의 전력-성능 상보 관계를 고려한

높은 에너지 효율의 메모리 시스템 설계

2017 년 2 월

서울대학교 융합과학기술대학원

융합과학부 지능형융합시스템전공

조 현 윤

Save Without Sacrifice: Understanding and Exploiting Power-performance Relationship of Energy-efficient Modern DRAM Devices

DRAM의 전력-성능 상보 관계를 고려한

높은 에너지 효율의 메모리 시스템 설계

지도교수 안 정 호

이 논문을 공학석사 학위논문으로 제출함

2017 년 1 월

서울대학교 융합과학기술대학원

융합과학부 지능형융합시스템전공

조 현 윤

조현윤의 공학석사 학위논문을 인준함

2017 년 1 월

위 원 장 곽 노 준 (인)

부위원장 안 정 호 (인)

위 원 박 재 흥 (인)

Abstract

Save without Sacrifice: Understanding and Exploiting Power-performance Relationship of Energy-efficient Modern DRAM Devices

Hyunyoon Cho

Intelligence Systems

Department of Transdisciplinary Studies

The Graduate School

Seoul National University

As servers are equipped with more memory modules each with larger capacity, main-memory systems are now the second highest energy-consuming component in big-memory servers and their energy consumption even becomes comparable to processors in some servers. Meanwhile, it is critical for big-memory servers and their main-memory systems to offer high energy efficiency. In pursuit of energy-efficient main memory systems, prior work exploited mobile LPDDR devices' advantages (lower power than DDR devices) while attempting to surmount their limitations (longer latency, lower bandwidth, or both). However, we demonstrate that such main memory architectures (based on the latest LPDDR4 devices) are no longer effective and even hurt overall energy efficiency of servers by 49% on memory intensive workloads compared to ones based on DDR4 devices. This is because the power

consumption of present DDR4 devices has substantially decreased by adopting the strength of mobile and graphics memory whereas LPDDR4 has sacrificed energy efficiency and focused more on increasing data transfer rates; we also exhibit that the power consumption of DDR4 devices can substantially vary across manufacturers. Moreover, investigating new energy-saving features of DDR4 devices in depth, we show that activating these features often hurts overall energy efficiency of servers due to their performance penalties. Subsequently, we propose a simple but effective scheme that adaptively exploits DRAM power-down modes which improves the system energy-delay product by 4.0%.

Keywords: Memory system, DDR4 SDRAM, Power/energy reduction, Latency, Data bus inversion, DBI, 3D-stack, TSV

Student Number : 2015-26051

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Servers for emerging applications such as big-data analytics and public cloud services [8] demand ever larger DRAM for main-memory systems. In particular, software layers of big-data analytics, such as Spark [48] and Storm [3], and high-performance key-value stores [29] increasingly exploit and seek larger main-memory systems for higher performance. Consequently, servers for such applications and services began to be equipped with more memory modules with larger capacity. This trend had made main-memory systems the second highest energy-consuming component trailing only processors in servers, and the energy consumption of main-memory systems became even comparable to that of processors in some server configurations [15]. Meanwhile, the increasing energy consumption of servers has been a growing concern for operating large-scale datacenters due to the huge impacts of consuming a large amount of energy on the environment and operating cost [5], [31]. Therefore, it is critical to maximize energy efficiency of datacenter servers and their main-memory systems, where Synchronous

DRAM (SDRAM) devices and their successors, such as DDR, DDR2, and DDR3 have been used as main memory of datacenters and most other computing segments for decades. The bandwidth, capacity, and energy efficiency of these mainstream DRAM devices have steadily improved. Nonetheless, these DRAM devices were suboptimal for big-memory servers as they were architected to be versatile for all computing segments by balancing between bandwidth, latency, capacity, reliability, and energy.

In pursuit of building more energy-efficient main-memory systems for big-memory servers, main-memory architectures exploiting LPDDR devices were proposed [30], [46]. This was because LPDDR devices, which mainly target mobile computing, consumed much lower power than DDR devices (but at the cost of longer latency and lower bandwidth). In this thesis, however, *we first demonstrate that such main-memory architectures do not make servers more energy-efficient than ones based on the latest DDR devices (i.e., DDR4) in most usage scenarios any more.* This is because both LPDDR and DDR devices have evolved over generations and the power consumption of current DDR4 devices has substantially decreased by adopting the strength of mobile and graphics memory, whereas the latest LPDDR4 has sacrificed energy efficiency and focused more on increasing data transfer rates.

More specifically, we show that DDR4 is far more energy-efficient than DDR3 not only because it is manufactured with finer-pitch technology but also because it adopts various advanced circuit-level techniques in particular to aggressively reduce static power consumption. This entails smaller relative power consumption gap between DDR4 and LPDDR4 (39%) than between DDR3 and LPDDR2 (77%). Moreover, during this analysis, *we also discover that static*

power consumption of DRAM devices notably varies across DRAM manufacturers (up to 2.2 ×) and may choose more energy-efficient DDR4 devices for big-memory servers; total power consumption of DDR4 devices from one manufacturer is 16–38% lower than ones from two other manufacturers.

Subsequently, we present in-depth analyses on new energy-saving features offered by contemporary DDR4 devices and show that they (e.g., data bus inversion (DBI)) *often hurt overall energy efficiency of big-memory servers and micro-servers [5] because they incur performance penalties.* This underscores the importance of offering energy-saving technologies that do not incur notable performance penalties. Subsequently, we propose a simple but effective scheme that exploits DRAM power-down modes adaptively which improves the energy-delay product (EDP) of a simulated big-memory system with eight energy-efficient DDR4 ranks per channel by 4.0% on memory intensive multi-programmed workloads.

In summary, we make the following key contributions:

- In contrast to prior proposals based on LPDDR2 devices, we demonstrate that main-memory architectures exploiting the advantages of LPDDR4 devices do not make big-memory servers and micro-servers more energy-efficient than ones based on DDR4.
- While exhibiting why DDR4 devices are no longer energy-inefficient than LPDDR4 devices, we expose that static power consumption of DDR4 devices notably varies across manufacturers.
- We present in-depth analyses on new energy-saving features supported by contemporary DDR4 devices and show

that these features are not effective when considering system-level energy efficiency.

- We enhance energy-saving features of DDR4 devices to improve energy efficiency of big-memory servers and evaluate their impacts on system performance and energy efficiency.

Chapter 2

Background and Related Work

2.1 DRAM Organization and Operation

Main memory DDRx DRAM devices are organized to achieve high capacity and bandwidth with reasonable latency and energy efficiency under stringent cost constraint [23] (Figure 1). Mobile LPDDRx [19] and graphics GDDRx [17] are organized similarly, but the former focuses more on energy efficiency whereas the latter emphasizes high data transfer rates per device. A modern DDR4 DRAM die stores 4Gb or 8Gb of data, consists of 16 banks, and has 4 (×4) or 8 (×8) data pins typically, each transferring data at the rates equal to or above 1.6Gbps. Each bank has a 2D array of DRAM cells, where a cell consists of an access transistor and a capacitor. In order to achieve high area efficiency, cells in a bank share wires and

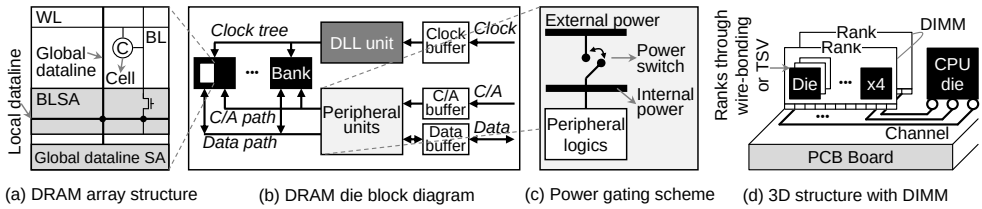


Figure 1. DRAM organizations and 3D system structure with DIMMs

peripheral circuitry of both control and datapath. As a DRAM bank comprises hundreds of millions of cells, the number of cells connected to a wordline (WL) or a bitline (BL) becomes too excessive, and both BLs and WLs are structured hierarchically. For datapath, each bank has dozens of rows of BL sense amplifiers (BLSAs) and there exist global datalines that span the entire height of the bank. Because the number of global datalines per bank, which is equal to the number of bits transferred per read/write transaction, is much smaller than the row (page) size of a bank, (de)multiplexers called local datalines exist per row of BLSAs.

This sharing of wires and circuitry goes beyond a DRAM bank boundary. All banks in a DRAM die work independently except that they share datapath and control wires. One or more DRAM dies are packaged in a DRAM device. Multiple dies stacked in a DRAM device are connected by through-silicon vias (TSVs) or wire-bonding pads. Several dies across DRAM devices are grouped together and operate in tandem receiving the same command and address signals, constituting a rank. A memory controller and multiple ranks are connected through a single memory channel, where command, address, and data signals are transferred. One or more ranks of DRAM devices are placed together on a module. The number of datapath wires in a memory channel is 64 in a modern dual-inline

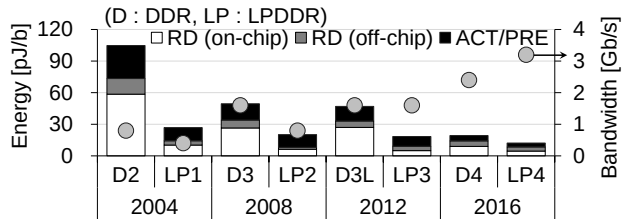
memory module (DIMM) excluding optional 8-bit wires for error checking and correction (ECC).

Popular DRAM devices, such as DDR3 [18], DDR4 [19], LPDDR4 [21], and GDDR5 [17], access data through a sequence of commands. To access data in a bank, the row including the data should first be latched to the corresponding BLSAs using an activate (ACT) command. After tRCD since ACT is issued, a read (RD) or write (WR) command can be issued to specify the column location within the latched row, and it takes tCL (tWL) to have the first data popped out of (shipped to) the device for RD (WR) and takes $tCCD_S$ to transfer a burst of data. Data in the selected cells are destroyed during row activation, and hence should be restored to keep the value, taking tRAS. WR needs time to update the data in the corresponding DRAM cells, defined as write recovery time or tWR. Once data are restored or updated, the bank can receive a precharge (PRE) command to deactivate the BLSAs and to precharge BLs to be ready for subsequent activate commands, taking tRP. tRAS + tRP constitutes a DRAM cycle time called tRC. BLSAs that hold a row specified by ACT are called a row buffer of the bank. ACT/PRE are row commands whereas RD/WR are column commands. The row (page) size of a DDR4 rank is 8KB. A DRAM bank operates at much lower clock frequency (defined to be $tCCD_L$) than the transfer rate of a data signal (around 2.4Gbps, which is 2b/tCK, in the latest DDR4 devices). Therefore, internal datapath of a bank is much (8×) wider than the datapath width of a DRAM device, determining burst length. For example, a ×8 DDR4 device has 64 global datalines per bank. Because $tCCD_L$ is still larger than $8\times{}^{tCK}/_{2}=2$, 16 banks of a DDR4 device are divided into 4 bank groups where data transfers to and from different bank groups can occur consecutively in time,

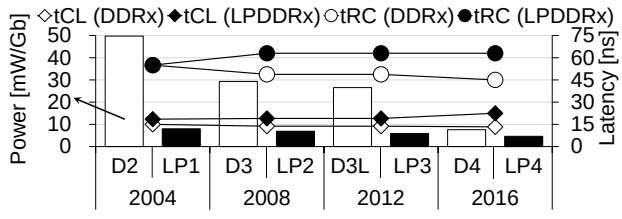
determining $tCCD_S$ = 4tCK.

2.2 Breaking Down DRAM Power Dissipation

DRAM dissipates most power by the following components: data read/write including inter-device signal transfers, activate/precharge to latch stored data in DRAM row buffers, refresh to retain values in leaky DRAM cells, and standby power from the DRAM internal units including delay-locked loop (DLL) that tracks the phase of master clock from a memory controller, input/output buffers, and peripheral circuits [16, 43]. We can classify these components by whether they consume power regardless of data transfer activities or not; refresh and standby can be categorized as static, whereas activate, precharge, read, and write components as dynamic. These dynamic and static power values are presented in



(a) Dynamic energy and bandwidth



(b) Static power and latency

Figure 2. Trends of the dynamic energy, bandwidth, static power and key latency values of DDRx and LPDDRx devices aligned by generations (the years in the bottom correspond to when standards were/are popular.).

DRAM datasheets using I_{DD} specifications. For example, I_{DD2N} specifies the current of a device when it has no active pages and stays at a standby mode. A DRAM in a standby mode (e.g., I_{DD3N}) can receive any commands whereas if it is in a power-down mode (e.g., I_{DD3P}), the device must exit out of the mode to process normal commands, such as ACT, PRE, RD, and WR. A device consumes less static power when it is in a power-down mode than a standby mode. The energy efficiency of a DRAM device has been improved substantially over time. The dynamic energy of main memory DRAM in a system depends on the frequency and characteristics of memory accesses, such as the ratios of row commands over column commands (d), whereas its static power is influenced by the memory capacity of the system and their states, such as temperature and the average number of active banks. Both dynamic energy and static power are heavily influenced by operating voltages (the lower the better) and fabrication technology (the narrower the better). Figure 2 shows the key latency, dynamic energy (pJ/b), and static power (mW/Gb) values over multiple generations of ×4 DDR and ×32 LPDDR devices from manufacturer **A**[1] We assume that a device operates at 85℃. The generations and per-pin data transfer rates are denoted by (LP)DDR**g-S**, where **g** is generation and **S** is data transfer rate. We use δ of 0.27, the average over memory intensive SPEC CPU2006 applications reported in [24]. ACT/PRE energy is proportional to δ. We paired DDR and LPDDR devices that were/are popular at similar years. DDR3L stands for DDR3 with lower operating voltage (VDD, 1.5V for DDR3 vs. 1.35V for DDR3L).

[1] Because the IDD values of LPDDR4 devices are not publicly available, we estimate those based on the projection from LPDDR3 devices considering operating voltage, data transfer rate, and fabrication process scaling. Also, the datapath width of LPDDR4 is ×16.

As VDD decreases and finer-pitch fabrication technologies are introduced over generations, both dynamic energy and static power have been improved steadily. LPDDR devices consume much lower static power than DDR devices of the same generations at a given capacity as LPDDR uses transistors with higher threshold voltage, which leak less but also operate slower. In addition, LPDDR adopts more aggressive power gating techniques for internal datapath. These all make LPDDR achieve substantially lower leakage power than DDR, but at the cost of higher latency values. For example, tRC of DDR4 is 45.3ns whereas that of LPDDR4 is 60ns. Also, the major timing parameters of DDR are reduced over time whereas those of LPDDR are growing.

2.3 Recent Progresses in Improving the Energy Efficiency of Main Memory Systems

The bandwidth and latency of main memory systems, which significantly affect the overall system performance and thus energy efficiency, are strongly dependent on the service order of memory requests. That is, sequential accesses to different rows within a bank lead to high latency and cannot be pipelined, whereas accesses to different banks or different words within a single row have low latency and can be pipelined. Therefore, memory requests can be scheduled (out of order) to maximize consecutive accesses to the same row in a bank or to different banks, which can greatly improve performance of main memory systems [36]. With such a scheduling technique, increasing the number of banks allows a memory system to service more memory requests in parallel. This entails lower

memory access latency (and thus higher system energy efficiency) but also incurs notably higher implementation cost. To cost-effectively support more parallel memory accesses, multiple sub-arrays constituting a modern DRAM bank has been exploited [25, 41, 49]. The sub-arrays of a bank share few global peripheral structures, but they can operate independently in most parts. Thus, different components of the bank access latencies on multiple requests can be overlapped such that they head to different subarrays within the same, effectively facilitating more parallel/pipelined memory accesses to each bank.

In modern DRAM, a row is typically comprised of a large number of cells (8–16Kb). Consequently, activating and precharging a row consume significant energy. When accesses to DRAM exhibit high spatial locality, the high energy cost of activation/precharge can be amortized. However, DRAM accesses by many-core processors lack spatial locality, and ensuing frequent row activations and precharges lead to significant energy inefficiency. Thus, various DRAM architectures have been devised to activate and precharge fewer cells of a row (i.e., lower energy per activation) without incurring high implementation cost [47, 49, 51].

As the data transfer rate steadily goes up, DRAM I/O energy has become another significant contributor to DRAM total energy. As DRAM I/O energy is also strongly data-dependent (e.g., the number of zeros or ones driven to data bus), simply counting the number of zeros (or ones) to be placed on the data bus and inverting the bit values if there are more zeros (or ones) can reduce DRAM I/O energy [4]. Besides, more bits per device lead to more energy consumption as DRAM cells should be refreshed periodically to retain their states. Because not all the DRAM cells require the same refresh

frequency, various selective refresh techniques have been explored [6, 9, 33, 50].

Providing memory systems with high energy efficiency and proportionality is critical for datacenter servers because they impact cost and scalability. The past DDR DRAM focused more on high bandwidth and capacity, and was not highly optimized for energy efficiency/proportionality. To offer highly energy-efficient and -proportional memory systems for datacenter servers, the use of mobile DRAM, which was optimized for energy efficiency at the cost of increased latency and reduced bandwidth, has been proposed [30, 46]. However, these studies did not fully consider the latency penalties listed in Figure 2, while assuming the timing parameters in favor of LPDDR2 devices [46]; we further elaborate these in Section 3.1. Also, although various low-power modes are supported by modern DRAM, they are too slow to be used by memory systems for datacenter servers and DRAM architecture supporting fast-transition low-power modes are investigated [31]. There have been studies to categorize data by their hotness (access frequency) and to allocate/migrate them to few ranks [26, 45] for better exploiting low-power modes, which are orthogonal to this thesis.

Lastly, even if some of the aforementioned techniques improve system energy efficiency by reducing average memory access latency values, many of the DRAM static or dynamic power saving techniques impact system performance negatively. Moreover, the degrees of power saving and performance degradation heavily depend on the material-, circuit-, and architecture-level techniques of both CPU and DRAM devices. Therefore, the effectiveness of certain techniques should be carefully quantified through popular metrics, such as system energy and energy-delay product (EDP), in

present and future systems, as the ideas that were valid once in the past, might not be compelling any more.

Chapter 3

Energy Efficiency and Performance Trade-Offs of Modern Main Memory Devices

Given that numerous energy saving techniques for DRAM based main memory compromise performance, it is critical to quantify their trade-offs using popular effectiveness metrics, such as system-level energy consumption and energy delay product (EDP), as each technique has different degrees of impact on DRAM static/dynamic power. We first re-visit the ideas of exploiting mobile LPDDR devices instead of mainstream DRAM devices were reasonable when those were proposed, but is not any more. Then, we assess the primary energy saving techniques introduced at the latest DDR4 devices and propose novel techniques to better exploit DRAM power-down modes and data bus inversion (DBI).

3.1 DDR4 is not Energy Inefficient Any More

We re-examine prior works to assess the effectiveness of utilizing low power mobile (LPDDRx) DRAM device. Both BOOM [46] and Malladi et al. [30] advocated using unmodified LPDDR devices (LPDDR2 in their studies). LPDDR2 devices had lower per-pin data transfer rate (0.8Gbps) compared to that of DDR3 devices (1.6Gbps) with superior (lower) dynamic energy and static power values as shown in Figure 2. Malladi et al. [30] reduce main-memory bandwidth accordingly to use LPDDR2 instead of DDR3, and reported substantial savings in both energy and total cost of ownership (TCO) on datacenter applications. Instead, BOOM [46] groups more pins to constitute a rank, increases per-pin data transfer rate between a memory controller and modules by having a buffer chip per module, and further improves energy efficiency by leveraging rank subsetting [1] which trades higher access latency with more ranks (tailored to better exploit bank-level parallelism) and smaller row buffers.

In contrast to Malladi et al. [30] and BOOM [46], we evaluate a

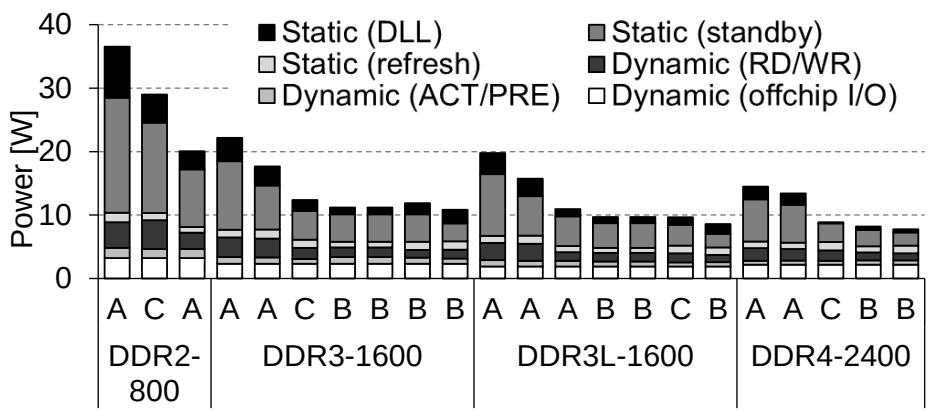


Figure 3. Power breakdown of DDR2/3/3L/4 DRAM ranks sold October 2016 from 3 major manufacturers (**A**, **B**, and **C**). We downloaded datasheet from DRAM vendors web page, and these are published document. We report the static power of 8 ranks connected at a channel, reflecting the I/O power accordingly.

modified version of LPDDR due to the following reasons. First, per-pin data transfer rate of the latest LPDDR4 devices is not lower than that of DDR4 devices at the same generation any more. LPDDR4-3200 devices are currently on the market, whereas DDR4-2400 is the fastest DDR4 devices, except ones from few overclocking vendors. Therefore, the idea of utilizing more pins per rank in BOOM is not directly applicable. Second, an LPDDRx device has wide datapath (×16 or ×32) whereas most DIMMs are equipped with ×4 or ×8 DDRx devices. The two aforementioned reasons make it difficult, if not impossible, to achieve the same degree of reliability without substantially sacrificing DRAM capacity with these wide datapath devices even through several techniques proposed [1, 30, 46]. Third, much better I/O energy efficiency of LPDDRx originates from better signal integrity of mobile systems as only few DRAM devices are connected to a memory controller through a bus with distance of up to few millimeters. Therefore, buffer chips are must for LPDDR-based memory modules such as BOOM, which increases access latency and power, whereas DDRx based memory modules can dispense with buffer chips when the number of banks per memory channel is low. Fourth, the burst length of LPDDR4 is 16 whereas that of DDR4 is 8. Longer burst length hurts the performance of applications with low spatial locality in memory accesses. We model the modified version of LPDDR4, what we call LPDDR4' hereafter, as follows; basically, LPDDR4' uses the material and circuit-level technologies of LPDDR4 (except I/O) and adopts the micro-architectural features of DDR4, such as datapath width, row buffer size, and burst length.

Meanwhile, the energy efficiency, especially the static power of mainstream DDRx devices has been improved substantially over time.

Figure 3 shows the power breakdown of DDR2/3/3L/4 DRAM ranks that are sold as of November 2015. We collected the values from three major DRAM manufacturers, distinguished by A, B, and C [32, 37, 39]. We report the static power of 8 ranks connected in a channel, and reflect the I/O power accordingly; except DDR2, we use load-reduced DIMM (LRDIMM) to connect 8 ranks, which increases I/O power due to the buffer chips in LRDIMM. ×4 devices are used. The capacity of a DDR2 device is 2Gb, which is the maximum size being sold, whereas that of other devices is 4Gb. We assume that each device transfers data at its highest rate and the ratio of ACT over RD commands (δ) is 0.27, the value used in Section 2.2 as well.

We make the following key observations from Figure 3. First, supply voltage levels decrease as newer standards are introduced (1.8V/1.5V/1.35V/1.2V for DDR2/3/3L/4) and hence DDR4 is most energy efficient, reinforcing the observations made of Figure 2. Second, material-, fabrication-, and circuit-level technologies make huge variation in power within and across DRAM manufacturers. This is more prominent for static power of DDR4 devices; a device from **A** consumes more than twice the static power compared to those from **B** and **C**. Multiple factors contribute to this huge difference. For example, delay-locked loops (DLLs) in DRAM are traditionally implemented using analog circuits, occupying a considerable fraction of the static power of DDR2/3 devices. The introduction of digital DLLs, enabling DLL to be turned off most of time and just periodically to re-calibrate reference clock phases [27], is conjectured to substantial reduction in DLL power of certain manufacturers.

These material-, fabrication-, and circuit-level evolutions narrow

the gap between the current DDR4 and LPDDR4' devices. As shown in Figure 3 and Table 3, the DDR4 devices from **A** and **B** vendors consume 2.8× and 1.4× more static power than the LPDDR4' device, respectively. We test six configurations using the following combinations; 2 and 8 ranks per memory channel, DDR4 from **A**, **B**, and LPDDR4'. The baseline is DDR4 from **B** The reference chip-multiprocessor (CMP) configuration is specified in Section 5. The performance penalties of using LPDDR4' instead of DDR4 on the CMP for memory-intensive multi-programmed workloads are 28% and 34% for 2 and 8 rank cases, respectively (details in Section 6). This means that LPDDR4' is more energy efficient than DDR4 only for high-capacity (8 ranks per channel) servers equipped with power-hungry DDR4 from **A**. Even this configuration is more efficient than the one using LPDDR4' in EDP.

3.2 Saving Standby Power by Exploiting Power-down Modes

Instead of adopting the material- and circuit-level techniques of LPDDR4, which incur high latency penalty but provide insufficient power saving, we pay more attention to the energy saving techniques introduced at DDR4. We first exploit the power-down (PD) mode, which can save DRAM static power. Big-memory servers have several DRAM ranks per memory channel. Because only one rank can be the source or target of data transfers at any given time on a channel, it is important to put these remaining ranks in a PD mode as often as possible with minimal performance impact. Even if static power has decreased substantially on recent DDR3L/4, it is still

Standby state	Symbol	DLL/ clock	Peri- power on	Cmd/Addr buffer	Row active	Latency overhead Enter	Latency overhead Exit	Relative power A	Relative power B
Active standby	I_{DD3N}	●	●	●	●	N/A	N/A	1	1
Precharge standby	I_{DD2N}	●	●	●		N/A	N/A	0.92	0.68
Precharge standby w/CAL	I_{DD2NL}	●	●			N/A	5tCK	0.65	0.44
Precharge power-down	I_{DD2P}	●				1tCK +5ns	1tCK +6ns	0.52	0.41

Table 1. The DRAM components that are turned on, the corresponding latency overheads, and the relative power dissipation on various DRAM states.

above half of total DRAM power when eight ranks are populated in a channel (Figure 3). Moreover, when systems do not utilize main memory at peak bandwidth, static power saving is even more important.

A DDR4 device enters and exits a conventional PD mode by having (I/O) buffers and power-gates internal datapath (inter-bank and global datalines) of a DRAM device. Compared to a device in precharge standby mode (i.e., all banks stay precharged but are ready to accept any command (cf. I_{DD2N} in Table 1), one in precharge PD mode, where all banks also stay precharged but cannot accept any command except PD exit, consumes 40% less power in DDR4 made by **B** (cf. I_{DD2P} in Table 1). A device in a PD mode has following constraints. First, once entered, it should stay in the PD mode for a certain time period at least tCKE (5ns for DDR4-2400). Second, a device needs to wait for tXP (6ns for DDR4-2400) to receive any valid command after it receives the PD exit command. If DLL is frozen to save more static power, a device needs more time than tXP to receive a RD command because DLL must be locked again, called slow-exit mode (tXARD/tXPDLL for DDR2/3). Due to improvement in DLL circuitry, however, DDR4 does not support the slow-exit mode as DLL power has decreased substantially. For example, as

shown in Figure 3, DDR4-2400 from **B** consumes 0.52W for DLL whereas DDR3-1600 from **A** does as much as 3.7W for DLL, a substantial shrink considering even higher data transfer rate.

DDR4 supports an alternative static power saving scheme, called command address latency (CAL). CAL turns off the I/O buffers of a device by default and turns them on only when a command is issued to the device. Because the I/O buffers should be ready to receive any valid command, CAL exploits the CS (chip select) pin to notify the device a few cycles ahead for a normal command (tCAL, 5tCK for DDR4- 2400). Therefore, CAL increases the latency of any command by tCAL but allows a DRAM device to stay at a low power state as long as possible. This is in contrast to the conventional toggle-based PD mode which has latency penalties only to the first command after a PD exit, but imposes a burden of explicitly specifying when to enter the PD mode to a memory controller. Besides, to facilitate short tCAL (i.e., smaller than tCKE+tXP), CAL does not power-gate peripheral circuitry, entailing less power saving than the conventional PD mode (I_{DD2NL} vs. I_{DD2P}).

3.3 Saving Data Transfer Energy with DBI/TSV

Data bus inversion (DBI), which has been used for graphics [17] memory, is introduced to mainstream DRAM at DDR4. There are three components of energy consumption for data transfers between CPU and DRAM devices. First, DC energy (E_{DC}) is consumed by the drivers of a transmitter and at the on-die termination (ODT) resistor of a receiver. Second, AC energy (E_{AC}) is consumed by data bus toggles which happen when a currently transferred value is different

from the previously sent one. E_{DC} is inversely proportional to the channel resistance, whereas E_{AC} is proportional to the data transfer rate, the channel capacitance, and the bus toggling rate. Typically, high voltage (VDDQ) represents data one and ground does data zero. As DDR4 adopts pseudo open drain interface (Figure 4(a)), it does not consume DC power when transferring data one. The last is energy consumed by components inside of DRAM devices (E_{INT}), such as inter-bank/global/local datalines, which is mostly the same regardless of the value being transferred, whereas the first two I/O components are data value dependent. Therefore, total data transfer energy (E_{TR}[2]) is represented by $E_{TR} = \gamma_{DC}E_{DC} + \gamma_{AC}E_{AC} + E_{INT}$, where γ_{DC} is the probability of sending value zeros and γ_{AC} is the probability of consecutive data being toggled. When random data values are transferred, both γ_{DC} and γ_{AC} are 0.5.

3.3.1 Benefits of DBI

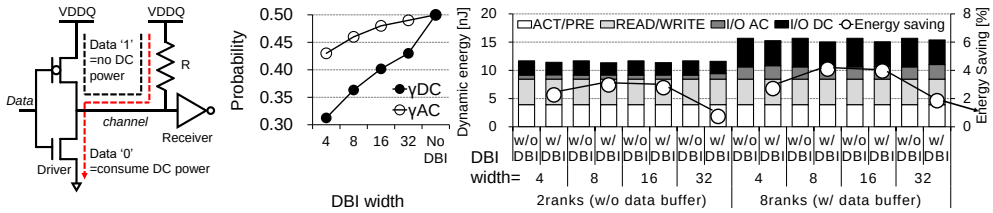
In a DDR4-2400 device, the data I/O consumes 46% of total dynamic power when it transfers data at peak bandwidth (Figure 3). Therefore, reducing data I/O energy can be as important as saving DRAM static power, especially for micro-servers that have just few ranks per memory channel. DBI in DDR4 counts the number of zeros on a group of data and flips them if zeros are majority, reducing the frequency of zero signals. In DDR4, the size of a group is equal to the datapath width of a DRAM device (e.g., 8 bits for ×8 devices). DBI decreases both the portion of zero values (lower γ_{DC}) and the frequency of data toggling (lower γ_{AC}). Throughout a Monte Carlo

[2] There is little difference between read and write energy, so we use the notation E_{TR} in this thesis.

simulation of transferring a million random numbers, the worst case scenario when transferring data through a channel, we observed that both γ_{DC} and γ_{AC} decrease with DBI as shown in Figure 4(b). As the size of a DBI group decreases, both probability values further decrease. Between the AC and DC components, γ_{DC} values are more sensitive to the DBI group size.

3.3.2 Energy savings by DBI considering its cost

However, these reduced probability values do not directly translate to the equivalent degree of DRAM energy saving as the cost of delivering information about whether data values are flipped or not should be considered. The additional DBI pin needed consumes both DC and AC energy. Figure 4(c) shows the DRAM dynamic energy breakdown with this overhead considered for the cases of 2 and 8 ranks per channel. With few (two) ranks in the channel, E_{DC} is much higher than E_{AC}. The cost due to the DBI pin is amortized as DRAM datapath width increases, but its benefit decreases for larger datapath widths, making DBI more efficient in data transfer energy



(a) Pseudo open drain (POD) scheme

(b) γ_{DC} and γ_{AC} over datapath width(× N)

(c) Total dynamic energy without data buffer (DB) chips and with DB chips. δ =0.27

Figure 4. Data bus inversion (DBI) on DDR4: (a) pseudo open drain (POD) DDR4 data bus I/O channel, (b) γ_{DC} and γ_{AC} over data bus width obtained through Monte Carlo simulation, (c) total DRAM dynamic energy with/without DBI over two DIMM configurations; 2 ranks for RDIMM and 8 ranks for LRDIMM.

for ×8 and ×16 devices. Combined with the fact that more pins in CPU induce higher cost premium, DDR4 does not support DBI for ×4 devices. Even if a ×8 device saves data transfer energy by 4.1%, the latency overhead of DBI and the resulting performance penalty should be considered carefully. DBI increases tCL, read command to first data out time, by 3tCK. Because system performance is most sensitive to tCL among DRAM timing parameters, small improvement in data transfer energy can be negated by additional energy consumed due to increased execution time, as shown in Figure 7 for 2 rank cases.

3.3.3 Impact of module types

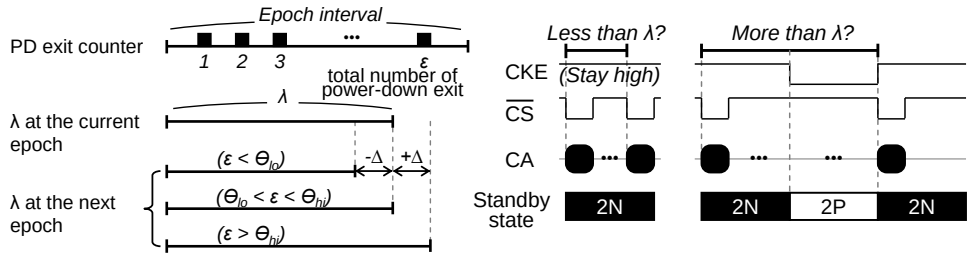
Registered DIMM (RDIMM), which repeats command/address signals with a buffer, is a must to servers because the number of attached DRAM devices per channel often surpasses several dozens, often reaching a few hundreds. When the number of ranks per channel increases, the signal integrity of data I/Os gets worsened as well, enforcing the channel to be operated at lower data transfer rates. Load-Reduced DIMM (LRDIMM) has data buffer (DB) chips placed between its DRAM device and a memory controller outside of the module. These DB chips reduce channel load seen by both the controller and DRAM devices and hence increase data transfer rates compared to the modules without them [20]. Adding data buffers increases all three components of data transfer energy as data I/Os are repeated (E_{AC} and E_{DC}) and a data buffer itself consumes energy internally for re-timing signals regardless of the values being repeated. For example, for a channel with 2 DIMMs and 4 ranks per DIMM (8 ranks total), the E_{AC} increases by several times compared

to the two-rank case reflecting the deteriorated signal integrity (Figure 4). Therefore, compared to the two-rank case, the absolute amount of energy saved by the DC/AC energy components are increased. The static power consumed by the DB chips is much lower than the static power of DRAM chips and not presented in Figure 4(c). Recently, TSV-RDIMM [35] is introduced as an alternative to LRDIMM. It 3D-stacks multiple (4 or 8) DDR4 dies and packages them as a single chip. Each chip has a master die, which serves the role of a data buffer as well. This buffering increases tCL of TSV-RDIMM by 2tCK, which is equal to the overhead due to the data buffer in LRDIMM. However, TSV-RDIMM has following advantages. First, data buffers repeat signals at the package level whereas the master die repeats signals to/from TSVs through micro-bumps. Package-level repeating consumes more power because pads and bumps have higher impedance values than TSVs and micro-bumps. From the data I/O perspective, TSV-RDIMM makes the cost of an 8 rank configuration the same as the two-rank case without data buffers, becoming more energy efficient then LRDIMM. Second, only one DLL and I/O buffers are needed per package, amortizing their power overheads. Third, because all the dies within a package are locked in clock, tRTRS within the die is 0tCK. This is useful because a server memory channel typically has several ranks and non-zero tRTRS values lower random access performance.

Chapter 4

Improving Main-Memory Efficiency Without Compromising Performance: Exploiting Power-Down Modes Adaptively

There have been proposals to exploit the power-down (PD) mode for saving DRAM static power, but with limited success. Entering/exiting the PD mode for every command causes excessive performance degradation due to the tCKE and tXP constraints explained in Section 3.2. Hur et al. [11] suggested enforcing a rank to stay in a standby mode at least for a certain time period (time-out) utilizing a per-rank counter, which being reset on every command to the rank. Even if the counter expires, the rank does not enter the PD mode if there is any pending request to the rank in the memory controller. However, details of specifying its duration are missing in [11]. Ahn et al. [1] suggested making a DRAM rank enter a PD mode when all the banks are at the precharge state. Although



(a) Variable time-out value (λ) (b) Timing diagrams

Figure 5. An adaptive power-down scheme that determines the variable time-out value (λ) based on rank's access history and its timing diagrams and states.

reasonable, it is applicable only to the closed-page management scheme, not even considering more recent adaptive schemes [13, 22]. In this thesis we propose a simple but effective scheme to better exploit the PD mode with minimal performance penalty. It adaptively changes the time-out value (λ) of Hur et al. [11] based on the access history of a rank (Figure 5). This per-rank epoch-based scheme counts the number of PD exits (ε). If e is above a certain threshold (θ_{hi}), it means that the rank enters the PD state too hastily, and hence the scheme increases λ by Δ at the next epoch. If ε is below (θ_{lo}), it is likely that the rank exits the PD state too slowly, so the scheme decreases λ by Δ at the next epoch. Otherwise, λ stays unchanged. λ changes within the range of (λ_{min}, λ_{max}). These rules are based on the following observations. Because there exists correlation between memory access patterns over time, adaptive memory scheduling policies [13, 14] are effective and so is this history-based PD management scheme. When a rank is busy serving requests, it is unlikely that the rank has no pending request. When it is mostly idle, it is better to stay at a PD mode. For both cases, the rank enters/exits the PD mode infrequently and it is better not to increase λ. By making θ_{hi} larger than θ_{lo}, we can make ε

oscillate less frequently. If λ goes up or down too far, it cannot return to its optimal value quickly on memory access pattern changes. Therefore, the range of λ (λ_{min}, λ_{max}) is required. The implementation cost of the proposed scheme, called Ad-PD is low. In addition to [11], one more register is needed per rank to hold ε and one register per channel to set an epoch. Throughout extensive simulation, we empirically set (epoch interval, λ_{min}, λ_{max}, θ_{lo}, θ_{hi}, Δ) as (20us, 30ns, 2us, 2, 5, 10ns) on the CMP system specified in Section 5. Unlike the proposal in [11] which uses a fixed and predetermined lambda, our scheme changes it dynamically over time. Our proposal is also different from RAMZzz [45] in that RAMZzz collects the histogram of idle periods (interval between two commands) over a much longer epoch (in the order of dozens of milliseconds) to adjust λ. While **Ad-PD** has much lower hardware complexity (a counter) compared to RAMZzz (80KB storage) per rank, **Ad-PD** tracks changes in memory access behaviors more nimbly and effectively improves the energy efficiency of the evaluated system, as quantified in Section 6.

Chapter 5

Experimental Setup

We simulated a chip-multiprocessor (CMP) system with DDR4 from two manufacturers (**A** and **B**) and the modified LPDDR4 (**LP4'**) to evaluate their performance and energy efficiency on multi-programmed and multi-threaded workloads; DDR4 from **C** and **B** has similar power consumption. Table 2 tabulates the default parameters of the simulated system. DRAM timing, dynamic energy, and static power values are listed in Table 3. Each ECC DIMM uses ×4 DRAM devices, where their per-pin data transfer rate is 2400Mbps. **LP4'** was modeled following the methodology described in Section 2.2 and 3.1; its VDD is 1.1V, same as LPDDR4, whereas it uses the I/O of DDR4 and (datapath width and page size) are (×4, 512 bits), instead of (×16 2,048 bits). RDIMMs are used for 2-rank configuration, and LRDIMMs or TSV-RDIMMs are used for 8-rank configuration. Dynamic energy and static power of **B-TSV** are estimated based on **B** with the overheads (e.g., additional data transfer energy through TSVs) detailed in [35] applied. A modified version of McPAT [28]

Resource	Value
Number of (cores, MCs)	(16, 4)
Coherence policy	MOESI
Per core:	
(Frequency, issue/commit width)	(3.6GHz, 4/4)
Issue policy	Out-of-Order
L1 I/D cache size/associativity	16KB/4
L2 cache size/associativity	1MB/16
L1, L2 cache line size	64B
Hardware (linear) prefetch	On
Per memory controller (MC):	
(Number of channels, Req. Q size)	(1, 32)
(Capacity per rank, BW)	(16GB, 19.2GB/s)
Scheduling policy	PAR-BS [34]
DRAM page policy	Adaptive open [12]

Table 2. Default parameters of the simulated system.

Parameter		A	B	B-TSV	LP4'
tRCD	(ns)	13.3	13.3	13.3	18.0
tCL	(ns)	15.0	15.0	15.0	24.2
tRAS	(ns)	32.0	32.0	32.0	42.0
tRP	(ns)	13.3	13.3	13.3	18.0
tRTRS	(ns)	0.8	0.8	0	5.0
$E_{ACT+PRE}$	(nJ)	12.0	14.5	14.5	13.4
$E_{RD/WR}$	(nJ)	6.45	4.52	5.0	5.53
$P_{standby}$	(W)	1.22	0.61	0.46	0.43

Table 3. DRAM timing, dynamic energy, and static power values. **B-TSV** is the TSV-RDIMM [35] from the manufacturer **B**, while **LP4'** is the modified LPDDR4. $P_{standby}$ is the standby power of one DRAM rank.

was used for modeling a CMP fabricated at the 14nm technology, where the processor dissipates 25W in idle. We modified McSimA+ [2] to support the various power-down (PD) modes including CAL and the adaptive PD scheme.

SPEC CPU2006 [10] benchmark suite was used for multiprogrammed workloads. We used Simpoint [38] to identify and use the most representative simulation point of each application, which consists of 100M instructions. We categorized the SPEC applications based on the memory access per kilo instruction values

and composed two mixes based on their memory bandwidth demands; mix-high consists of two instances of mcf, milc, leslie3d, soplex, GemsFDTD, libquantum, and lbm, and one instance of omnetpp and sphinx3; mix-blend selects 16 applications randomly and assigns one instance each to cores from perlbench, bzip2, gobmk, dealII, bwaves, zeusmp, sjeng, h264ref, astar, xalancbmk, mcf, milc, GemsFDTD, lbm, omnetpp, and sphinx3. We reported aggregate IPC for multi-programmed workloads as they closely tracked the weighted speedup [40] values. For multi-threaded workloads, we ran the regions of interest of MICA [29] (a high-performance key-value store), fluidanimate in PARSEC [7] and LU in SPLASH-2X [44]. MICA is configured to run at the exclusive read/write and full LRU mode with 128B evenly distributed keys and 1024B values. LU and fluidanimate use simlarge datasets.

Chapter 6

Evaluation

We evaluate the performance (IPC) and energy efficiency (energy-delay product (EDP)) of exploiting low-power mobile DRAM technologies, 3D stacking, various power-down modes for static power saving, and data bus inversion for dynamic energy saving using multi-programmed and multithreaded workloads on the simulated chip multiprocessor systems. Figure 6 shows the relative IPC and EDP as well as power breakdown of the workloads with DDR4 from **A**, **B**, **B** with TSV-RDIMM (**B-TSV**), and LPDDR4' (**LP4'**). We make the following key observations. First, compared to the system with less power-efficient DDR4 from **A**, the system with **LP4'** provides better (lower) EDP over the tested multi-programmed and multi-threaded workloads when 8 ranks are populated per channel. With 8 ranks per channel, **A** dissipates large static power from DRAM devices. Even if **LP4'** performs worse than DDR4 due to larger timing parameter values, its superior energy efficiency leads to better EDP. However, with fewer ranks populated (reflecting more popular

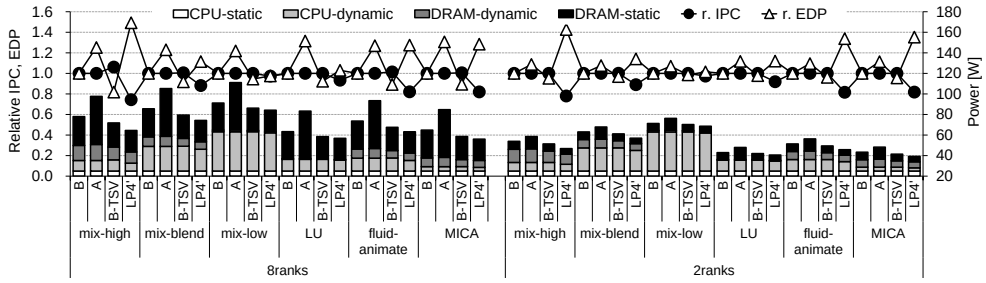


Figure 6. Relative IPC (higher is better) and EDP (lower is better) as well as power breakdown of multi-programmed and multi-threaded workloads on the simulated chip multiprocessor systems with DDR4 from **A**, **B**, **B** with TSV-RDIMM (**B-TSV**), and LPDDR4' (**LP4'**). We set **B** as baseline for a given application and ranks per channel.

datacenter systems), the DRAM static power portion decreases on **A** whereas the performance gap between DDR4 and **LP4'** gets widened, and hence **LP4'** is worse than **A** in EDP. Second, the system with more power-efficient DDR4 from **B** is consistently superior to **A** and **LP4'**. **B** and **A** have the same timing values, so the more energy-efficient, the better EDP. **LP4'** dissipates lower power than **B**, but the gap between two is smaller than that between **LP4'** and **A**, and hence the impact of lower performance of **LP4'** is larger than the difference in power consumption to EDP. Third, lowering DRAM dynamic energy by utilizing TSV-RDIMM is effective. **B-TSV** consumes less power than already energy-efficient **B**. TSV-RDIMM is more effective on the 8-rank configuration because the other DRAM devices are augmented with data buffer (DB) chips to retain the data transfer rate at the worse signal integrity, which increases DRAM static power noticeable. Moreover, TSV-RDIMM brings performance gain as well because there is no tRTRS penalty in the memory channel ownership changes between ranks that are stacked together in a TSV-RDIMM. The evaluated power-down (PD) schemes are effective in power

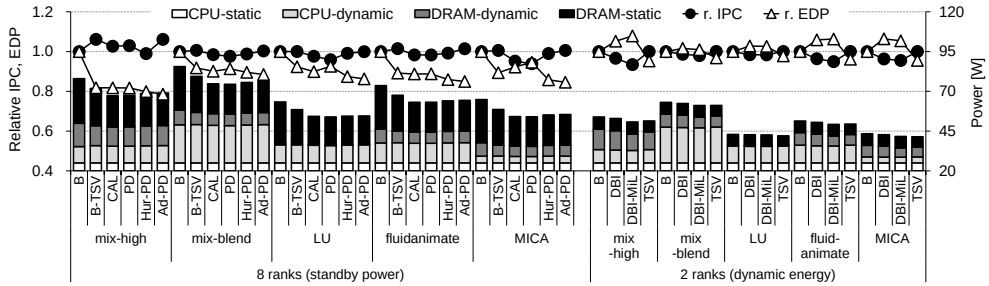


Figure 7. Relative IPC and EDP as well as power breakdown of the workloads on the simulated systems with 2 and 8 ranks per memory channel. On the 8 rank systems, the baseline (**B**), **B** with TSV-RDIMM without power-down (PD) (**B-TSV**), **B-TSV** with CAL (**CAL**), **B-TSV** with PD of [1] (**PD**), **B-TSV** with PD of [11] (**Hur-PD**), and **B-TSV** with adaptive PD (**Ad-PD**). On the 2 rank systems, the baseline (**B** without DBI), **B** with DBI in DDR4 (**DBI**), **B** with DBI proposed in [42] (**DBI-MiL**), and TSV-RDIMM without DBI (**TSV**) are compared.

reducing DRAM power consumption, among which **CAL** dissipates the smallest power, whereas the proposed adaptive PD scheme (**Ad-PD**) achieves the best (lowest) EDP. Figure 7(left) shows the relative IPC and EDP as well as power breakdown of the multi-threaded and multi-programmed workloads on memory-capacity demanding systems with 8 ranks per memory channel with the baseline without **PD** (**B**), **B** with TSV-RDIMM (**B-TSV**), **B-TSV** with CAL (**CAL**), **B-TSV** with PD of Ahn et al. [1] (**PD**), **B-TSV** with PD of Hur et al. [11] (Hur-PD), and **B-TSV** with the proposed PD scheme (**Ad-PD**). **CAL** makes a DRAM device stay in a PD mode (IDD2NL) most frequently as it just turns the input/output buffers on when commands are delivered to the device, reducing the DRAM power most. However, **CAL** makes every DRAM command experience an additional delay of tCAL. **PD** enforces a rank to stay at a standby mode, where it consumes more static power but can receive commands without latency penalty; anytime the corresponding controller has at least a pending request to the rank [1]. This decreases average memory access latency for memory intensive workloads, such as mix-high,

compared to **CAL** as **PD** experiences PD exit penalty less frequently. On the contrary, **PD** suffers from PD enter/exit latency penalty, which is several times higher than tCAL, on applications with light/medium bandwidth demands, such as mix-blend and LU. Therefore, there is no clear winner between **PD** and **CAL** in EDP. **Ad-PD** performs better than **PD** because the former suffers less frequently from the hasty PD entries, the phenomenon explained in Section 4.1, and results in better EDP for the tested applications. For example, **Ad-PD** is better than **B-TSV**, **CAL**, **PD** in EDP on mix-high by 4.0%, 4.0%, and 3.9%, and on LU of SPLASH-2X by 6.6%, 4.1%, and 6.9%.

Our simulation results show that using data bus inversion (DBI) techniques are not effective in EDP with the latency penalty specified in the DDR4 standard. We used the two-rank configurations as DRAM dynamic power takes more portion of total system power with fewer ranks populated. Even if the DDR4 standard does not define DBI for ×4 devices, we assume that DBI is implemented with an additional DBI pin per DRAM device and denote it by **DBI**. Recently, Song et al. [42] proposed More is Less, which utilizes a bandwidth-inefficient but energy-efficient DBI code when channels are lighted loaded, and another bandwidth-efficient but less-energy-efficient code for heavily loaded cases. To understand the upper-bound of its energy savings, we model the I/O energy of the energy-efficient (3-LWC [42], only up to three zeros in a 8-bit group of data burst) code and the bandwidth penalty of the bandwidth-efficient (MiLC [42], burst length 10 instead of 8) code, and denote it by **DBI-MiL**. The latency penalty in tCL is 3tCK for both **DBI** and **DBI-MiL**. As shown in Figure 7(right), **DBI** and **DBI-MiL** achieve always higher (worse) EDP values than the baseline. Both lower DRAM power but performance penalty due to increased tCL outweighs the power

saving, leading this worse EDP. **DBI-MiL** further decreases DRAM I/O energy compared to **DBI**, but its longer burst length (10 instead of 8) exacerbates performance for memory intensive workloads such as mix-high, whereas DRAM I/O energy saving takes a very small portion of system power/energy for applications with medium to low main memory bandwidth demands.

Chapter 7

Conclusion

Mainstream DRAMs for servers/desktops have adopted the advantages of fabrication technologies, circuit techniques, and microarchitectures used by popular graphics or mobile DRAMs. Based on this observation, we demonstrated that the prior proposal applying mobile DRAMs to big-memory servers becomes ineffective due to insufficient energy saving over performance penalty that increases the energy consumption of other system components such as CPU. Thus, we paid more attention to other energy saving techniques introduced by the latest DDR4. Especially, we found that the data transfer energy saving by data bus inversion (DBI) does not overcome the energy overhead induced by performance penalty, whereas exploiting power-down (PD) modes pays off the cost of PD entrance/exit latencies as it reduces DRAM standby power, a major portion of DRAM power consumption for big-memory servers. Subsequently, we proposed simple but effective PD scheme and improved *system-level energy-delay product* by 4.0% over the default PD schemes on memory-intensive multiprogrammed

workloads. Lastly, we analyzed and quantified the benefits of combining our proposals with TSV-RDIMM on performance and energy efficiency for big-memory servers.

Bibliography

[1] J. Ahn et al., “Future Scaling of Processor-Memory Interfaces,” in SC, 2009.

[2] J. Ahn et al., “McSimA+: A Manycore Simulator with Application level+ Simulation and Detailed Microarchitecture Modeling,” in ISPASS, 2013.

[3] Apache, “Apache Storm homepage.” [Online]. Available: https://storm.apache.org/

[4] S.-J. Bae et al., “An 80 nm 4 Gb/s/pin 32 bit 512 Mb GDDR4 Graphics

DRAM With Low Power and Low Noise Data Bus Inversion,” JSSC, 2008.

[5] L. A. Barroso et al., “The Datacenter as a Computer: An Introduction to the Design of Warehouse-scale Machines,” Synthesis lectures on computer architecture, vol. 8, no. 3, pp. 1–154, 2013.

[6] I. Bhati et al., “Flexible Auto-refresh: Enabling Scalable and Energy efficient DRAM Refresh Reductions,” in ISCA, 2015.

[7] C. Bienia et al., “The PARSEC Benchmark Suite: Characterization and Architectural Implications,” in PACT, 2008.

[8] M. Ferdman et al., “Clearing the Clouds: A Study of Emerging Scale-out Workloads on Modern Hardware,” in ASPLOS, 2012.

[9] Y. Han et al., “Data-aware DRAM Refresh to Squeeze the Margin of Retention Time in Hybrid Memory Cube,” in ICCAD, 2014.

[10] J. L. Henning, “SPEC CPU2006 Memory Footprint,” Computer Architecture News, vol. 35, no. 1, 2007.

[11] I. Hur et al., “A Comprehensive Approach to DRAM Power Management,” in HPCA, 2008.

[12] Intel, Intel Xeon Processor 7500 Series Datasheet, 2010.

[13] Intel, Intel Xeon Processor E5-1600/2400/2600/4600 (E5-Product Family) Product Families Datasheet, 2012.

[14] E. Ipek et al., “Self-Optimizing Memory Controllers: A Reinforcement Learning Approach,” in ISCA, 2008.

[15] B. Jacob, The Memory System: You Can’t Avoid It, You Can’t Ignore It, You Can’t Fake It. Morgan and Claypool Publishers, 2009.

[16] B. Jacob et al., Memory Systems: Cache, DRAM, Disk. Morgan Kaufmann Publishers Inc, 2007.

[17] JEDEC, “Graphic Double Data Rate 5 (GDDR5) Specification,” 2009.

[18] JEDEC, “Double Data Rate 3 (DDR3) Specification,” 2010.

[19] JEDEC, “DDR4 SDRAM Specification,” 2012.

[20] JEDEC, “JEDEC Standard: DDR4 SDRAM Load Reduced DIMM (LRDIMM) Design Specification,” 2014.

[21] JEDEC, “Low Power Double Data Rate 4 (LPDDR4) Specification,” 2014.

[22] D. Kaseridis et al., “Minimalist Open-page: A DRAM Page-mode Scheduling Policy for the Many-core Era,” in MICRO,

2011.

[23] B. Keeth et al., DRAM Circuit Design, 2nd ed. IEEE, 2008.

[24] Y. Kim et al., “Thread Cluster Memory Scheduling: Exploiting Differences in Memory Access Behavior,” in MICRO, 2010.

[25] Y. Kim et al., “A Case for Exploiting Subarray-Level Parallelism (SALP) in DRAM,” in ISCA, 2012.

[26] A. R. Lebeck et al., “Power Aware Page Allocation,” in ASPLOS, 2000.

[27] H.-W. Lee et al., “Survey and Analysis of Delay-Locked Loops Used in DRAM Interfaces,” VLSI, 2014.

[28] 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.

[29] H. Lim et al., “MICA: A Holistic Approach to Fast In-Memory Key-Value Storage,” in NSDI, 2014.

[30] K. T. Malladi et al., “Towards Energy-proportional Datacenter Memory with Mobile DRAM,” in ISCA, 2012.

[31] K. T. Malladi et al., “Rethinking DRAM Power Modes for Energy Proportionality,” in MICRO, 2012.

[32] Micron, “Micron DRAM products.” [Online]. Available: http://www.micron.com/products/dram

[33] J. Mukundan et al., “Understanding and Mitigating Refresh Overheads in High-density DDR4 DRAM Systems,” in ISCA, 2013.

[34] O. Mutlu et al., “Parallelism-Aware Batch Scheduling: Enhancing both Performance and Fairness of Shared DRAM Systems,” in ISCA, 2008.

[35] R. Oh et al., “Design technologies for a 1.2 V 2.4 Gb/s/pin high capacity DDR4 SDRAM with TSVs,” in VLSI Circuits Digest of

Technical Papers, Symposium on, 2014.

[36] S. Rixner et al., “Memory Access Scheduling,” in ISCA, 2000.

[37] Samsung, “Samsung Server DRAM products.” [Online]. Available: http://www.samsung.com/semiconductor/products/dram/server-dram/

[38] T. Sherwood et al., “Automatically Characterizing Large Scale Program Behavior,” in ASPLOS, 2002.

[39] SK Hynix, “SK Hynix Computing solution products.” [Online]. Available: https://www.skhynix.com/eng/product/solComputing.jsp

[40] A. Snavely et al., “Symbiotic Job Scheduling for a Simultaneous Mutlithreading Processor,” in ASPLOS, 2000.

[41] Y. H. Son et al., “Microbank: Architecting Through-Silicon Interposer-Based Main Memory Systems,” in SC, 2014.

[42] Y. Song et al., “More is Less: Improving the Energy Efficiency of Data Movement via Opportunistic Use of Sparse Codes,” in MICRO, 2015.

[43] T. Vogelsang, “Understanding the Energy Consumption of Dynamic Random Access Memories,” in MICRO, 2010.

[44] S. C. Woo et al., “The SPLASH-2 Programs: Characterization and Methodological Considerations,” in ISCA, 1995.

[45] D. Wu et al., “RAMZzz: Rank-aware DRAM power management with dynamic migrations and demotions,” in SC, 2012.

[46] D. H. Yoon et al., “BOOM: Enabling Mobile Memory Based Low-Power Server DIMMs,” in ISCA, 2012.

[47] D. H. Yoon et al., “Adaptive Granularity Memory Systems: A Tradeoff between Storage Efficiency and Throughput,” in ISCA,

2011.

[48] M. Zaharia et al., “Spark: Cluster Computing with Working Sets,” in HotCloud, 2010.

[49] T. Zhang et al., “Half-DRAM: a High-bandwidth and Low-power DRAM Architecture from the Rethinking of Fine-grained Activation,” in ISCA, 2014.

[50] T. Zhang et al., “CREAM: a Concurrent-Refresh-Aware DRAM Memory Architecture,” in HPCA, 2014.

[51] H. Zheng et al., “Mini-Rank: Adaptive DRAM Architecture for Improving Memory Power Efficiency,” in MICRO, 2008.

국문 초록

최근 서버에 요구되는 주기억장치의 용량이 증가되면서 기존에 비해 많은 개수의 기억장치 모듈이 추가적으로 장착되기 시작하였다. 이로 인해 대용량 주기억장치를 갖춘 서버 시스템에서 주기억장치가 프로세서에 이어 두 번째로 많은 에너지를 소모하는 구성 성분이 되었다. 게다가 특정 서버에서는 시스템 구성 방법에 따라서는 주기억장치가 프로세서에 맞먹는 에너지를 소모하는 경우까지 있다. 따라서 대용량 주기억장치를 가진 서버 시스템에서 주기억장치의 에너지 효율을 높이는 것이 매우 중요해졌다. 기존의 연구들은 보다 에너지 효율적인 주기억장치 시스템을 구성하기 위해서 모바일용 DRAM인 LPDDR을 활용하려고 하였다. LPDDR은 기존 DDR 대비 전력 소모가 적다는 장점이 있다. 그러나 대신 데이터 접근 지연시간이 너무 크고 대역폭이 낮다는 단점도 동시에 가지고 있다. 따라서 에너지 효율을 높이기 위하여 성능 제약을 극복하려고 애써왔다. 하지만 본 논문에서 DDR4대신 LPDDR4를 기반으로 모바일 DRAM을 대신 사용하는 주기억장치 아키텍처가 더 이상 효과적이지 않다는 것을 실험으로 확인하였다. 주기억장치를 빈번하게 사용하는 워크로드에서는 기준점인 DDR4 대비 LPDDR4를 사용하는 시스템의 에너지 효율이 49% 감소한다. 그 이유는 DDR4가 모바일과 그래픽용 DRAM의 장점(낮은 전력 소모, 높은 대역폭, 많은 뱅크 등)을 벤치마킹하여 적용함으로써 성능과 에너지 효율을 동시에 개선하고자 하였으나, LPDDR4에서 더 높은 대역폭 확보를 위해 대신 에너지 효율을 희생하였기 때문이다. 추가적으로 DDR4의 전력 소모가 제조사별로 산포가 존재하는 것을 확인하였다. 그리고 DDR4의 새로운 에너지 소모 감소 기

술에 대하여 심도 있게 조사하였다. 그래서 이 기술들을 적용하였을 경우 에너지 효율이 오히려 나빠질 수 있다는 것을 실험으로 확인하였다. 앞서 나열한 사항에 근거하여, 궁극적으로 에너지 소모 감소를 위하여 가변적으로 DRAM의 power-down 모드를 활용하는, 간단하고 효과적인 방법을 제안한다. 제안하는 방법을 적용하였을 경우 에너지-지연시간의 곱이 기존 power-down 대비 4% 개선됨을 확인하였다.

주요어: 메모리 시스템, DDR4 SDRAM, 전력/에너지 감소, 지연시간, DBI, 3차원 적층, TSV
학 번: 2015-26051