ABSTRACT

VENKATESAN, RAVI KRISHNAN. Power-Scalable Memory: Exploiting Typical Charge Retention in DRAM and Charge-Voltage Decoupling in ZettaRAM. (Under the direction of Dr. Eric Rotenberg).

DRAM faces two fundamental power scaling challenges in the future. Firstly, DRAM requires continuous refresh because of charge leakage, resulting in power consumption even during standby operation. As DRAM capacity quadruples with each new generation, standby power grows correspondingly and becomes a significant fraction of system power. Secondly, we are fast approaching a lower limit on DRAM voltage scaling because charge in a DRAM capacitor depends directly on voltage, and a minimum amount of charge is needed for reliable operation.

Scalable standby power is achieved by exploiting typical charge leakage instead of worst-case charge leakage. Most DRAM cells have very low leakage currents and consequently long retention times. However, sparse outliers have high leakage currents and short retention times, and are the cause for frequent refreshes of the whole DRAM. More generally, retention times of DRAM cells and even whole pages vary widely, with an average retention time on the order of seconds. We propose *Retention-Aware Placement in DRAM (RAPID)*, novel software techniques that can exploit off-the-shelf DRAMs to reduce refresh power to vanishingly small levels. The key idea is to favor allocation of longer-retention pages over shorter-retention pages, and then select a single refresh period that depends on the shortest-retention page among populated pages instead of the shortest-retention page overall. This refresh period is much higher than the default refresh period of commodity DRAMs, thereby significantly reducing the frequency of refreshes. Thus, RAPID reduces DRAM standby power to near-zero levels, yielding favorable standby power scaling despite quadrupling of DRAM capacity from one generation to the next.

To overcome DRAM voltage scaling limits, we explore the long-term power-scalability of ZettaRAM™, a nascent DRAM technology. ZettaRAM is based on conventional DRAM architectures but replaces the DRAM capacitor with a new molecular capacitor. The molecular capacitor is fully charged/discharged if the applied voltage is above/below a discrete threshold voltage. Therefore, unlike a conventional capacitor, the *amount* of charge deposited on the molecular capacitor is independent of applied voltage. Charge-voltage decoupling holds the key for viable voltage scaling from one generation to the next and opens up two unprecedented power scaling opportunities.

(1) *Exploiting molecular engineering for long-term power scalability*: Read/write operations can be performed at lower voltages, while still maintaining the minimum amount of charge needed for reliable sensing. Precise tuning of molecular attributes provides an inexpensive path for scaling voltage, hence power, from one generation to the next, whereas conventional DRAM requires major cell redesigns to maintain a fixed charge while scaling voltage.

(2) *Intelligent management of ZettaRAM*: While the fixed charge is voltage-independent, speed is voltage-dependent. Thus, the applied voltage is padded to achieve the same speed as DRAM. A key architectural insight is leveraged to manage the speed-voltage dependence and lower the voltage even further: most of the bitline activity is caused by non-critical L2-cache writeback requests. Accordingly, slow operations (lower voltage) are applied to non-critical writebacks and fast operations (higher voltage) to critical fetches. This hybrid policy combines the power efficiency of uniformly slow operations with the high performance of uniformly fast operations, further extending ZettaRAM's power scaling advantage.

The combination of RAPID and ZettaRAM leads to a power-scalable form of DRAM, extending the roadmap of this important memory technology.

POWER-SCALABLE MEMORY: EXPLOITING TYPICAL CHARGE RETENTION IN DRAM AND CHARGE-VOLTAGE DECOUPLING IN ZETTARAM

by

RAVI KRISHNAN VENKATESAN

A dissertation submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

COMPUTER ENGINEERING

Raleigh

2006

APPROVED BY:

_____

Dr. Eric Rotenberg, Chair of Advisory Committee

_____ _____

Dr. W. Rhett Davis Dr. Yan Solihin

_____

Dr. Vincent W. Freeh

தொட்டனைத் தூறும் மணற்கேணி மாந்தர்க்குக்
கற்றனைத் தூறும் அறிவு

– திருவள்ளுவர்

“Deeper you dig the sand, the higher a spring flows, the
ardor you devote to learning, the fuller the knowledge”

To My Dear Mum, Dad and sister Meenakshi……

They have been a constant source of inspiration throughout my life

BIOGRAPHY

Ravi Krishnan Venkatesan was born on Decembee 22, 1976 in the city of Madras, India to Natarajan Venkatesan and Mrs. Leela Venkatesan. He received his schooling at Padma Seshadri Bala Bhavan, Madras. He was awarded the National Merit Scholarship for his high school and undergraduate studies. He obtained his Bachelors degree from Indian Institute of Technology (IIT) Madras, India in May 1999 and his Masters degree in Computer Engineering from North Carolina State University, Raleigh, NC, USA in May 2001. He worked as a PowerPC Development Engineer at IBM Microelectronics Division, RTP, from 2001-2002. Since January 2003, Ravi has been pursuing his PhD degree in Computer Engineering at North Carolina State University under the guidance of Dr. Eric Rotenberg on processor and memory architectures with focus on developing architectural techniques to achieve low-power memories. His interests include sports, photography, hiking, and traveling.

Upon graduation, Ravi is moving back to India and joining the Mobile Platforms Architecture Division at Intel, India, as a senior platform architecture engineer.

ACKNOWLEDGEMENTS

Working on my PhD degree has been a fun and rewarding experience and I would like to take this opportunity to thank all of those peope who have offered me their grestest support in making my PhD journey at North Carolina State University a successful one.

First and foremost, I would like to thank my advisor, Dr. Eric Rotenberg for his guidance throughout my doctoral degree. This dissertation would have been possible without his excellent guidance and support. Eric has not only been a great advisor to work with, but also a very good teacher and friend. Eric's work ethic, enthusiasm and attention to detail is unbelievable and second to none. His unending drive and dedication to the field of computer architecture never ceases to amaze me. Eric has allowed me to work independently, yet, always making sure that I was on the right track. Eric has always urged me to focus on the big picture and the crux of the problem and to effectively convey the "big picture" idea to the readers. Eric is also an excellent writer and I have learned a lot about writing from him. Working with Eric has taught me the true meaning of perfection. I have been privileged to have Eric as my advisor, mentor, and friend and I will strive to make him proud by excelling in my future endeavors.

Words cannot adequately describe the contributions of my dear parents, Venkatesan and Leela, and my sister Meenakshi towards my success. They have constantly cheered and inspired me on this wonderful and exciting academic journey. Thank you for always being there to lift my spirits whenever things looked tough.

I would like to thank my committee members, Dr. Yan Solihin, Dr. Vince Freeh and Dr. Rhett Davis. Their feedback and comments have been invaluable and has helped me refine my dissertation.

I would like to thank Ken Mobley and Dr. Werner Kuhr (ZettaCore Inc.) for valuable discussions on modeling the molecular capacitor. I also would like to thank Dr. Rhett Davis for insights on multiple supply voltages and a brief crash course/tutorial on statistical analysis, Dr. Vince Freeh for letting me try out the WattsUp Pro power meter, and Dr. Alex Dean for providing the oscilloscope for power measurements and multimeter with thermistor for monitoring temperature of the DRAM chip. I also would like to thank Rudy Salas for providing the heat gun for raising the temperature of the DRAM chip.

Krishnan Siva, an undergraduate student, assisted me in characterizing different molecules as part of the ZettaRAM project from May-August 2004. Stephen Herr, also an undergraduate student, was actively involved in the RAPID project from May-August 2005 and assisted me in designing the test strategies and characterizing retention times of DRAM pages. Ahmed AL-Zawawi, my fellow research mate at the CESR lab, was actively involved in designing the SPICE models for the ZettaRAM molecular capacitor.

I wish to express my sincere thanks to the fabulous *´CESR's ROman´* squad for making my stay at North Carolina State University an enjoyable one. I would like to thank Ahmed-AL-Zawawi, Ali El-Haj-Mahmoud, Aravindh Anantaraman, Hashem Hashemi, Muawya Al-Otoom, and Vimal Reddy for providing me insightful feedback during my practice talks.

I would to thank Sandy Bronson, CESR's administrative assistant for taking care of all my paperwork.

I would like to thank my partners in crime, Prakash and Prasanna for enduring so many WolfPack heartbreakers at Carter-Finley, RBC Center and on TV in the last four years.

To all my other friends in Raleigh, thank you for the great many enjoyable moments over these past few years.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

Introduction

Dynamic Random Access Memory (DRAM) is the predominant volatile memory technology in modern computer systems. This dissertation addresses growing concerns about DRAM power scaling, as density quadruples from one generation to the next. Unfavorable power scaling is due to two factors. Firstly, DRAM must be continuously refreshed because of charge leakage, resulting in power consumption even during standby operation. As DRAM capacity quadruples with each new generation, standby power grows correspondingly and becomes a significant fraction of system power. Secondly, while DRAM voltage has scaled with technology in the past, voltage scaling is not viable indefinitely. The charge in a DRAM capacitor depends directly on voltage, and a minimum amount of charge is needed for reliable operation. Quadrupling DRAM capacity without lowering voltage can lead to severe power problems.

The two factors, charge leakage and charge-voltage dependence, can be directly attributed to the two parts of a DRAM cell. A DRAM cell typically consists of a transistor-capacitor pair (referred to as a 1T-1C cell), as shown in Figure 1-1. Standby power is due to leakage currents in the access transistor. The voltage scaling limit is an artifact of the capacitor, which couples charge (Q) and voltage (V) through capacitance (C), i.e., $Q = CV$.

To overcome the DRAM power scaling challenge, both factors are addressed in this dissertation. Our approach is to identify unique properties of the underlying technology and

exploit them at the system level. Firstly, on the transistor side, we propose a software technique that nearly eliminates standby power. The technique exploits the fact that most access transistors actually have very low leakage currents, hence, most cells have very long retention times. Intelligent placement of data by the operating system can avoid the use of atypical leaky cells, allowing very infrequent memory refreshes. Secondly, on the capacitor side, we explore the long-term power-scalability of ZettaRAM™, an emerging DRAM technology which has its roots in molecular electronics. A ZettaRAM cell is also 1T-1C, except the conventional capacitor is replaced with a molecular capacitor that decouples charge from voltage. Charge-voltage decoupling holds the key for viable voltage scaling from one generation to the next.

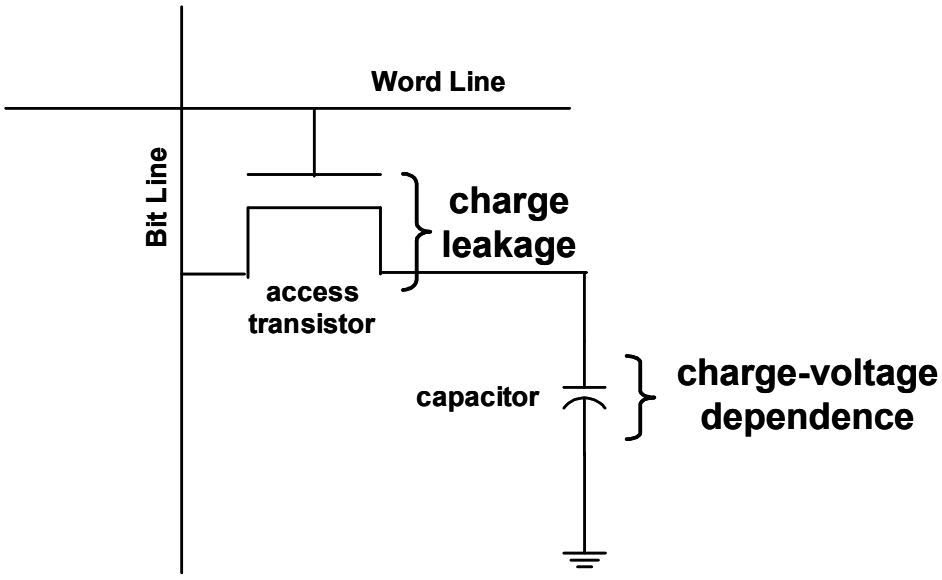


Figure 1-1. 1T-1C DRAM cell.

Specifically, there are two major contributions of this dissertation.

- *Scalable standby power in DRAM.* We propose *Retention-Aware Placement in DRAM (RAPID)*, novel software approaches that can exploit off-the-shelf DRAMs to reduce

refresh power to vanishingly small levels. RAPID exploits retention-time variations among different memory pages in a single DRAM chip. The key idea is to favor allocation of longer-retention pages over shorter-retention pages. A single refresh period is then set, that depends on the shortest-retention page among populated pages, instead of the shortest-retention page overall. This refresh period can be up to two orders of magnitude higher than the default refresh period of commodity DRAMs, significantly reducing the frequency of refreshes. As a result, RAPID reduces DRAM standby power to near-zero levels, yielding favorable standby power scaling despite quadrupling of DRAM capacity from one generation to the next.

- *Overcoming voltage scaling limits of DRAM.* We explore the superior power-scaling characteristics of ZettaRAM, a nascent DRAM technology. ZettaRAM is based on conventional DRAM architectures but the conventional capacitor is replaced with a new molecular capacitor. As a result, the benefits of the underlying DRAM architecture – high density and fast access – are preserved. The unusual characteristic of this new technology is that the molecular capacitor is fully charged or discharged if the applied voltage is above or below a discrete threshold voltage. In other words, while voltage still controls charging and discharging, the *amount* of charge deposited on the molecular capacitor (critical for reliable sensing) is independent of write voltage. In contrast, with a conventional capacitor, voltage has dual roles of controlling charging/discharging and influencing the *amount* of charge. Separating control and amount of charge is key to viable voltage scaling in the future and opens up two unprecedented opportunities for power scaling in a DRAM-like architecture. (1) *Exploiting molecular engineering for long-term power-scalability*: Read/write operations can be performed at much lower

voltages, while still maintaining the minimum amount of charge needed for reliable sensing. The flexibility in engineering of molecular attributes provides an inexpensive path for scaling voltage, hence power, from one memory generation to the next. This dissertation is the first work to understand and characterize the power scalability of ZettaRAM. (2) *Intelligent management of ZettaRAM*: ZettaRAM presents a novel speed/energy tradeoff. While the fixed charge is voltage-independent, speed of charging/discharging is voltage-dependent (faster with higher voltage). Thus, the applied voltage is padded to achieve the same speed as DRAM. We propose architectural techniques to manage this speed/energy tradeoff and lower the voltage even further. The write voltage is dynamically modulated based on the criticality of memory requests, further extending ZettaRAM's power scaling advantage with little system slowdown.

The combination of RAPID and ZettaRAM leads to a power-scalable form of DRAM, extending the roadmap of this important memory technology.

The rest of this chapter is organized as follows. Section 1.1 provides an overview of RAPID for scalable standby power. Section 1.2 contrasts voltage scaling in conventional DRAM and ZettaRAM, to explain superior power scaling of the latter. Section 1.3 summarizes the contributions of this dissertation.

1.1 Scaling Standby Power: Retention-Aware Placement in DRAM (RAPID)

DRAM refresh (needed to preserve stored information) continuously drains battery energy even in standby operation. Detailed power analysis of the prototype ITSY computer

[68], developed at Compaq WRL, shows that, in the deepest standby mode that still preserves the DRAM contents ("sleep"), DRAM consumes a third of the total power. This fraction will only increase in the future with higher memory capacity and even lower power conservation modes.

Figure 1-2 illustrates DRAM density scaling trends published in the International Technology Roadmap for Semiconductors (ITRS) [4]. Notice that DRAM density quadruples with each new generation. With increasing density, more bits need to be refreshed. Therefore, standby power grows correspondingly and becomes a significant fraction of overall system power.

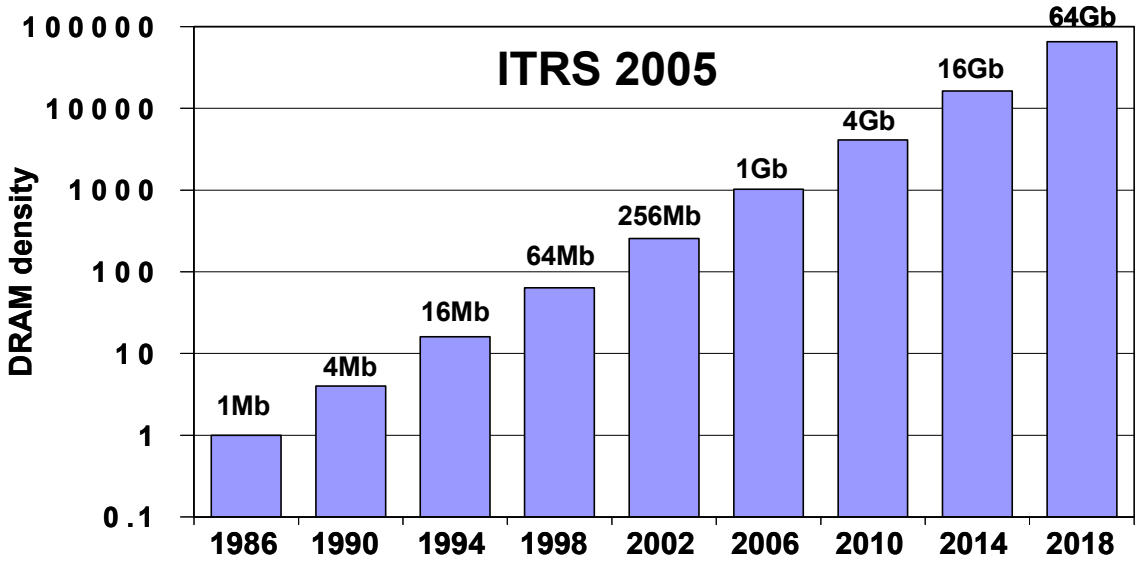


Figure 1-2. DRAM density scaling trends (ITRS).

The key to reducing DRAM refresh power lies with exploiting variations in retention times among different memory cells, rows (pages), or arbitrary groups of cells within a single DRAM chip. A DRAM cell consists of a transistor-capacitor pair. The charge in the DRAM capacitor leaks over a period of time via the transistor. Process variations cause the leakage

current to differ from one cell to another. As a result, different cells retain information for longer or shorter periods of time [19]. Currently, a single worst-case refresh period (typically 64 milliseconds) is selected based on the cell with the shortest retention time. The worst-case refresh period also accounts for worst-case temperature (leakage increases with temperature). Limited support for temperature-compensated refresh (TCR) is now being deployed in some DRAMs [8][64]. However, TCR does not exploit retention-time variations among different cells, i.e., all cells are still treated uniformly with a single temperature-dependent refresh period, determined by the worst cell. Several researchers have recognized the opportunity presented by retention-time variations, proposing custom DRAM support for refreshing different cells or blocks of cells at different refresh rates [25][33][34][35][39][48][52][70]. However, to our knowledge, this approach has not been implemented in commodity DRAMs.

Instead of custom DRAM support, we propose *Retention-Aware Placement in DRAM* (RAPID), software methods for exploiting retention-time variations among different pages to reduce the refresh power of off-the-shelf DRAMs to vanishingly small levels, approaching non-volatile memory. The key idea is to allocate longer-retention pages to application programs before allocating shorter-retention pages. A single refresh period is then selected based on the shortest-retention page among pages that are actually populated, rather than the shortest-retention page overall. Note that all pages in memory are still refreshed, as usual, albeit at a higher refresh period. This extended refresh period (lower refresh rate) selected by RAPID is only safe for occupied pages, which is sufficient for overall correctness.

Three versions of RAPID are proposed and explored, described below in order of increasing sophistication.

- *RAPID-1*: Our retention-time measurements of a real DRAM chip reveal a handful of outlier pages with atypically short retention times, consistent with previous retention time characterizations [19]. For example, if the worst 1% of pages are not populated, then a refresh period of 3 seconds is sufficient for the remaining 99% of pages at room temperature (25°C). Likewise, a 1 second refresh period is sufficient at the industry-standard worst-case temperature of 70°C. In contrast, the worst outlier page requires a 500 millisecond (ms) refresh period at room temperature and 150 ms at 70°C. RAPID-1 is a static approach in which the handful of outlier pages (1%) having very short retention times are mapped out and never populated, yielding a very reasonable refresh period (e.g., 3 s at room temperature and 1 s at 70°C) with only a negligible reduction in DRAM capacity (e.g., 1%). For example, in a 128 MB DRAM, nearly 127 MB can still be used. Note that the refresh period set by RAPID-1 is 6X longer than the refresh period based on the worst cell, yielding significant refresh power savings.

- *RAPID-2*: RAPID-2 is a dynamic approach. A list of pages is maintained, sorted according to their retention times. Longer-retaining pages are allocated before shorter-retaining pages. Thus, the refresh period starts out at the highest setting corresponding to the longest-retention page. The refresh period is decreased, as more pages are populated. The refresh period can be increased after a deallocation, only if the shortest-retention page was deallocated. In other words, after every allocation/deallocation, the refresh period is adjusted to the shortest retention time among the populated pages. Once the refresh period has been decreased, increasing it again is opportunistic, since it depends on which pages happen to be freed by application programs.

- *RAPID-3*: RAPID-3 builds on RAPID-2. As longer-retention pages are freed by application programs, data from shorter-retention pages are migrated into the newly freed longer-retention pages. Thus, RAPID-3 continuously reconsolidates data into the longest-retention pages possible. Migration can be naturally implemented as part of our proposed allocation/deallocation routines. A downside is extra power consumption due to migration, which must be weighed against extra refresh power savings. This hints at a deep design space, in terms of designing good criteria and timing for migrating pages.

The refresh period set by RAPID-1, a static approach, is independent of actual DRAM utilization. On the other hand, the refresh periods set by dynamic techniques, RAPID-2 and RAPID-3, are dependent on the dynamic DRAM utilization. The refresh periods selected by RAPID-3 more closely reflect actual dynamic DRAM utilization than RAPID-2, because RAPID-3 continuously reconsolidates data into the longest-retention pages possible. Note that outlier pages are discarded in both RAPID-2 and RAPID-3, so that the refresh power of RAPID-2 and RAPID-3 are at least as good as RAPID-1 refresh power.

To the best of our knowledge, RAPID is the first software-only technique to exploit retention-time variations to nearly eliminate refresh power. We also propose an interrupt-driven refresh implementation, keeping in mind the available refresh options in commodity DRAMs, so that RAPID will work with arbitrary off-the-shelf DRAMs.

To sum up, RAPID reduces DRAM standby power to near-zero levels, yielding favorable standby power scaling. The next section addresses the second power scaling

challenge, the voltage scaling limit of DRAM, and discusses how ZettaRAM can overcome this limit.

1.2 Scaling Voltage: Charge-Voltage Decoupling in ZettaRAM

ZettaRAM is a new memory technology under development by the startup ZettaCore as a potential replacement for conventional DRAM [72]. ZettaRAM had its genesis in a DARPA-sponsored molecular electronics project [57] and one of the goals was to eventually deploy individual charge-storage molecules as 1-bit memory elements and integrate them with other molecular-scale electronics. However, as a near-term goal, ZettaCore currently exploits many molecules in aggregate to create a molecular capacitor which can store charge similar to a conventional capacitor in a DRAM cell. ZettaRAM is a DRAM derivative, based on conventional DRAM architectures – address decoder, wordline, access transistor, bitline, sense amp, etc. The key innovation is replacing the conventional capacitor with the molecular capacitor.

Even though molecules are used in aggregate to form a molecular capacitor, ZettaRAM still retains some of the key benefits of the underlying nanotechnology from which it is derived.

1. *Cost-effective density scaling:* Self-assembly is a process by which, the thousands of molecules that make up a molecular capacitor automatically arrange themselves into a uniform, dense monolayer. Moreover, the charge density of the molecular capacitor is greater than the charge density of the conventional capacitor. These two factors – self-assembly and high charge density – hold the key to increasing the density of DRAM in a cost-effective manner. While conventional DRAM may certainly scale, the costs would

be potentially astronomical due to the need for many complex masks for constructing elaborate three-dimensional capacitor structures. These "tall" structures are needed in order to reduce the cell area, while maintaining an amount of charge that can be sensed. On the other hand, the aggregate molecular capacitor does not require complex masks because the same amount of charge (or more) can be packed into the desired area via a self-assembled monolayer.

2. *Precise control of molecules' attributes:* The new molecular capacitor benefits from "molecular engineering" – engineering the properties of constituent molecules with remarkable precision. Synthetic chemists can precisely tune key properties of the molecules through the choice of molecular "groups" and "linkers", such as the speed with which electrons can be added/removed (affecting read and write speed), the voltage at which electrons can be added/removed (affecting read and write power consumption), and monolayer density (affecting charge density and thus overall memory density). Engineering a molecule is also highly precise, predictable/repeatable, and can be done in inexpensive laboratories, whereas tuning bulk properties of semiconductors is expensive and subject to many environmental factors. And there is tremendous flexibility in the selection of performance (by way of electron transfer rates), power consumption (by way of oxidation/reduction potentials), and other attributes of molecular capacitors, as guided by the demands of different applications and systems, although there are certainly tradeoffs among these attributes.

We approach this new technology from a computer architecture perspective and show that the benefits of ZettaRAM extend beyond reducing fabrication complexity and costs.

Analysis of the circuit-level behavior of the molecular capacitor, reveals an unusual property: the amount of charge (Q) deposited on the molecular capacitor is decoupled from the write voltage (V), i.e., there is a discrete threshold voltage above/below which the molecular capacitor is fully charged/discharged. In contrast, the DRAM capacitor is constrained by conventional charge-voltage coupling, Q = CV. Decoupling control and amount of charge is key to viable voltage scaling in the future and opens up two unprecedented opportunities for power scaling in a DRAM-like architecture:

- *Exploiting Molecular Engineering for Long-Term Power-Scalability*: Read/write operations can be performed at much lower voltages, while still maintaining the minimum amount of charge needed for reliable sensing. The write voltage can be lowered by precisely tuning molecular attributes (threshold voltage, electron transfer rates and surface concentration). This flexibility provides an inexpensive path for scaling voltage, hence power, from one memory generation to the next. In DRAM, where Q = CV, each new DRAM generation requires major redesigns to keep Q nearly fixed as both cell area and operating voltage of successive generations of DRAM are reduced [31][45][60]. This power-scaling aspect gives ZettaRAM a strategic advantage over conventional DRAM.
- *Intelligent Management of ZettaRAM*: ZettaRAM presents a novel speed/energy tradeoff. While the fixed charge is independent of voltage, the speed of charging/discharging the molecules depends on the difference between the applied voltage and the threshold voltage (faster with larger differential). Thus, the applied voltage is padded with respect to the threshold voltage to achieve the same speed as DRAM. Thus, there is some room to lower voltage further, if performance can be dealt

with by some other means. A key architectural insight is leveraged to manage this tradeoff and lower the voltage even further: most of the bitline activity is caused by L2 cache writeback requests. So, a hybrid policy is proposed, wherein slow writes (lower voltage) are applied to non-critical writebacks and fast writes (higher voltage) to critical fetches. The proposed dynamic voltage modulation further extends ZettaRAM's power scaling advantage with little system slowdown.

Section 1.2.1 further explains the charge-voltage decoupling characteristic. Sections 1.2.2 and 1.2.3 describe the two power scaling opportunities afforded by ZettaRAM.

1.2.1 ZettaRAM: Charge-Voltage Decoupling

This section highlights the key distinction between the conventional capacitor in DRAM and the molecular capacitor in ZettaRAM.

The graph in Figure 1-3 shows charge density (charge per unit area) as a function of write voltage, for the DRAM capacitor. We use charge density, to ensure that comparisons are with respect to the same cell area. The amount of charge deposited on a conventional capacitor depends linearly on the applied write voltage ($Q = C \cdot V$, where Q is charge, C is cell capacitance, V is write voltage). Thus, there is a minimum write voltage, below which not enough charge is deposited on the conventional capacitor for the sense amplifier to reliably detect a "1" during a later read operation. The minimum charge density for reliable sensing is shown with the dashed horizontal line superimposed on the graph in Figure 1-3. The minimum write voltage corresponds to where this line intersects the conventional capacitor curve.

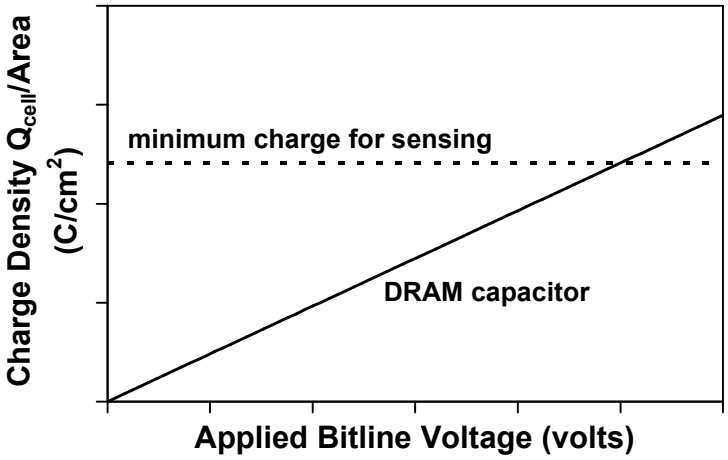


Figure 1-3. Charge density vs. voltage for a conventional capacitor in DRAM.

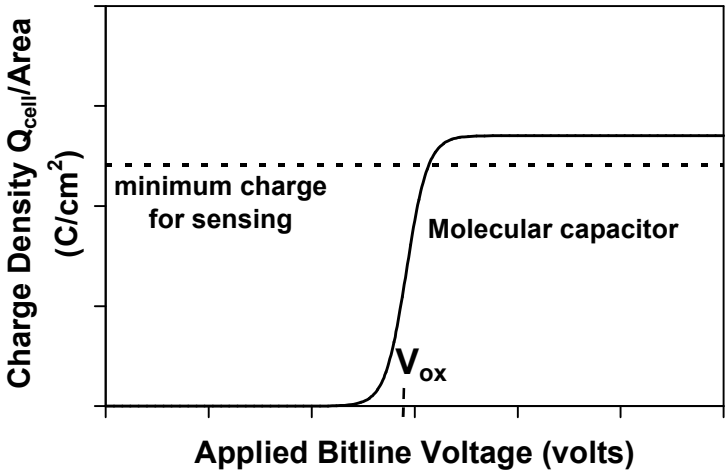


Figure 1-4. Charge density vs. voltage for a molecular capacitor in ZettaRAM.

The graph in Figure 1-4 shows charge density as a function of write voltage, for the molecular capacitor used in ZettaRAM. The relationship is nonlinear and centers around a discrete threshold voltage called V_{ox}, the oxidation potential of the molecule. When the write voltage is above V_{ox}, the molecules are fully charged. Conversely, when the write voltage is below V_{ox}, the molecules are fully discharged. In other words, whereas voltage still controls charging and discharging, the *amount* of charge is independent of voltage. In contrast, in conventional DRAM, voltage has dual roles of controlling charging/discharging and

influencing the amount of charge. Separating control and amount of charge is key to viable voltage scaling in the future.

1.2.2 Exploiting Molecular Engineering for Long-Term Power Scalability

We now explore how charge-voltage decoupling enables long-term power scalability of ZettaRAM, in terms of being able to continue the trend of lowering operating voltage from one memory generation to the next.

1.2.2.1 Limitations of DRAM Voltage Scaling

Figure 1-5 illustrates DRAM voltage scaling trends from one generation to the next, published by ITRS [5]. The DRAM voltage remained at 12 V until 1990. Then, the voltage scaled down to 5 V and remained at 5 V until the mid-90s, i.e., until the 0.7 micron generation. Since then, voltage has been decreasing 30% per generation. However, ITRS projects a voltage decrease of only 10% per generation in the future and there is no known solution to lower voltages below 1 V.

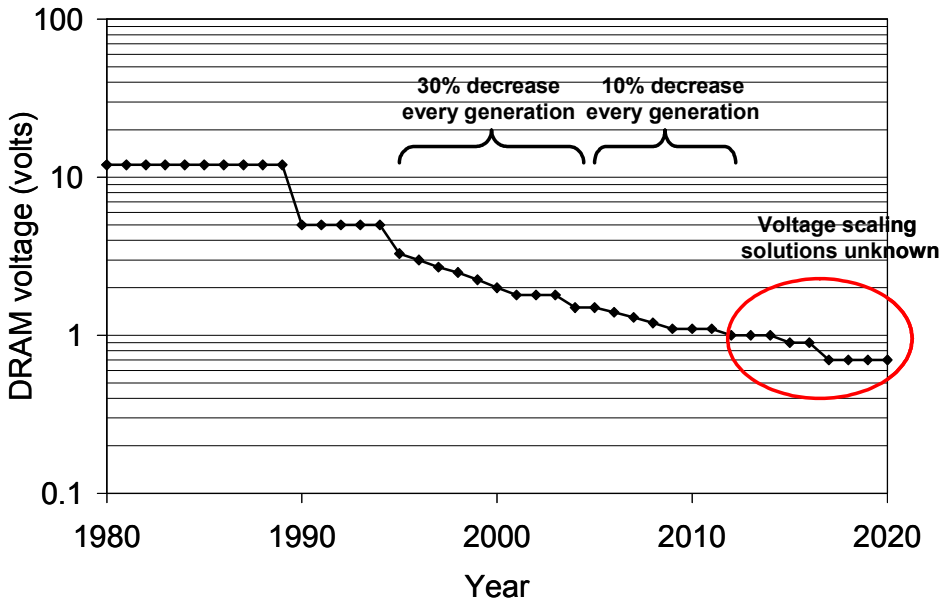


Figure 1-5. DRAM voltage scaling trends (ITRS).

Therefore, while DRAM voltage has certainly scaled with technology in the past, we surmise that we are approaching a fundamental limit to voltage scaling in DRAM. As described in the previous section and Figure 1-3, DRAM charge is constrained by the equation Q = CV. This constraint makes it extremely difficult to continue scaling DRAM voltage in future generations.

The graph in Figure 1-6 shows charge density as a function of write voltage for future DRAM generations, illustrating DRAM voltage scaling trends. From Figure 1-6, we observe that the minimum charge required for reliable sensing remains constant from one generation to the next. The charge in the capacitor (Q) must be large enough to generate a bitline voltage change that can be reliably sensed, including compensating for various noise sources (radiation, leakage current, and electrical imbalances between pairs of bitlines). While noise induced by radiation decreases with each generation, leakage current and electrical imbalances remain nearly the same from one generation to the next. Therefore, the minimum charge required has not reduced much with each new generation and this trend is expected to continue in the future [31][45]. Now, if we want to lower V while keeping Q constant, C must be increased, corresponding to steeper slopes in Figure 1-6. Yet, DRAM scaling trends show that the DRAM cell capacitance will remain steady at around 30 fF to 40 fF and will not scale with technology any further [45][60].

The voltage scaling problem is compounded if we want to simultaneously lower both voltage and area with each new generation. Reducing the two-dimensional area of the capacitor (higher density) is already difficult: it requires building a correspondingly taller capacitor to keep C the same, hence, Q the same. If we also want to lower the operating voltage, C must actually be increased to compensate.

In summary, charge-voltage coupling imposes fundamental limits on DRAM voltage scaling.

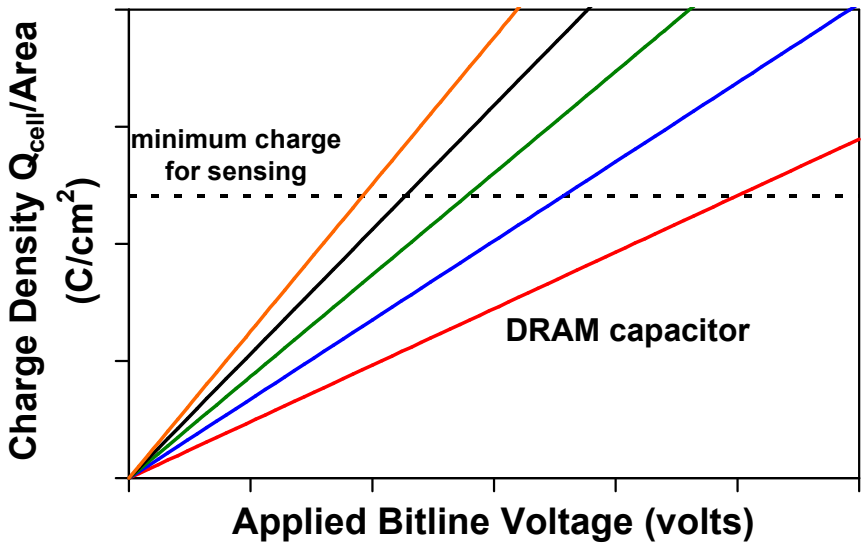


Figure 1-6. DRAM Voltage Scaling Trends.

1.2.2.2 ZettaRAM: Inexpensive Path to Power Scaling

In contrast with conventional DRAM, scaling voltage of ZettaRAM is viable, thanks to charge-voltage decoupling of the molecular capacitor. The graph in Figure 1-7 shows charge density as a function of write voltage for different molecule types, illustrating ZettaRAM voltage scaling trends from one generation to the next. As explained in the previous sub-section, the minimum charge required for reliable sensing does not decrease. Fortunately, the fixed charge of the molecular capacitor is independent of voltage. Therefore, voltages can be lowered in ZettaRAM from one generation to the next by synthesizing new molecules with lower threshold voltages. Different molecules with a range of oxidation potentials have been demonstrated in the literature and are evaluated in this dissertation.

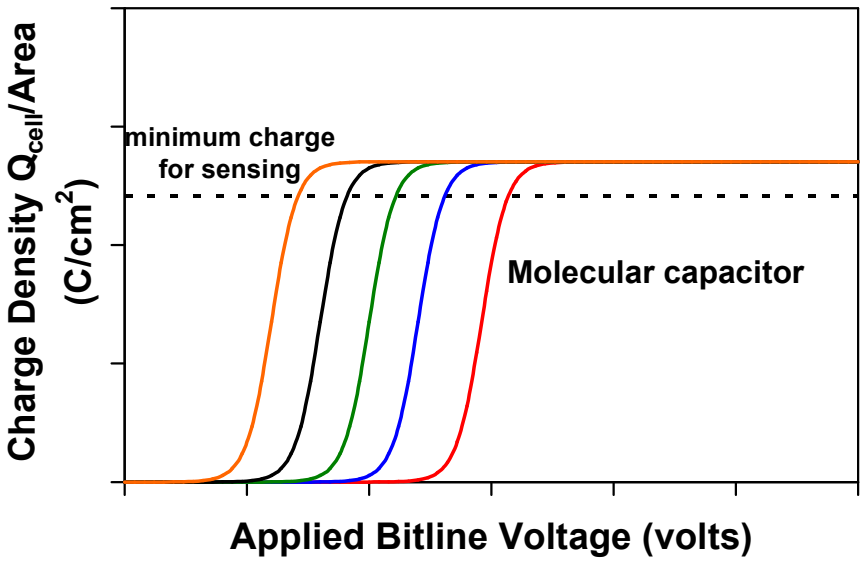


Figure 1-7. ZettaRAM Voltage Scaling Trends.

Key properties of the molecules can be precisely tuned through the choice of molecular "groups" and "linkers", such as the oxidation potential (V_{ox}), electron transfer rate (k^0), and surface concentration (charge density). As we show in this dissertation, all three parameters influence the operating voltage either directly or indirectly. Flexibility in engineering of molecular attributes provides an inexpensive path for scaling voltage, hence power, from one memory generation to the next.

We demonstrate the long-term power scalability of ZettaRAM by exploring dozens of molecules, synthesized and characterized by ZettaCore. Among these, we identify molecules that yield ZettaRAMs comparable to or better than DRAM in all respects – density (as determined by *cell area*), performance (as determined by *write latency*), and power (as determined by *voltage*). Our studies reveal that the best molecule yields a ZettaRAM voltage of 0.65V, whereas the charge-voltage dependence of the conventional capacitor limits DRAM voltage to 1.25V.

1.2.3 Intelligent Management of ZettaRAM

In the previous section, we showed that charge-voltage decoupling of the molecular capacitor enables ZettaRAM to overcome conventional DRAM voltage scaling limits.

However, while the fixed charge is independent of applied voltage, the speed of charging/discharging the molecules depends on the difference between the applied voltage and the oxidation potential. Charging/discharging the molecules becomes exponentially slower, the closer the applied voltage is to the threshold voltage (V_{ox}). This is illustrated in the graph in Figure 1-8, which shows the intrinsic latency of charging/discharging the molecules as a function of voltage. Superimposed on this graph is the DRAM write latency (dashed line). The overall speed of ZettaRAM is determined by either the speed of charging/discharging the molecules or the conventional peripheral circuitry used to access the molecular capacitor, whichever is slower. Accordingly, from Figure 1-8, we can see that ZettaRAM yields the same performance as DRAM if voltages that are used for charging and discharging the capacitor are sufficiently far away from the threshold voltage (V_{ox}), since the intrinsic speed of the molecules are not the bottleneck at these voltages. Our SPICE simulations, presented later in Section 3.3.2, also confirm this observation.

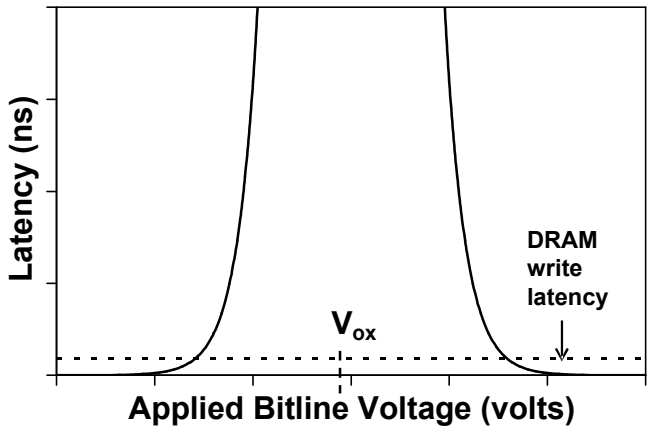


Figure 1-8. Intrinsic latency of charging and discharging molecules vs. voltage.

The operating voltage can be set closer to or farther away from V_{ox} to favor either energy or performance, respectively. To achieve the same performance as DRAM, the applied voltage is padded with respect to the oxidation potential. If competitive performance could be achieved by some other means, voltage could be lowered further. We propose architectural techniques to intelligently manage the performance/energy tradeoff. Specifically, we describe a novel hybrid write policy, which dynamically modulates the write voltage based on the criticality of memory requests to maximize energy savings potential of ZettaRAM with little system slowdown. The proposed dynamic voltage modulation further extends ZettaRAM's power scaling advantage over DRAM.

Requests to DRAM are usually serviced from a row buffer – an entire row (page) of the memory bank held in the row buffer. The row buffer typically contains the most recently accessed memory page. When a memory request (initiated by the L2 cache) misses in the row buffer, the current open page is closed (write operation) before opening a new page (read operation) to service the request. Bitline voltage swings are caused by both write and read operations. In ZettaRAM, the read operation can be performed only at a fixed voltage, as explained in detail in Section 3.2.1. However, the write operation can be performed at either a high voltage (fast, but high energy), favoring performance, or a low voltage (slow, but low energy), favoring energy savings.

Two types of memory requests are initiated by the L2 cache, fetch block and writeback block. Several factors converge nicely to direct focus on L2 writebacks: (i) they account for 80% of row buffer misses, thus most of the energy savings potential, and (ii) they do not directly stall the processor and thereby offer scheduling flexibility for tolerating extended molecule latency. On the other hand, L2 fetch requests typically stall the processor even with

out-of-order execution, because the instruction scheduling window is not large enough to accommodate the high memory round-trip latency. Accordingly, we propose a hybrid policy, in which slow writes (low energy) are applied to non-critical writebacks that miss in the row buffer, and fast writes (high energy) to critical fetches that miss in the row buffer. Applying slow writes to writebacks taps most of the energy savings potential, and applying fast writes to fetches ensures little performance degradation.

As an example, we characterized the additional energy savings yielded by the hybrid write policy for one of the first synthesized porphyrin molecules. This molecule has V_{ox} = 0.73V. We show that, if fast and slow writes are done at 1.2 V and 1.0 V, respectively, then the hybrid write policy yields 34% bitline energy savings (out of a possible 41% with uniformly slow writes) with only a 10% increase in execution time (as opposed to 81% with uniformly slow writes). Thus, the hybrid policy combines the performance of uniformly fast writes and the energy savings of uniformly slow writes. The residual 10% performance degradation still exists because, although deferred writebacks do not directly stall the processor, they may fill up the request queues in the memory controller, eventually stalling critical fetches.

One approach to avoid any queue-full stalls, is to increase the request queue size. The residual performance degradation is reduced from 10% to less than 1% when the request queues are increased from 4 to 64 entries. However, enlarging the queues increases system cost (each entry contains an entire cache block) and complexity (fetches that bypass queued writebacks must first search the queue for read-after-write hazards).

To avoid the cost and complexity of larger queues, as an alternative approach, we propose employing the eager writeback policy in the L2 cache [41] to evenly spread out

writeback requests, nearly eliminating queue-full stalls. In the eager writeback policy, a writeback is issued as soon as a dirty block becomes the LRU block in its set, instead of waiting for the block to be evicted. Issuing the writeback early from the L2 cache compensates for delaying it in the memory controller. The hybrid write policy in conjunction with eager writebacks yields 34% energy savings with less than 1% performance loss. This is achieved without enlarging the request queue size with respect to the baseline system (4 entries). Thus, the cost and complexity of the memory controller is kept the same as the baseline. We also noticed that for the SPEC2K benchmarks used in our simulations, eager writebacks have little effect in the baseline system, but are remarkably synergistic with ZettaRAM for eliminating performance degradation otherwise caused by delayed writebacks.

Hybrid fast/slow writes coupled with L2 eager writebacks taps most of the energy savings potential of ZettaRAM with little performance loss and the least complexity in the memory controller. Intelligently managed ZettaRAM further extends ZettaRAM's power scaling advantage over DRAM by 25-40%. Moreover, the hybrid write policy is able to extract energy savings from 6 molecules that otherwise consume more energy than DRAM.

1.3 Dissertation Contributions

This dissertation makes the following major contributions:

1. *RAPID yields scalable DRAM standby power by nearly eliminating refreshes.*

   - *RAPID-1 (static):* RAPID-1 is a static technique, where a few outlier pages are mapped out, achieving near-100% DRAM availability with a reasonably good refresh period that is independent of actual DRAM utilization.

- *RAPID-2 and RAPID-3 (dynamic):* RAPID-2 and RAPID-3 favor allocation of longer-retention pages over shorter-retention pages. A single refresh period is then set, based on the shortest-retention time among the occupied pages, rather than the shortest-retention page overall. RAPID-3 builds on RAPID-2, with the added ability to continuously reconsolidate pages into the longest-retention pages possible.

- *Software-only approach:* To the best of our knowledge, RAPID is the first software-only technique that exploits retention-time variations to reduce refresh power.

- *Compatibility with off-the-shelf refresh options:* Our proposed interrupt-driven refresh implementation takes into account the refresh options available in off-the-shelf DRAMs. As a result, RAPID can be readily used in conjunction with commodity DRAMs to virtually eliminate refresh power.

- *Testing Strategy:* We also propose an end-user test strategy to characterize retention times of DRAM pages, providing insights on run-time and power overheads of testing. Each DRAM page is tested with different test patterns in accordance with the standard data retention tests carried out in the industry. Testing is done within a rigorous statistical framework to ensure reliable test results with bounded error.

- *Implement RAPID on a real testbed:* We successfully implemented the first two RAPID techniques on an embedded system development board, that contains a real DRAM chip.

2. *ZettaRAM overcomes voltage scaling limits of DRAM.*

- *Understanding charge-voltage decoupling of molecular capacitor:* A contribution of this dissertation is uncovering the unusual property of ZettaRAM. The charge

deposited in the molecular capacitor is independent of write voltage. Charge-voltage decoupling opens up two unprecendented opportunities for power scaling in a DRAM-like architecture.

- *Novel SPICE model for molecular capacitor:* SPICE simulations and other analyses confirm charge-voltage decoupling and the speed/energy tradeoff of the molecular capacitor.

- *Exploiting molecular engineering for long-term power-scalability of ZettaRAM:* Fixed charge combined with flexible molecule engineering provides an inexpensive path for scaling voltage, hence power, from one memory generation to the next, whereas charge-voltage coupling limits conventional DRAM voltage scaling. Through simulations and other analyses, we identify molecules that have favorable attributes for voltage scaling.

- *Intelligent management of ZettaRAM:* We architecturally manage the speed/energy tradeoff to further extend ZettaRAM's power scaling advantage over conventional DRAM. The write voltage is dynamically modulated to maximize the energy savings potential of ZettaRAM with little system slowdown.

1.4 Dissertation Organization

The remainder of this dissertation is organized as follows. Chapter 2 illustrates how RAPID techniques can nearly eliminate refresh power yielding quasi-non-volatile DRAM. Chapter 3 describes in detail the two power-scaling facets provided by ZettaRAM by virtue

of charge-voltage decoupling. Related work is discussed in Chapter 4. Chapter 5 summarizes the contributions of this dissertation and Chapter 6 describes directions for future research.

Chapter 2

Retention-Aware Placement in DRAM (RAPID)

This chapter presents RAPID, software techniques for achieving quasi-non-volatile DRAM. RAPID exploits retention-time variations among different memory pages in a single DRAM chip. The key idea is to favor allocation of longer-retention pages over shorter-retention pages. A single refresh period is then set, that depends on the shortest-retention page among populated pages, instead of the shortest-retention page overall. This refresh period can be up to two orders of magnitude higher than the default refresh period of commodity DRAMs, significantly reducing the frequency of refreshes. As a result, RAPID reduces DRAM standby power to near-zero levels, yielding favorable standby power scaling despite quadrupling of DRAM capacity from one generation to the next.

We describe three variations of the RAPID technique. RAPID-1 is a static technique, where 1% of the outlier pages having low retention times are discarded and mapped out. RAPID-2 is a dynamic technique, which allocates longer-retention pages to application programs before allocating shorter-retention pages. RAPID-3 builds on RAPID-2, by continuously reconsolidating data to the longest-retention pages possible.

This chapter is organized as follows. Section 2.1 describes our test strategy and section 2.2 presents measured page retention times for a real DRAM chip. Section 2.3 presents the RAPID techniques and illustrates how RAPID can be coupled with off-the-shelf DRAMs. Our evaluation methodology and results are described in Section 2.4 and Section 2.5,

respectively. Section 2.6 provides a brief description of our implementation of RAPID-1 and RAPID-2 on a prototype testbed.

2.1 Retention Time Testing Strategies

This section discusses strategies for testing page retention times and presents test results for a real DRAM chip. We use the DRAM of an embedded systems development board (IP3023) donated by Ubicom [61]. Ubicom develops embedded microprocessors for wireless applications, including some of the first multithreaded embedded microprocessors, such as the IP3023. The IP3023 development board configuration is given in Table 2-1.

Table 2-1. Ubicom IP3023 Development Board Configuration

Hardware threads	8
Pipeline	10-stage in-order scalar (shared among the 8 threads)
Branch prediction	Static
Instruction scratchpad memory (on-chip)	64 KB
Data scratchpad memory (on-chip)	256 KB
SDRAM (IS42S81600A manufactured by ISSI) On-chip SDRAM controller	16 MB
FLASH memory	4 MB

The ISSI DRAM supports the two refresh options typically provided by off-the-shelf DRAMs, auto-refresh and self-refresh. Both auto-refresh and self-refresh can be disabled, a useful feature for testing page retention times.

The experimental setup for testing page retention times is pictured in Figure 2-1. The four parts are labeled 1-4 in the picture: (1) the Ubicom board with DRAM under test, (2) a PC workstation with Ubicom software development environment for compiling and downloading programs to the Flash memory via an ethernet connection to the board, (3) a

multimeter with attached thermistor for temperature readout, and (4) a heat gun for varying ambient temperature.

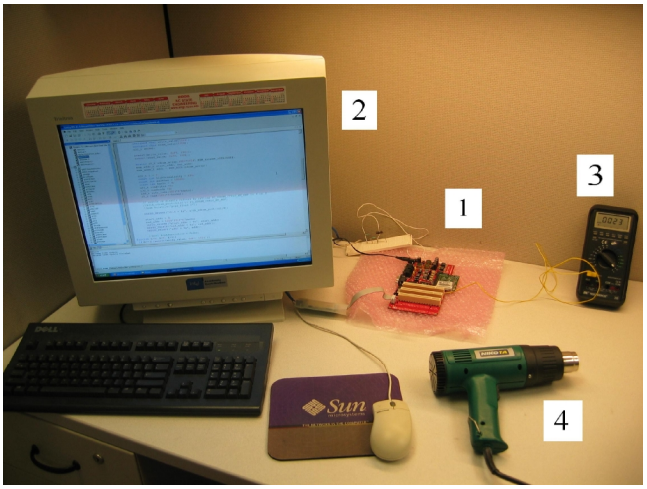


Figure 2-1. Experimental setup for testing page retention times.

2.1.1 Testing Algorithm

To characterize retention times of DRAM pages, both auto-refresh and self-refresh are disabled in the memory controller and DRAM, respectively, so that pages are not refreshed. A single page is tested by writing different test patterns to the page, for example, all 1's or all 0's, and reading the page back after a specified amount of time (wait time), to check if the pattern remains intact. The wait time is steadily increased until the checking of the pattern fails (one or more bit errors). The immediately previous wait time is recorded as the retention time of the page. In other words, if the pattern is retained for a wait time of T, the test is repeated for a wait time of $T + \Delta t$. If this next attempt fails (one or more bit errors), the retention time for the page is recorded as T (since it worked for a wait time of T but not $T + \Delta t$).

We measured retention times of some pages approaching a minute at room temperature. Assuming an initial wait time T of 1 second and proceeding in increments Δt of 1 second, it

takes 22 minutes (1 + 2 + … + 51 seconds) for just one trial of a page that retains information for 50 seconds. Testing pages one at a time would take far too long. Fortunately, all pages can be tested in parallel. The test program writes the entire DRAM with the test pattern, waits for the next wait interval $T + \Delta t$, checks the contents of every page, and records a retention time of T for any page that retained for T but not for $T + \Delta t$. One trial is completed for all pages collectively before attempting other trials.

The time to write and check the entire ISSI DRAM is about 74 ms and 250 ms, respectively. A recorded retention time reflects only the wait time after writing and before checking, therefore, the overheads can only mean that the actual retention time for a page is slightly longer than recorded, a safe margin of error as it will ensure that the page is refreshed slightly more often than required.

It is interesting to note that the test time for a trial is insensitive to the number of pages tested in parallel. We stop testing pages whose retention times become known midway through the trial, only as a matter of power efficiency for on-line testing (more on this in the next section). Rather, test time is dominated by the long wait times.

Specifically, the test time for one trial depends directly on the initial wait time, final wait time, and the wait time increment Δt. The initial and final wait times depend on the pages with the shortest and longest retention times, respectively. Therefore, the initial and final wait times are fixed for a given DRAM chip. The wait time increment Δt is the most flexible parameter of the three. Increasing this granularity Δt decreases the testing time, but also decreases the precision of retention time measurements, as a result of which, we may not be able to tap the full potential of refresh-power savings.

In order to avoid increasing the testing time dramatically, while at the same time maintaining the accuracy of retention time measurements, we used a non-uniform granularity in our testing strategy. We used a granularity of 1 second for tests above 3 seconds, 100 ms for tests between 1 and 3 seconds, and 10 ms for tests below 1 second. Moreover, we first test above 3 seconds, so that only pages that do not retain above 3 seconds are considered in the finer tests (the overhead of writing and checking the entire DRAM would skew the finer tests). Likewise, we test between 1 and 3 seconds before testing below 1 second so that only residual outlier pages are tested at the finest granularity. This testing strategy allows for precise measurements of the shortest-retention pages while keeping the test time reasonable. However, these parameters can be adjusted in the interest of test time or precision, as desired. At room temperature, we found that less than 1% of the DRAM pages had retention times greater than 50s. Therefore, we set the final wait time to 50s to avoid unnecessarily increasing the testing times. The overall testing time for measuring retention times at room temperature using this strategy was 22.3 minutes for one trial.

Table 2-2 summarizes the test time and the number of page writes/checks for one complete trial of the DRAM. Retention times from the first trial can be exploited to reduce the number of page writes/checks in subsequent trials. Subsequent trials can begin at the median of the distribution and spiral outward from the median, eliminating the greatest number of pages earliest in the trial. Thus, the number of page writes/checks in Table 2-2 is a worst-case number for the first trial. The test time is not reduced for subsequent trials because we still cycle through all the same wait times.

Table 2-2. One complete trial of the DRAM.

Test time	22.3 minutes
# page writes/reads (worst-case – first trial)	274,000

2.1.2 Off-line vs. On-line Testing

Page retention times can be tested off-line by the DRAM manufacturer, off-line by the system designer, or on-line during actual use of the system.

Currently, the DRAM manufacturer only tests chips at the worst-case refresh period (e.g., 64 ms) to confirm correct operation of the chip. This basic test is much faster than our generalized test algorithm due to its single short wait time. Applying our test algorithm would increase DRAM tester time, causing an unwanted increase in price. Furthermore, conveying chip-specific page retention times to DRAM customers poses unusual logistics problems – a small Flash table coupled with the DRAM in a multi-chip package or a web-accessible database are improbable.

Alternatively, system designers (e.g., cell phone designers) or advanced end users (e.g., sensor network researchers) could run the tests off-line for their particular applications. Since extreme energy efficiency is needed for these applications, this form of off-line testing is potentially justifiable from a business standpoint.

In the case of on-line testing, trials are performed gradually, opportunistically, and overall as non-intrusively as possible, e.g., while recharging the batteries of a mobile device or during extensive idle periods. (Or, for mobile devices, a Windows-style installation wizard could give users the option of testing the device upon first use or at some later time.) Multiple trials are collected over days, weeks, or even months. Testing time and power overheads recede over time as the DRAM is mapped. RAPID is only employed when the trials are complete. Correct operation is assured because the system safely reverts to the default refresh policy when trials are incomplete.

2.1.3 Testing and Temperature

Conventional DRAMs use a single worst-case refresh period (typically 64 ms), safe for the highest rated temperature and all lower temperatures. CellularRAM™ temperature-compensated refresh (TCR) provides four refresh settings, corresponding to four temperature ranges. A given refresh setting is safe for the highest temperature in its range and all lower temperatures. Exploiting TCR requires an external temperature sensor or *a priori* knowledge of the peak operating temperature, to guide selection of the lowest safe refresh setting.

Temperature can be accommodated in RAPID similarly, using either a single or multiple temperature ranges. The only difference is the use of page-dependent information instead of page-independent information.

However, the testing method – off-line vs. on-line – affects temperature options. A key advantage of off-line testing is that temperature can be explicitly controlled. Thus, testing can be done for the highest rated temperature and possibly several lower temperatures. If a temperature sensor is available, we can select page retention times corresponding to the current temperature. If a temperature sensor is not available, we can at least select page retention times corresponding to the highest rated temperature.

On-line testing is less flexible in two ways. First, a temperature sensor is mandatory, to tag each trial with the maximum temperature for which the trial is safe. The sensor is likewise needed to select safe page retention times corresponding to the current temperature. Second, temperature cannot be controlled during testing. It is possible for the current temperature to exceed the maximum recorded temperature among trials, in which case a RAPID refresh period cannot be determined and we must downgrade to either of the two

conventional refresh policies (worst-case refresh period or lowest safe refresh setting of TCR).

2.1.4 DRAM Test Patterns

Multiple test patterns are used in the DRAM industry to identify data retention fails with high coverage. Diverse test patterns are needed to identify potential cross-cell retention-time degradation or other interactions. The most common test patterns used to test data retention times are the checkerboard and inverse checkerboard (alternating rows of 1010… and 0101…) [20][38]. Other common test patterns are all 1s, all 0s, alternating 1s and 0s, and alternating 0s and 1s (i.e., 1010… and 0101…) [65][66][67][69].

In our retention time measurements, presented in Section 2.2, we use the six industry-standard test patterns above to test the ISSI DRAM chip. Our measurements of the ISSI DRAM chip show that five of the test patterns yield the same retention time, for 75% of the pages. We observe different retention times for these five test patterns, for only 25% of the pages, with a maximum variation in retention time of 40% among the five test patterns. For example, the retention time of a particular DRAM page (page #12) ranges between 6 seconds and 10 seconds among the five test patterns. The all 0s test pattern yields a much higher retention time than the other five test patterns (more on this in Section 2.2).

2.1.5 Statistical Framework

Testing is done within a rigorous statistical framework to ensure that assigned retention times are safe. The retention time of each page is estimated as follows:

1. Twenty trials are performed for each test pattern.

2. For a given test pattern, we calculate an adjusted retention time, equal to the mean of the 20 trials minus a margin of error based on confidence intervals [47]. Using confidence intervals is more statistically rigorous, hence, safer, than simply using the minimum retention time among the twenty trials. Statistical inference is safer because it emulates performing infinite trials (with a certain confidence level). Taking the minimum of twenty trials is only as reliable as these trials and may not provide a safe lower bound.
3. We estimate the retention time of the page to be the minimum of the adjusted retention time among all test patterns.

The goal of our statistical framework is to estimate a safe retention time from only twenty trials. The twety trials constitute a simple random sample (SRS). The central limit theorem states that as the sample size increases, the distribution of a simple random sample approaches a normal distribution. This is important, because rigorous statistical analyses are available for normal distributions. In particular, the student-t distribution can be used to project a full distribution from only a sample distribution, as long as there are 15 or more samples. The projected distribution provides a means to safely estimate a lower bound on retention time of the page.

A measured parameter "*x*" is represented as follows (where $\bar{x}$ is the sample mean):

$$\bar{x} \pm \text{margin\_of\_error for Y\% confidence}$$

In other words, a random measurement has Y% chance of being between $\bar{x}$ + margin_of_error and $\bar{x}$ -margin_of_error.

The following student-t distribution equation is used to determine the margin of error.

$$\bar{x} \pm t^* \frac{s}{\sqrt{n}}$$

t^* is the critical value corresponding to the student-t distribution, n is the sample size, and s is the sample standard deviation or standard error. $t^* \frac{s}{\sqrt{n}}$ is the margin of error.

The sample size and confidence level determine t^*. As the sample size increases, t^* decreases for a given confidence level (since more samples reveal more about the actual distribution, thus s is a better estimate of actual standard deviation and requires less adjustment by t^*). A higher confidence level increases t^* for a given sample size. The graphs in Figure 2-2 show hypothetical normal distributions (i.e., infinite samples) and illustrate the relationship between confidence level and margin of error. The mean must be padded with a larger margin of error for higher-confidence estimations.

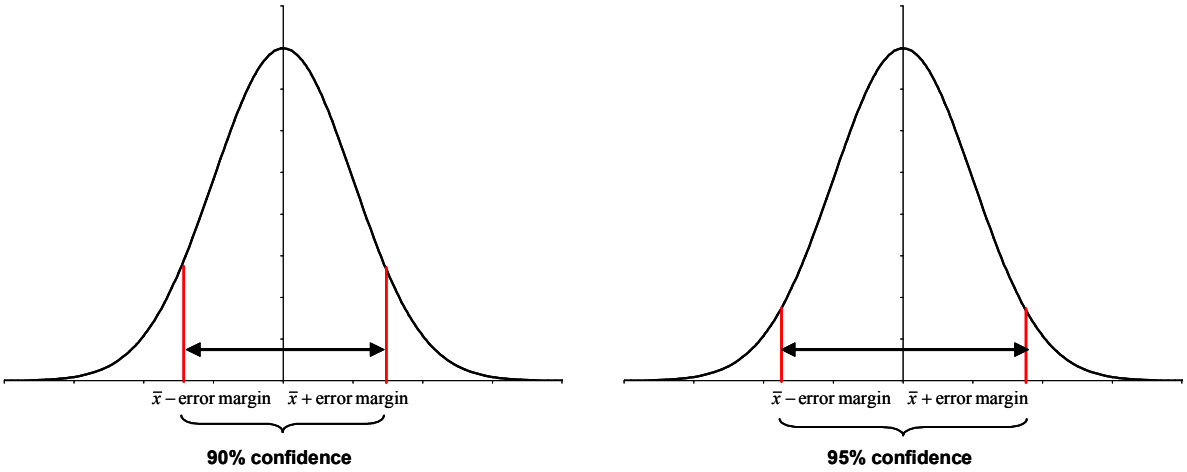


Figure 2-2. Estimation with (a) 90% confidence (b) 95% confidence.

Figure 2-3 shows the student-t distribution table used to look up the t* critical value. Each row corresponds to a number of degrees of freedom (sample size n has n-1 degrees of

freedom). Each column corresponds to a confidence level. The highlighted row is the one used for our retention time tests (20 trials, i.e., 19 degrees of freedom). Within this row, we select the value of t^* corresponding to the desired confidence level.

1	1.0000	1.3764	1.9626	3.0777	6.3137	12.7062	15.8945	31.8210	63.6559	127.3211	318.2888	636.5776
2	0.8165	1.0607	1.3862	1.8856	2.9200	4.3027	4.8487	6.9645	9.9250	14.0892	22.3285	31.5998
3	0.7649	0.9785	1.2498	1.6377	2.3534	3.1824	3.4819	4.5407	5.8408	7.4532	10.2143	12.9244
4	0.7407	0.9410	1.1896	1.5332	2.1318	2.7765	2.9985	3.7469	4.6041	5.5975	7.1729	8.6101
5	0.7267	0.9195	1.1558	1.4759	2.0150	2.5706	2.7565	3.3649	4.0321	4.7733	5.8935	6.8685
6	0.7176	0.9057	1.1342	1.4398	1.9432	2.4469	2.6122	3.1427	3.7074	4.3168	5.2075	5.9587
7	0.7111	0.8960	1.1192	1.4149	1.8946	2.3646	2.5168	2.9979	3.4995	4.0294	4.7853	5.4081
8	0.7064	0.8889	1.1081	1.3968	1.8595	2.3060	2.4490	2.8965	3.3554	3.8325	4.5008	5.0414
9	0.7027	0.8834	1.0997	1.3830	1.8331	2.2622	2.3984	2.8214	3.2498	3.6896	4.2969	4.7809
10	0.6998	0.8791	1.0931	1.3722	1.8125	2.2281	2.3593	2.7638	3.1693	3.5814	4.1437	4.5868
11	0.6974	0.8755	1.0877	1.3634	1.7959	2.2010	2.3281	2.7181	3.1058	3.4966	4.0248	4.4369
12	0.6955	0.8726	1.0832	1.3562	1.7823	2.1788	2.3027	2.6810	3.0545	3.4284	3.9296	4.3178
13	0.6938	0.8702	1.0795	1.3502	1.7709	2.1604	2.2816	2.6503	3.0123	3.3725	3.8520	4.2209
14	0.6924	0.8681	1.0763	1.3450	1.7613	2.1448	2.2638	2.6245	2.9768	3.3257	3.7874	4.1403
15	0.6912	0.8662	1.0735	1.3406	1.7531	2.1315	2.2485	2.6025	2.9467	3.2860	3.7329	4.0728
16	0.6901	0.8647	1.0711	1.3368	1.7459	2.1199	2.2354	2.5835	2.9208	3.2520	3.6861	4.0149
17	0.6892	0.8633	1.0690	1.3334	1.7396	2.1098	2.2238	2.5669	2.8982	3.2224	3.6458	3.9651
18	0.6884	0.8620	1.0672	1.3304	1.7341	2.1009	2.2137	2.5524	2.8784	3.1966	3.6105	3.9217
19	0.6876	0.8610	1.0655	1.3277	1.7291	2.0930	2.2047	2.5395	2.8609	3.1737	3.5793	3.8833
20	0.6870	0.8600	1.0640	1.3253	1.7247	2.0860	2.1967	2.5280	2.8453	3.1534	3.5518	3.8496
21	0.6864	0.8591	1.0627	1.3232	1.7207	2.0796	2.1894	2.5176	2.8314	3.1352	3.5271	3.8193
22	0.6858	0.8583	1.0614	1.3212	1.7171	2.0739	2.1829	2.5083	2.8188	3.1188	3.5050	3.7922
23	0.6853	0.8575	1.0603	1.3195	1.7139	2.0687	2.1770	2.4999	2.8073	3.1040	3.4850	3.7676
24	0.6848	0.8569	1.0593	1.3178	1.7109	2.0639	2.1715	2.4922	2.7970	3.0905	3.4668	3.7454
25	0.6844	0.8562	1.0584	1.3163	1.7081	2.0595	2.1666	2.4851	2.7874	3.0782	3.4502	3.7251
26	0.6840	0.8557	1.0575	1.3150	1.7056	2.0555	2.1620	2.4786	2.7787	3.0669	3.4350	3.7067
27	0.6837	0.8551	1.0567	1.3137	1.7033	2.0518	2.1578	2.4727	2.7707	3.0565	3.4210	3.6895
28	0.6834	0.8546	1.0560	1.3125	1.7011	2.0484	2.1539	2.4671	2.7633	3.0470	3.4082	3.6739
29	0.6830	0.8542	1.0553	1.3114	1.6991	2.0452	2.1503	2.4620	2.7564	3.0380	3.3963	3.6595
30	0.6828	0.8538	1.0547	1.3104	1.6973	2.0423	2.1470	2.4573	2.7500	3.0298	3.3852	3.6460
40	0.6807	0.8507	1.0500	1.3031	1.6839	2.0211	2.1229	2.4233	2.7045	2.9712	3.3069	3.5510
50	0.6794	0.8489	1.0473	1.2987	1.6759	2.0086	2.1087	2.4033	2.6778	2.9370	3.2614	3.4960
60	0.6786	0.8477	1.0455	1.2958	1.6706	2.0003	2.0994	2.3901	2.6603	2.9146	3.2317	3.4602
80	0.6776	0.8461	1.0432	1.2922	1.6641	1.9901	2.0878	2.3739	2.6387	2.8870	3.1952	3.4164
100	0.6770	0.8452	1.0418	1.2901	1.6602	1.9840	2.0809	2.3642	2.6259	2.8707	3.1738	3.3905
1000	0.6747	0.8420	1.0370	1.2824	1.6464	1.9623	2.0564	2.3301	2.5807	2.8133	3.0984	3.3002
infinity	0.6745	0.8416	1.0364	1.2816	1.6449	1.9600	2.0537	2.3263	2.5758	2.8070	3.0903	3.2906
	50%	**60%**	**70%**	**80%**	**90%**	**95%**	**96%**	**98%**	**99%**	**99.5%**	**99.8%**	**99.9%**
						Confidence level						

Figure 2-3. Student-t distribution table (t* critical value).

For a given page, shorter or longer adjusted retention times can be estimated for higher or lower confidence levels, respectively. This is illustrated in Table 2-3, which shows the margins of error for different confidence levels and the corresponding adjusted retention time for one particular page, page #1651. We observe that, as expected, the margin of error increases for higher-confidence estimations. We chose a 99% confidence level for our purposes.

Table 2-3. Adjusted Retention Time for Page #1651 (sample mean retention time = 39 s).

% confidence	% margin of error	Adjusted Retention Time (seconds)
95%	4.01%	37.44
96%	4.22%	37.35
98%	4.87%	37.10
99%	5.48%	36.86
99.5%	6.08%	36.63
99.8%	6.86%	36.33
99.9%	7.44%	36.09

Figure 2-4 shows the margins of error at 99% confidence level for different DRAM pages and different test patterns. We observe that, for all test patterns, 92-95% of the pages have a margin of error less than 2% and 98.9-99.8% of the pages have a margin of error less than 5%. Only around 0.2-1.1% of the pages have a margin of error between 5-15%. Therefore, there are very few pages with outlier samples among the 20 trials. Regardless, padding each page with its safe error margin, based on classic statistical inference, ensures reliability of our retention time measurements.

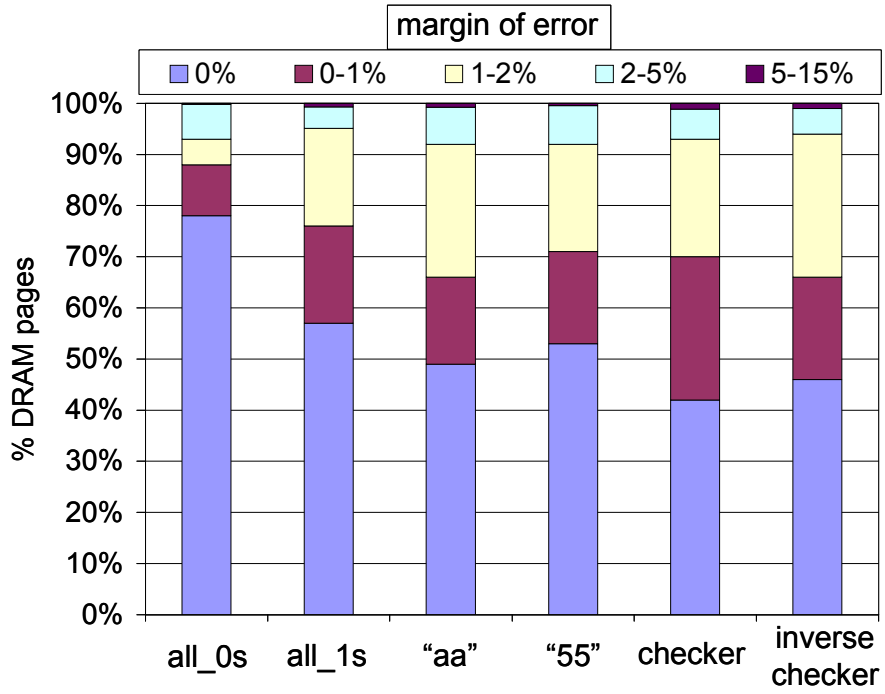


Figure 2-4. Margin of error at 99% confidence level for different DRAM pages and different test patterns.

2.2 Page retention times

In this section, we measure and graph the retention times of pages for the ISSI DRAM, using our testing algorithm. The ISSI DRAM under test implements a "close page" policy, i.e., a page is immediately written back to the memory array after a read or write operation. Therefore, the row buffer is not used as a cache, guaranteeing reads and writes always truly access the memory cells.

Section 2.2.1 presents the page retention times of the entire DRAM chip at room temperature (24°C). Measurements are done for all six test patterns. The retention time of a page is its mean retention time (20 trials) adjusted with a safe error margin, calculated based on statistical inference, as explained in the previous section.

Section 2.2.2 presents the pages retention times of the only the first DRAM bank at two higher operating temperatures, 45°C and 70°C. Page retention times are measured only using the test pattern of all ones. The retention time of a page is the minimum among the 20 trials. Note, the statistical analysis described in the previous section is not done for page retention time measurements at these two higher temperatures.

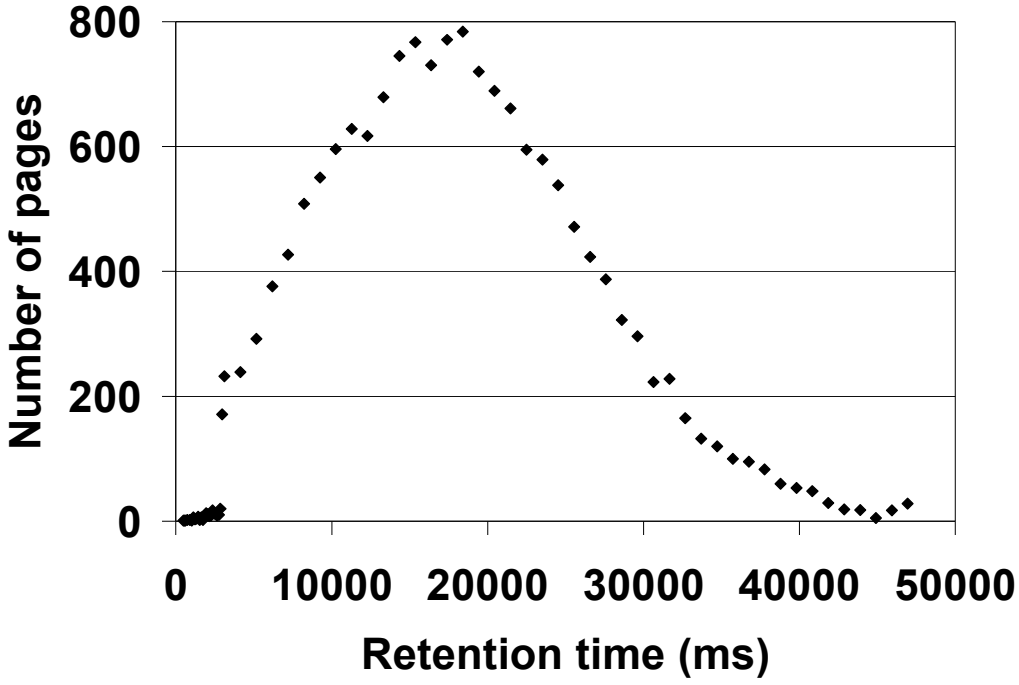
2.2.1 Page Retention Times at Room Temperature

Figure 2-5 shows the retention time distribution and cumulative retention time distribution of all pages in the DRAM for the test pattern of all ones, at room temperature (24°C). The distribution of page retention times, shown in Figure 2-5(a), is consistent with previous retention time characterizations [19]. The first observation is that there is a wide variation in retention times, ranging from 500 ms to slightly over 50 seconds. The second observation is that most pages have longer retention times than the default 64 ms refresh

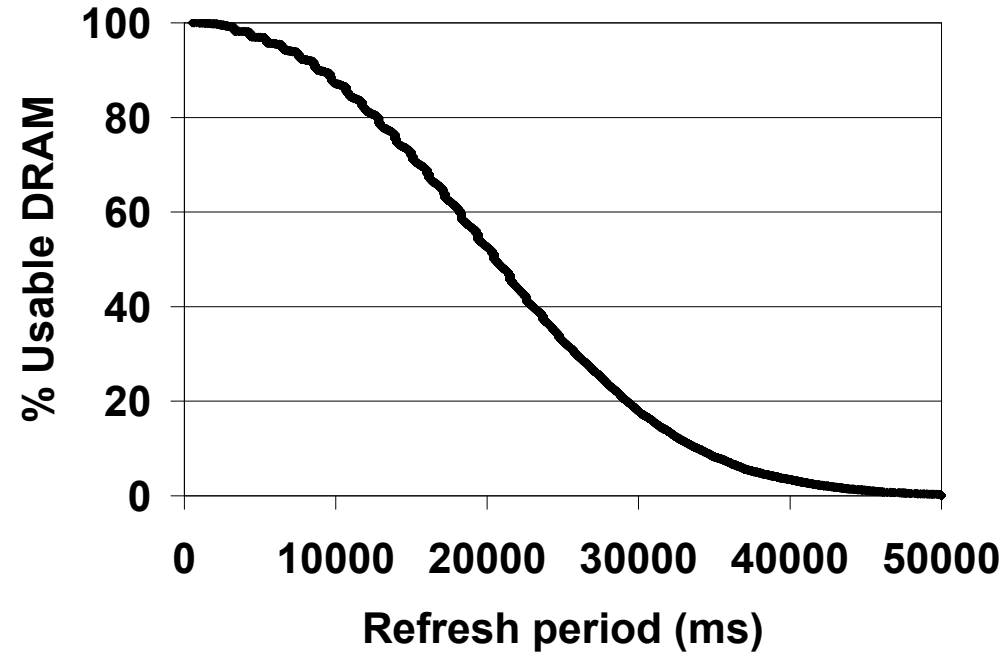
period or even the longest temperature-compensated refresh period (TCR), which is at most 500 ms based on the shortest-retention page in Figure 2-5(a). The corresponding cumulative distribution in Figure 2-5(b) shows the percent usable DRAM for a given target refresh period. For example, a refresh period of 1.4 s covers 99.9% of all pages and a refresh period of 3.1 s covers 99% of all pages. For a 99% DRAM utilization, RAPID-1 can consolidate data into the longest-retention 99% of the DRAM, thus, permitting a refresh period as long as 3.1 s without losing information.

Figure 2-6 shows that the all zeros test pattern yields very high retention times, with almost 90% of pages having retention times greater than 50 seconds, i.e., the leakage from 0 to 1 is much slower than leakage from 1 to 0. The other four test patterns (alternating ones and zeros, alternating zeros and ones, checkerboard, and inverse checkerboard) show distributions similar to all ones, as illustrated in Figure 2-7, Figure 2-8, Figure 2-9, and Figure 2-10. This is because 75% of the DRAM pages have the same mean retention times for all test patterns, except the all zeros test pattern.

We estimate the retention time of the page to be the minimum of the adjusted retention time among all test patterns. Figure 2-11 shows the retention time distribution and cumulative retention time distribution of this *estimated retention time* all pages in the DRAM. We use these *estimated page retention times* in our results section to evaluate the refresh power savings yielded by our RAPID techniques.

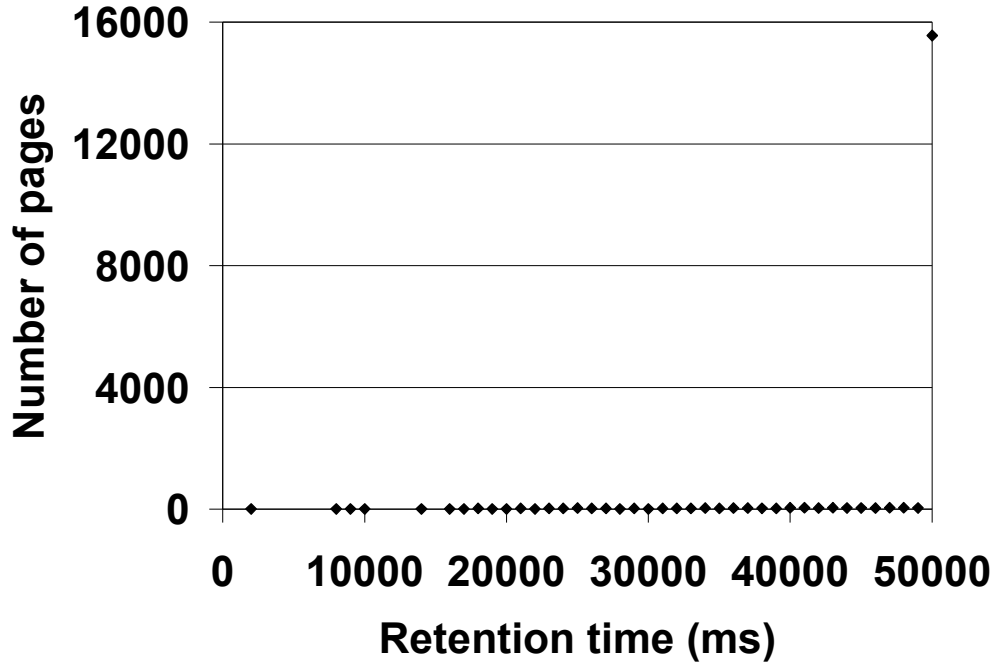


(a)

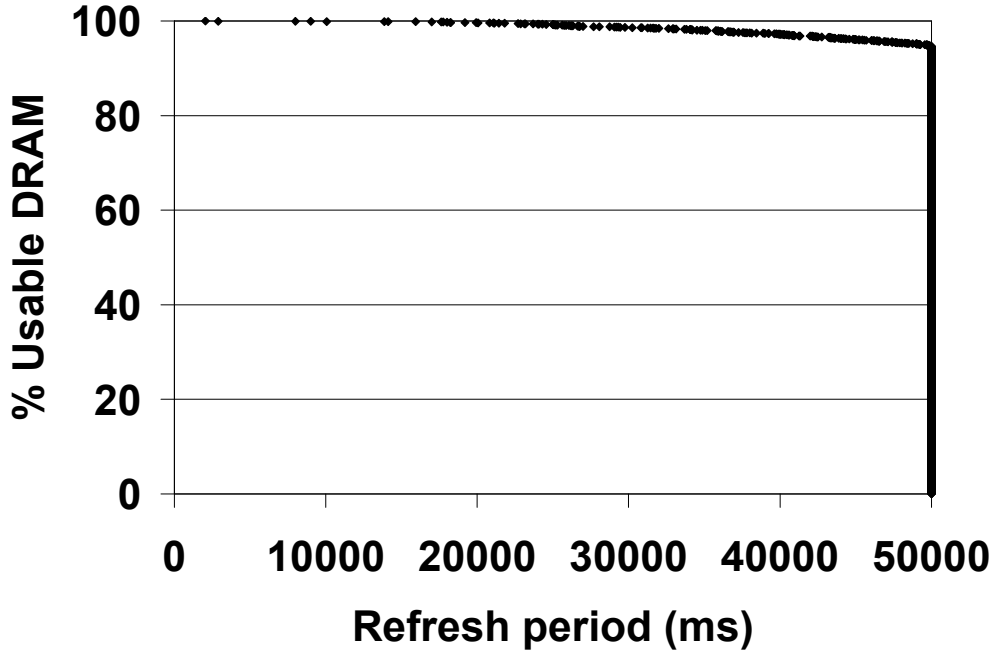


(b)

Figure 2-5. (a) Distribution and (b) cumulative distribution of page retention times, for entire DRAM at room temperature (24°C, test pattern all 1s).

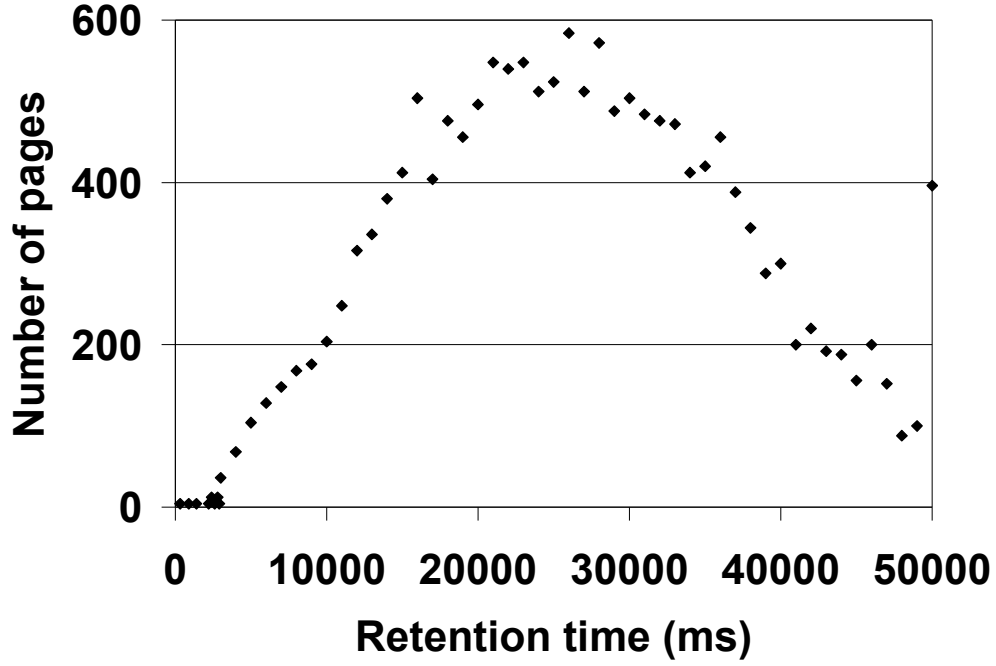


(a)

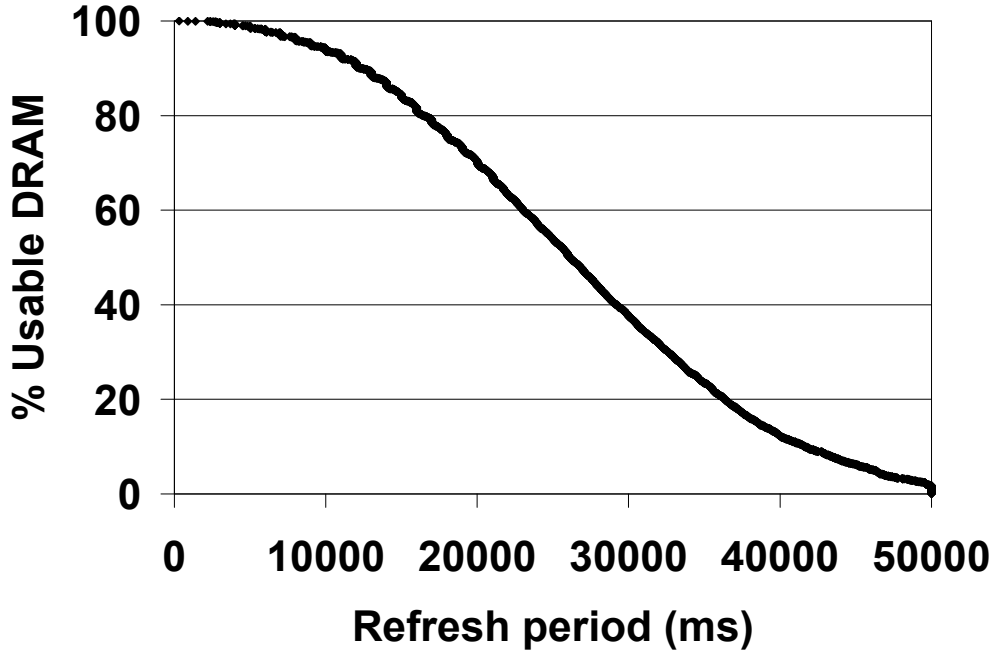


(b)

Figure 2-6. (a) Distribution and (b) cumulative distribution of page retention times, for entire DRAM at room temperature (24°C, test pattern all 0s).

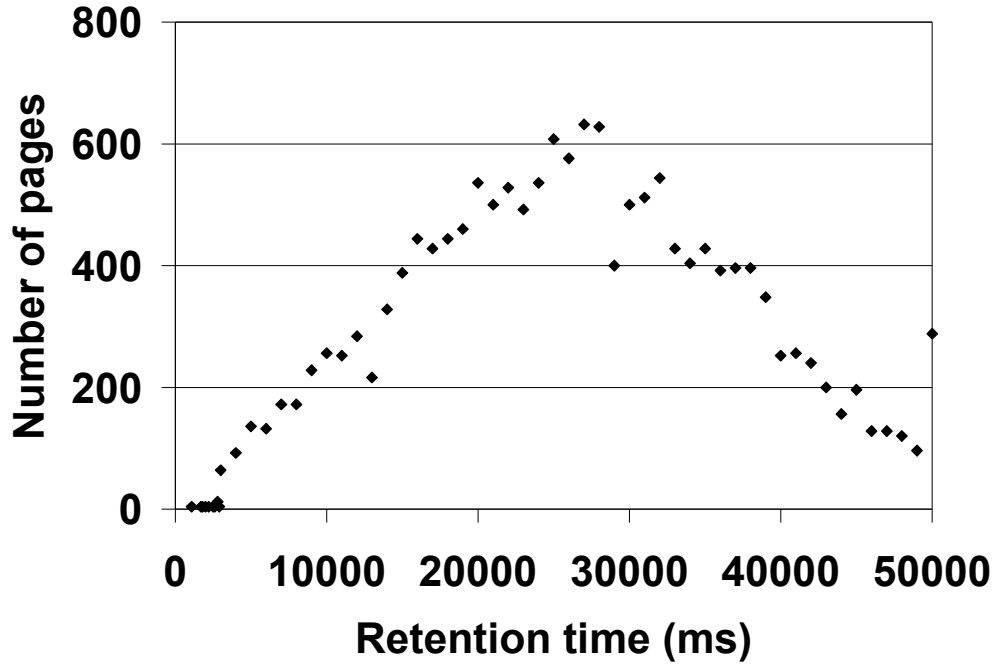


(a)

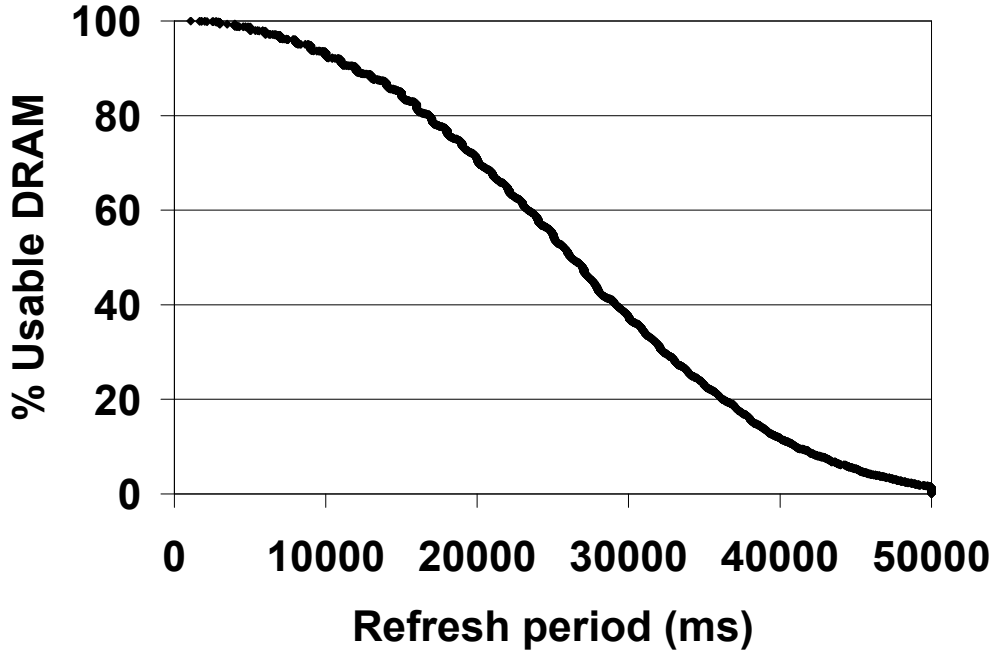


(b)

Figure 2-7. (a) Distribution and (b) cumulative distribution of page retention times, for entire DRAM at room temperature (24°C, test pattern alternating 1s and 0s – "aa"s).

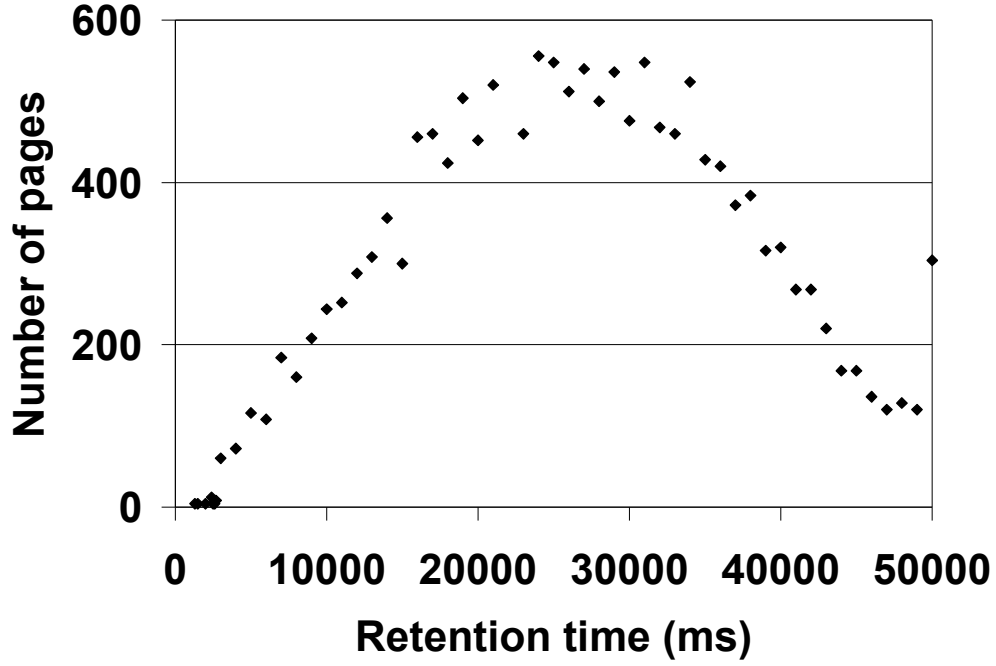


(a)

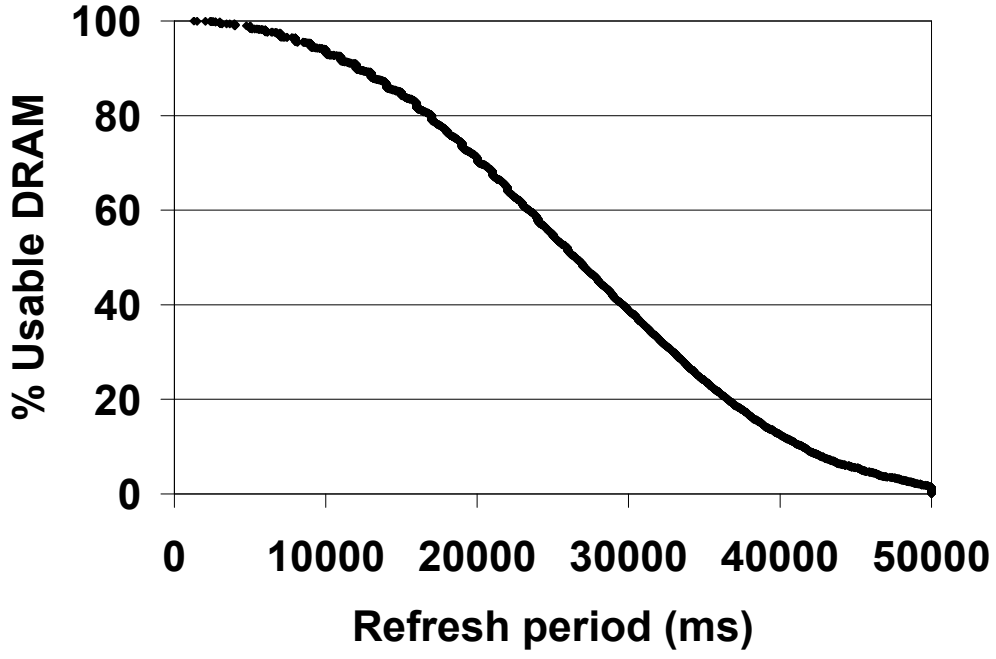


(b)

Figure 2-8. (a) Distribution and (b) cumulative distribution of page retention times, for entire DRAM at room temperature (24°C, test pattern alternating 0s and 1s – "55"s).

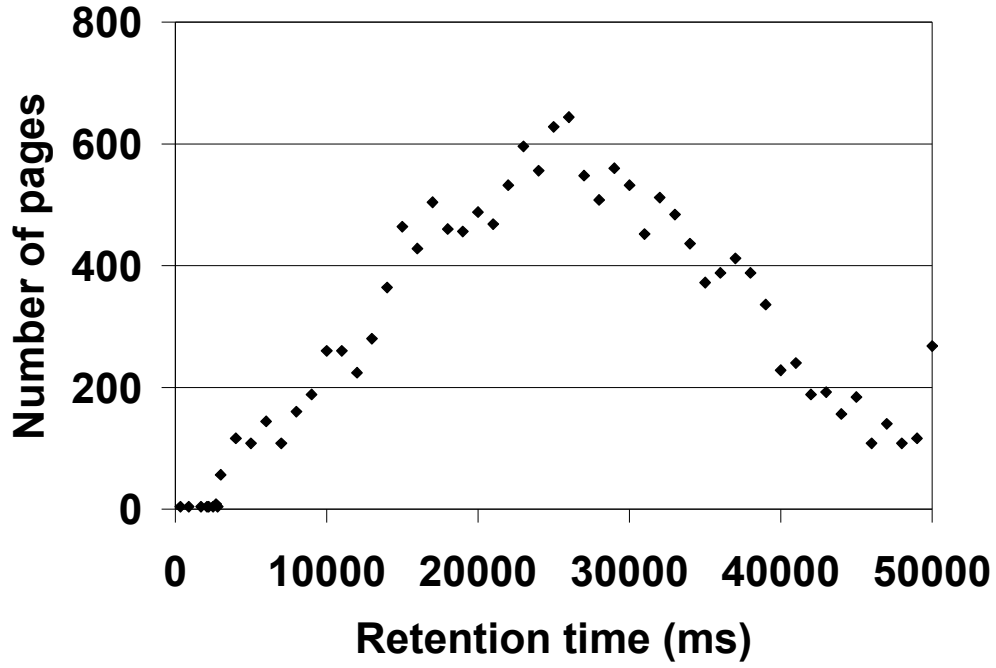


(a)

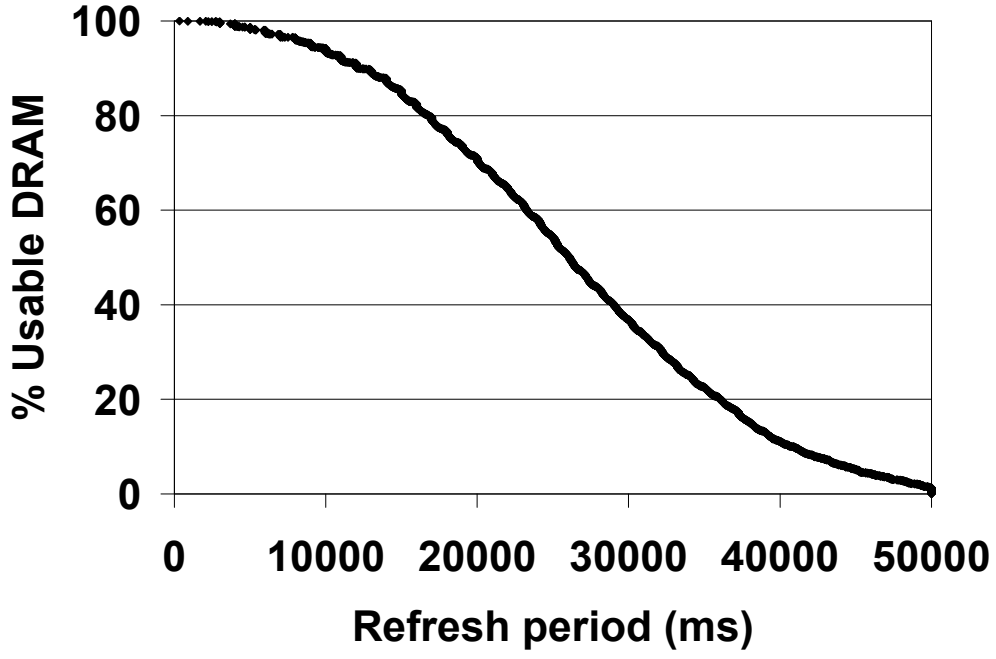


(b)

Figure 2-9. (a) Distribution and (b) cumulative distribution of page retention times, for entire DRAM at room temperature (24°C, test pattern checkboard).

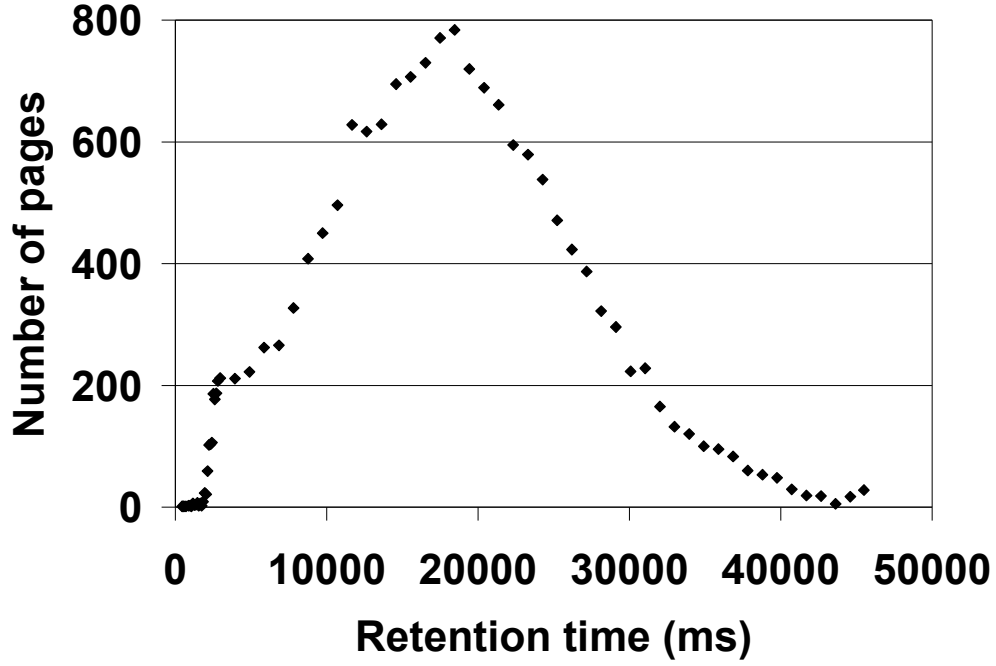


(a)

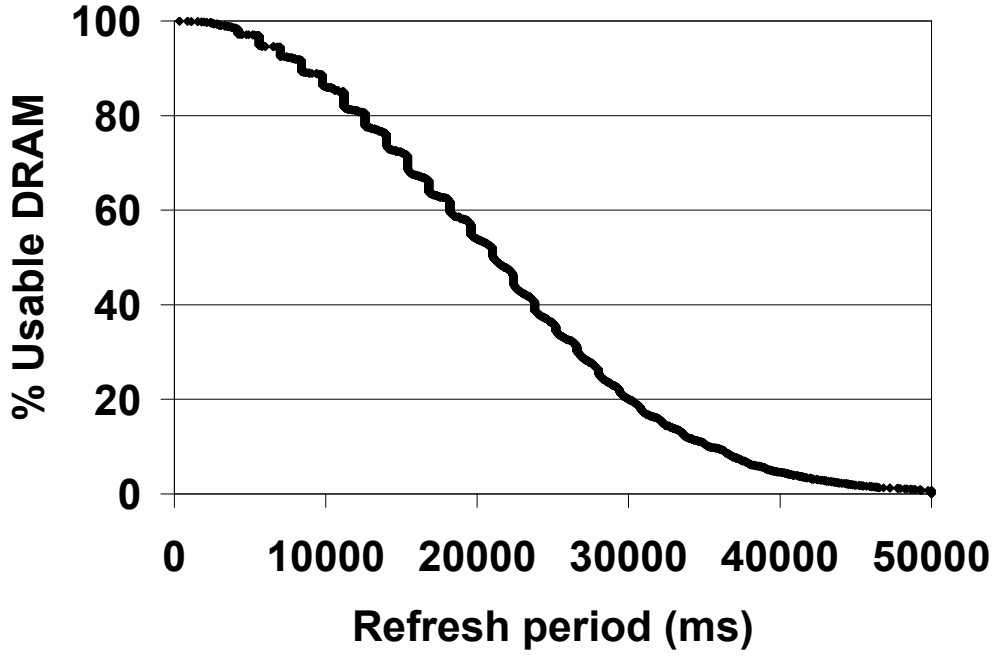


(b)

Figure 2-10. (a) Distribution and (b) cumulative distribution of page retention times, for entire DRAM at room temperature (24°C, test pattern inverse checkerboard).



(a)

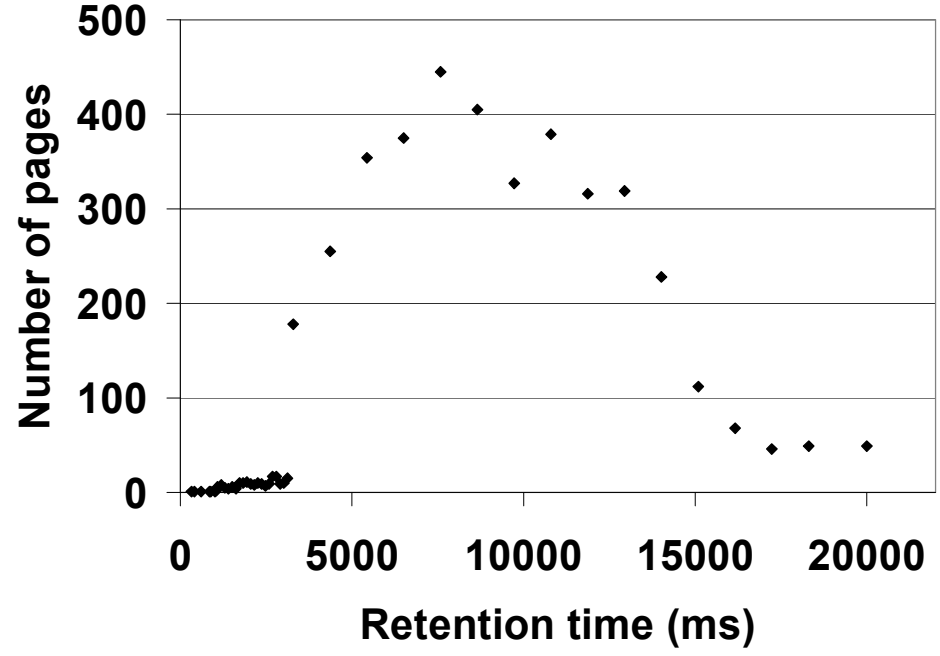


(b)

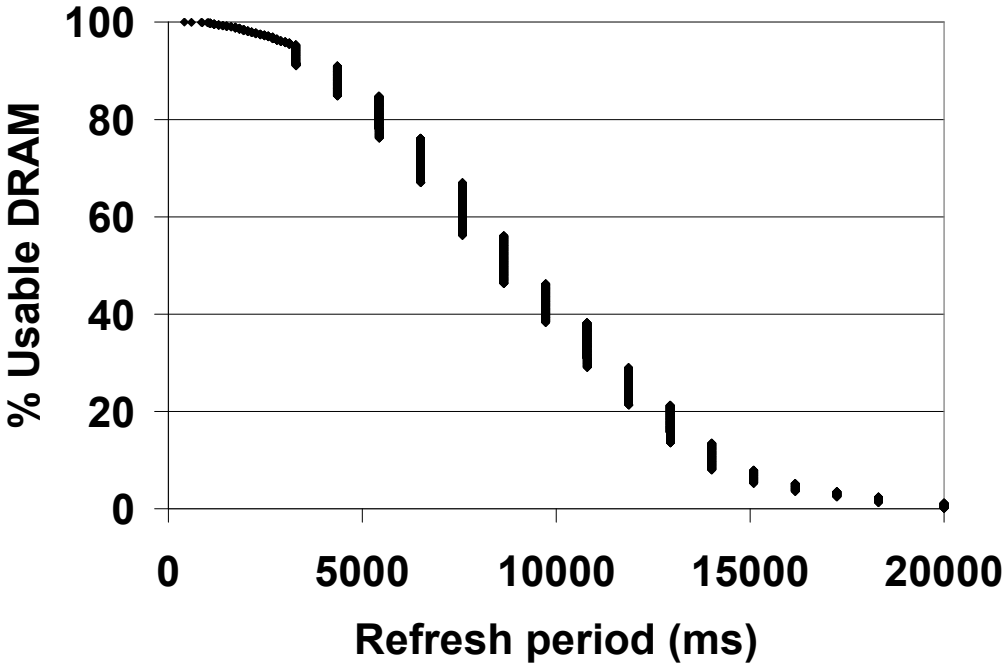
Figure 2-11. (a) Distribution and (b) cumulative distribution of page retention times, for entire DRAM at room temperature (24°C, minimum of all six test patterns).

2.2.2 Page Retention Times at Higher Temperatures

Temperature is known to have a significant effect on leakage and, therefore, retention time in DRAM. The leakage current increases with temperature. Therefore, retention times decrease with increasing temperature. We measured the page retention times of only the first DRAM bank (out of four banks) and only for the all ones test pattern, at two higher temperatures, 45°C and 70°C (the latter is the maximum operating temperature for the ISSI DRAM chip). The temperature was raised using the heat gun pictured in Figure 2-1 (labeled 4) and monitored during testing, using the multi-meter with temperature-sensing thermistor (labeled 3) placed in contact with the DRAM package. The temperature was kept within two degrees of the desired temperature. The results are shown in Figure 2-12 and Figure 2-13 for 45°C and 70°C, respectively. The distributions have the similar shape as before but, as expected, retention times decrease with increasing temperature. Nonetheless, we observe that even at 70°C, almost 99% of DRAM pages have retention times of 1 second or more.

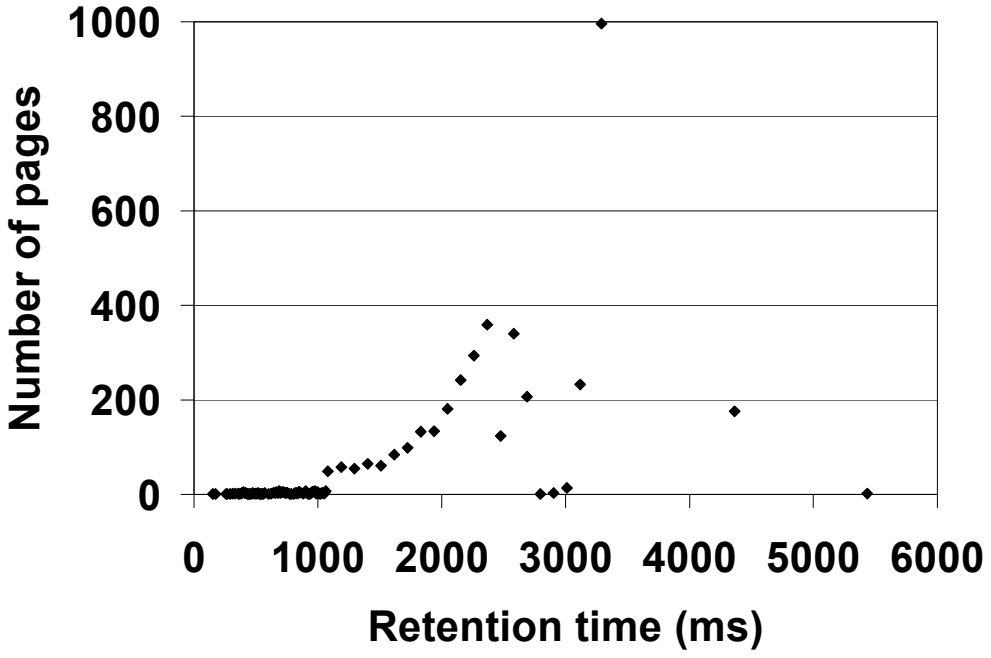


(a)

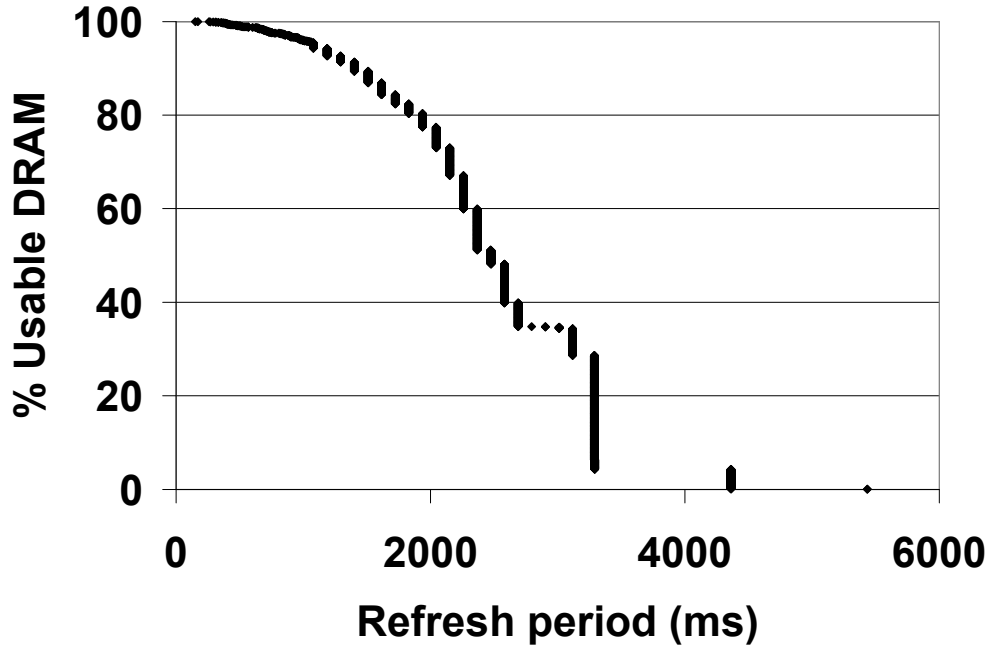


(b)

Figure 2-12. (a) Distribution and (b) cumulative distribution of page retention times, for the first DRAM bank at 45°C (test patterns all 1s).



(a)



(b)

Figure 2-13. (a) Distribution and (b) cumulative distribution of page retention times, for the first DRAM bank at 70°C (test pattern all 1s).

2.3 Retention-Aware Placement in DRAM (RAPID)

RAPID is a software-only solution for exploiting retention-time variations among different pages to virtually eliminate refresh power in contemporary off-the-shelf DRAMs, a major component of power consumption in standby operation. The key idea is that longer-retention pages are allocated before shorter-retention pages. This enables selecting a global refresh period that is the longest possible, based on the shortest retention time among occupied pages instead of the shortest retention time among all DRAM pages.

The primary software support is modifications to routines which allocate and deallocate physical pages in memory. For example, the Linux kernel's virtual memory manager maintains an *Inactive List*, a free-list of inactive physical pages (pages not touched for extended periods of time and deemed inactive). The Inactive List contains pages marked as "Inactive Dirty" or "Inactive Clean" (inactive dirty pages have to be flushed to their corresponding pages in the next-level non-volatile storage). The RAPID-1 allocation/deallocation routines merely exclude outlier pages from the initial Inactive List. The RAPID-2 and RAPID-3 allocation/deallocation routines transform the single Inactive List into multiple Inactive Lists, one for each retention time bin. For systems with a virtual memory manager, existing Inactive List management routines would be adapted for RAPID as just described. Many embedded systems do not implement virtual memory, in which case equivalent user routines would be adapted or substituted with the RAPID allocation/deallocation routines.

The three RAPID techniques are described in Sections 2.3.1 through 2.3.3. Some of the software implementation costs, in terms of software tables, are summarized in Section 2.3.4. Section 2.3.5 gives background on common refresh options available in off-the-shelf

DRAMs, and explains how the single refresh period selected by RAPID can be applied to commodity DRAMs without custom hardware support.

2.3.1 RAPID-1

RAPID-1 is a static approach in which a handful of shortest-retention outlier pages are discarded and are never populated. They are made unavailable to application programs by excluding these pages from the Inactive List when it is initialized.

We exclude all pages with retention times below 3 s (at room temperature), a total of 168 pages (out of 16,384 pages). This yields a static refresh period of 3 s. At this refresh period, 99% of the DRAM can still be utilized. Because the RAPID-1 refresh period is static, energy savings is independent of actual DRAM utilization.

2.3.2 RAPID-2

Pages are placed into bins according to their retention times. Ten bins are used. Using more than ten bins yields little additional power scaings. The 10 bins are equally spaced between 3 s and 50 s. The shortest retention time is 3 s because the 168 pages below 3 s are excluded, like in RAPID-1. Note that using only 10 bins can be exploited *a priori* to reduce the run time of the testing algorithm described in Section 2.1.1.

The original Inactive List is split into 10 Inactive Lists, one for each retention time bin. Pages within each Inactive List are unordered, as usual, so allocating and deallocating pages from a given list remains an O(1) operation.

The bins are ordered from longest to shortest retention time. (The retention time of each bin corresponds to the shortest retention time among pages in that bin.) When none of the

DRAM pages are populated, i.e., all Inactive Lists are full, the RAPID refresh period corresponds to the retention time of the highest (longest-retention) bin.

A page allocation request is satisfied from the highest bin that has a free page. If no more free pages are available in any of the bins that have populated pages, a page is allocated from the next lower bin. However, before the page is given over to the application program, the RAPID refresh period is decreased to accommodate the shorter retention time of the fresh bin.

When a page is deallocated, the page is returned to the Inactive List correponding to its retention time. The page's entry in the page bin table (see Section 2.3.4) indicates the RAPID-2 bin to which the page belongs. This facilitates returning the page to the correct Inactive List. Each bin knows how many pages (active plus inactive) belong to it. Thus, if an Inactive List becomes full when a page is returned to it, and the corresponding bin was previously the lowest bin with populated pages, then the RAPID refresh period is increased to the next higher bin that still has populated pages. Thus, once the refresh period has been decreased, increasing it again is opportunistic, since it depends on which pages happen to be freed by application programs.

2.3.3 RAPID-3

RAPID-3 builds on RAPID-2. It uses the same allocation/deallocation routines, enhanced with page migration to reconsolidate data from lower to higher bins whenever higher bins become fragmented, i.e., underutilized. Thus, RAPID-3 continuously reconsolidates data into the longest-retention bins possible. This causes the RAPID-3 refresh period to respond faster to decreases in DRAM utilization, compared to RAPID-2.

Many migration policies are possible, from migrating immediately when a page in a higher bin becomes free, to periodically reconsolidating pages *en masse*. For our experiments, we migrate immediately when an opportunity presents itself.

The downside of migration is that it incurs power overhead. In the results section, a single migration is given a power penalty that is 2X the power for a single refresh.

The refresh period of RAPID-2 and RAPID-3 depends on the actual dynamic DRAM utilization, because they are dynamic techniques. Note that outlier pages are discarded in both RAPID-2 and RAPID-3, so that the refresh power savings of RAPID-2 and RAPID-3 are at least as good as that of RAPID-1.

2.3.4 RAPID software storage costs

There are three major software data structures. The first is unique to RAPID. The last two have counterparts in existing page-managed systems.

- *Page bin table*. This table indicates which RAPID-2 bin each page belongs too. For 16K DRAM pages, there are 16K 4-bit entries (to encode up to 16 bins) for a storage cost of 8KB.
- *Logical-to-physical address translation table*. This is the well-known page table in traditional virtual memory systems or user-level counterpart in non-virtualized embedded systems. The RAPID allocation routine does not change the structure or management of the traditional page table, it only affects which DRAM pages are otherwise sequentially or randomly allocated. For 16K DRAM pages, a one-to-one mapping of logical to physical pages requires 16K 14-bit entries for a storage cost of 28KB.

- *Inactive list*. Although the single Inactive List is divided into multiple Inactive Lists for RAPID-2 and RAPID-3, the total storage cost remains the same as in traditional page-managed systems. For 16K DRAM pages, 16K list nodes are needed when no pages are populated. Assuming 8-byte nodes, the storage cost is 128KB.

2.3.5 Using RAPID refresh period with off-the-shelf DRAMs

Contemporary off-the-shelf DRAMs typically provide one or both of the following conventional refresh options: auto-refresh and self-refresh.

- *Self-refresh*. Self-refresh is implemented solely within the DRAM chip itself.
- *Auto-refresh*: The external memory controller issues regularly timed auto-refresh commands and the DRAM chip refreshes the next row in sequence. The memory controller does not send an address since the DRAM keeps track of the next row to be refreshed in sequence.

Self-refresh is significantly more power-efficient than auto-refresh, both in terms of DRAM power and external memory controller power. However, to the best of our knowledge, commodity DRAMs do not have a programmable self-refresh period. It is either 64 ms or temperature compensated, neither of which approaches the quasi-non-volatile RAPID refresh periods. If there exists a DRAM with a programmable self-refresh period, RAPID can exploit self-refresh directly by setting the self-refresh period to the RAPID refresh period.

An external memory controller typically supports a single programmable auto-refresh period. While directly compatible with the RAPID refresh period, this is not a good general solution because auto-refresh is too power inefficient during long standby periods, i.e., it is

better to power down the memory controller and DRAM leaving only self-refresh on. Auto-refresh may make sense during active periods. In fact, some DRAMs (like our ISSI DRAM) only support auto-refresh during active periods and self-refresh during standby periods, to explicitly avoid conflicts between internal self-refresh and external requests.

The CellularRAM™ [8] and UtRAM™ [64] for ultra-low-power systems only support the self-refresh option. Self-refresh is used during both active and standby periods. However, this self-refresh period is not programmable except for limited temperature compensation support. As is typical, self-refresh can be enabled/disabled via a configuration register. Thus, we propose the following approach for coupling RAPID with CellularRAM™ or UtRAM™:

- When RAPID is in use, self-refresh is disabled by default.
- RAPID sets up a periodic timer interrupt, its period equal to the RAPID refresh period.
- When the interrupt occurs, a lightweight handler enables the DRAM's self-refresh and sets up a near-term interrupt for 64 ms in the future. During the 64 ms interval, self-refresh transparently refreshes the entire DRAM. The near-term interrupt invokes a second lightweight handler that disables the DRAM's self-refresh.
- In this way, the entire DRAM is refreshed only once – in a burst fashion – every RAPID refresh period.

While we discussed the above approach in the context of DRAMs with only self-refresh, the approach can be adapted to exploit programmable auto-refresh during active periods and self-refresh during standby periods, for the ISSI DRAM and others with a similar refresh dichotomy.

2.4 Evaluation Methodology

Mobile devices often exhibit short bursts of activity (active mode) followed by long idle periods (standby mode) [26][34][35]. During active mode, DRAM refresh is typically a small component of overall system power, whereas, during standby mode, DRAM refresh may be the largest power component.

Obtaining activity traces of a real mobile device or sensor node is extremely difficult and beyond the scope of this work. We develop a simplified, yet effective evaluation technique, based on the above bimodal active/standby mode characterization of mobile devices to compare different refresh optimization techniques. This evaluation technique focuses on DRAM utilization over long periods of time.

We divide the long timeline into consecutive 100-second intervals and randomly inject 100-second active periods in place of otherwise standby periods, with a random probability that yields a typical active utilization, e.g., 5%. During active periods, we inject a random number of page requests. A request is equally probable to be an allocation vs. a deallocation so that DRAM utilization remains on average what the initial utilization was, making it possible to target a particular average DRAM utilization while still having significant utilization fluctuations during the timeline. Different refresh techniques may or may not exploit fluctuations in DRAM utilization, yielding a means for comparing techniques in a generic way. We compare the following refresh techniques:

- *TCR* (TCR): Optimal temperature-compensated refresh (TCR), i.e., the self-refresh period is based on the shortest retention time among all pages for the current temperature,

as measured in Section 2.2. We use this as the baseline since the default 64 ms self-refresh is overly pessimistic and TCR is available in some current DRAMs.

- *RAPID-1* (R-1)*, RAPID-2* (R-2)*, RAPID-3* (R-3): These are the new RAPID techniques.
- *HW-Multiperiod* (HW-M): A custom hardware solution, in which each page is refreshed at a tailored refresh period that is a multiple of the shortest refresh period among all DRAM pages [52].
- *HW-Multiperiod-Occupied* (HW-M-O): Same as HW-Multiperiod, but only pages that are currently occupied are refreshed.
- *HW-Ideal* (HW-I): An ideal custom hardware solution, in which each page is refreshed at its own tailored refresh period.
- *HW-Ideal-Occupied* (HW-I-O): Same as HW-Ideal, but only pages that are currently occupied are refreshed.

The simulated timeline is 24 hours. All simulations are configured for 5% activity and 95% standby operation. Average DRAM utilizations of 25%, 50%, and 75% are targeted. Three operating temperatures are simulated, 25°C (room temperature), 45°C, and 70°C.

The Micron CellularRAM™ [8] is the basis for refresh power. Using the CellularRAM™ refresh power at its default self-refresh period, refresh power can be calculated for an arbitrary refresh period via simple scaling.

2.5 Results

Figure 2-14 shows the refresh energy consumption (in mW·hours) for an average DRAM utilization of 75% at three different temperatures. TCR uses the worst-case refresh

period at a given temperature (500 ms at 25°C, 323 ms at 45°C, and 151 ms at 70°C). By simply discarding 1% of outlier pages, RAPID-1 yields 83% (25°C), 80% (45°C), and 70% (70°C) energy savings with respect to TCR. RAPID-2 yields 93% (25°C and 45°C) and 92% (70°C) energy savings with respect to TCR. RAPID-3 yields 95% (25°C and 45°C) and 93% (70°C) energy savings with respect to TCR. Another positive result is that RAPID-2 and RAPID-3 are nearly as effective as the custom hardware approaches.

Energy savings of RAPID-1 decreases only moderately as temperature increases, from 83% to 70% over the full temperature range. One factor contributing to this decline is that, some pages that lie above the RAPID-1 retention-time threshold at 25°C, lie below this threshold at 70°C. These pages become "outliers" with respect to the 25°C RAPID-1 standard, yet they are not excluded.

In contrast, energy savings of RAPID-2 and RAPID-3 hardly decline with increasing temperature. At 75% average DRAM utilization, typical RAPID-2 and RAPID-3 refresh periods are substantially long even at 70°C.

Energy consumption of RAPID-1 at 70°C (worst-case RAPID-1) is comparable to energy consumption of TCR at 25°C (best-case TCR). This implies that a worst-case, non-temperature-adjusted RAPID-1 implementation yields the same or better energy than TCR, suggesting a potentially simpler alternative to temperature-aware DRAM design. The same case can be made in even stronger terms, for non-temperature-adjusted RAPID-2 and RAPID-3 implementations.

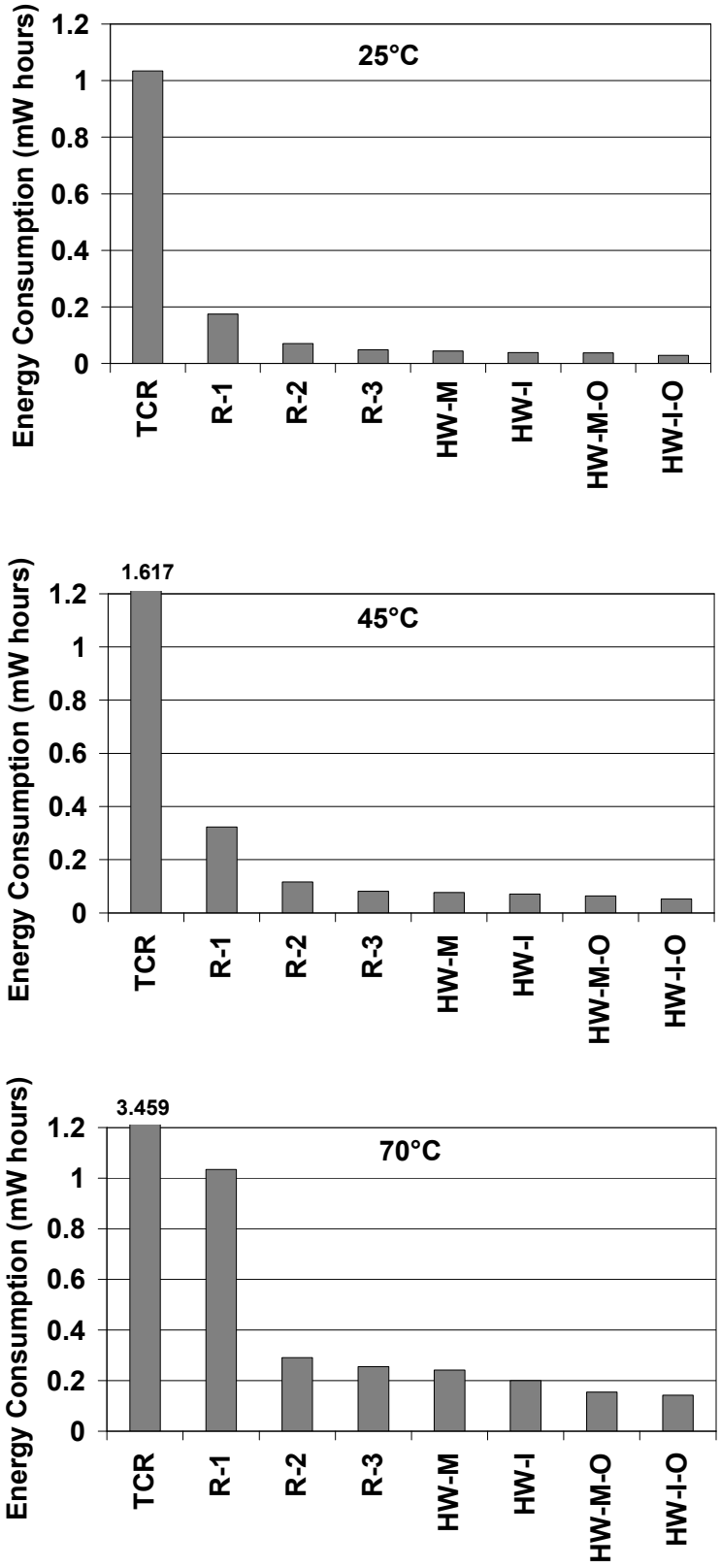


Figure 2-14. Refresh energy for 75% average DRAM utilization, at 25°C, 45°C, and 70°C.

Figure 2-15 shows refresh energy at 25°C for average DRAM utilizations of 75%, 50%, and 25%. A key observation is that, as the DRAM utilization decreases, both RAPID-2 and RAPID-3 yield more energy savings than the two hardware techniques HW-M and HW-I. At 75% average utilization, the energy consumption of RAPID-2 and RAPID-3 are slightly higher than HW-M and HW-I. Yet, at 50% and 25% average utilizations, both RAPID-2 and RAPID-3 consume less energy than HW-M and even HW-I. At lower DRAM utilizations, many lines are unoccupied and RAPID's refresh period exceeds the "average" period of the hardware techniques.

The ideal hardware technique, HW-I-O, provides a more reliable lower bound on energy savings because it does not refresh unoccupied pages. For 25% average utilization, RAPID-3 (and even RAPID-2) approaches this lower bound on energy savings.

However, even HW-I-O does not necessarily provide a lower bound on refresh energy. For non-RAPID implementations, pages are allocated without regard to page retention times. Thus, HW-I-O energy can differ for the same DRAM utilization, depending on which pages are allocated. For example, for 25% utilization, the energy of HW-I-O can vary from 0.01 mW·hours to 0.023 mW·hours, depending on whether the best or worst 25% of the DRAM pages are occupied. Energy of RAPID-3 at 25% utilization is 0.018 mW·hours. Therefore, it is possible for RAPID-3 to outperform HW-I-O.

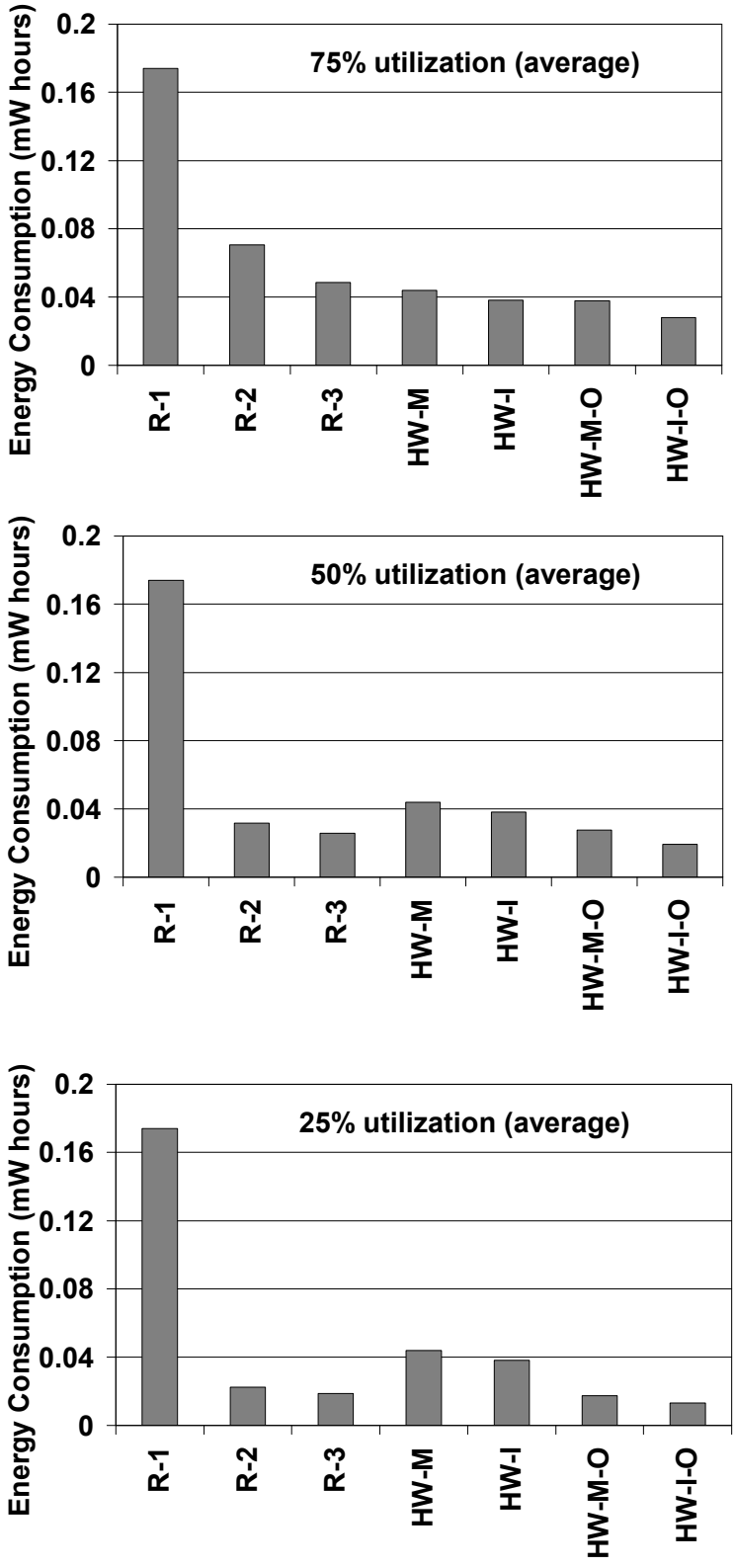


Figure 2-15. Refresh energy at 25°C for average DRAM utilizations of 75%, 50%, and 25%.

2.6 Implementation of RAPID on a Prototype Testbed

RAPID is a software-only technique that can be coupled with any off-the-shelf DRAM. Therefore, we implemented the first two RAPID techniques, i.e., RAPID-1 and RAPID-2, on the Ubicom IP3023 embedded system development board that contains the ISSI DRAM chip. Note, this is the same DRAM chip that we characterized previously.

The Ubicom IP3023 multithreaded processor supports up to 8 hardware threads. We use two hardware threads to emulate RAPID.

(i) *rapid_thread*: This thread services page allocation/deallocation requests from the application. It implements the RAPID-1 and RAPID-2 allocation/deallocation routines. After every allocation/deallocation, the refresh period is reevaluated and set accordingly. This thread also resumes the *refresh_thread* (below) after every RAPID refresh period or whenever the RAPID refresh period is decreased.

(ii) *refresh_thread*: This thread is dedicated solely to performing periodic RAPID refreshes, emulating the interrupt-based approach. When this thread is activated by *rapid_thread*, it enables auto-refresh. After 64 milliseconds (default DRAM refresh period), the thread disables auto-refresh and suspends itself.

Note, RAPID-1 and RAPID-2 are implemented similarly, the only difference is that the refresh period set by RAPID-1 is never modified, whereas the refresh period set by RAPID-2 changes with DRAM utilization.

Although, the above description is in terms of enabling/disabling auto-refresh, self-refresh could be used instead. Self-refresh is implicitly enabled and disabled when the

DRAM transitions into sleep mode and active mode, respectively, whereas auto-refresh needs to be explicitly enabled/disabled by the user or operating system.

To confirm that we are indeed enabling and disabling refreshes, we measured the power change on an oscilloscope and corroborated the measured auto-refresh power with auto-refresh power of the ISSI DRAM from its datasheets.

To confirm that the DRAM is refreshed once and only once every RAPID refresh period, we test pages that have retention times longer and shorter than the current RAPID refresh period, by writing a random data pattern to the pages. All pages that have retention times longer than the RAPID refresh period pass the test, while all pages that have retention times shorter than the RAPID refresh period fail the test. We also observe a power change on the oscilloscope whenever RAPID enables DRAM refresh. This power change on the oscilloscope corresponds to the auto-refresh power.

2.7 RAPID Summary

To summarize this chapter, RAPID virtually eliminates standby power in DRAM, yielding favorable scaling of standby power. Also, RAPID approaches the energy savings obtained by hardware techniques that require custom DRAM support to exploit retention time variations.

Chapter 3

ZettaRAM: Power-Scalable DRAM Alternative

ZettaRAM is a nascent memory technology based on conventional DRAM architectures, in which the conventional capacitor is replaced with a new molecular capacitor. The unusual characteristic of this new technology is that the amount of charge deposited on the molecular capacitor (critical for reliable sensing) is independent of write voltage. Charge-voltage decoupling holds the key for viable voltage scaling from one generation to the next, because voltage can be reduced arbitrarily while maintaining charge for reliable sensing. This opens up two new opportunities for power scaling in a DRAM-like architecture. (1) *Exploiting molecular engineering for long-term power scalability*: Precise tuning of molecular attributes provides an inexpensive path for scaling voltage, hence power, from one generation to the next. (2) *Intelligent management of ZettaRAM*: While the fixed charge is voltage-independent, speed is voltage-dependent. This tradeoff is architecturally managed by dynamically modulating voltage based on criticality of memory requests, further extending ZettaRAM's voltage scaling advantage with little system slowdown.

This chapter describes in detail these two power-scaling opportunities provided by ZettaRAM. This chapter is divided into the seven sections. Sections 3.1–3.5 describe the charge-voltage decoupling characteristic of the molecular capacitor and its possible implications. Sections 3.6 and 3.7 describe the two novel power-scaling opportunities provided by charge-voltage decoupling.

Section 3.1, Limitations of DRAM Voltage Scaling, explains the DRAM voltage scaling challenge.

Section 3.2, ZettaRAM Molecular Capacitor, provides background on the molecular capacitor, including basic read/write operation, our novel SPICE device model, and our derivation of charge density as a function of write voltage. This section also explains how charge-voltage decoupling overcomes DRAM voltage scaling limitations.

Section 3.3, SPICE Results, presents SPICE results for DRAM and ZettaRAM to derive nominal operating voltages and read/write latencies.

Section 3.4, Experimental Framework, describes the experimental framework for system-level simulations.

Section 3.5, Baseline DRAM Energy and Performance, presents system-level simulation results for the baseline DRAM system. Results in sections 3.6 and 3.7 are with respect to this baseline DRAM.

Section 3.6, Exploiting Molecular Engineering for Long-Term Power Scalability**,** explores first power-scaling opportunity. We study the impact of key molecular attributes on operating voltage and then presents system-level results for 23 different molecules.

Section 3.7, Intelligent Management of ZettaRAM**,** explores the second power-scaling opportunity. We present architectural management of ZettaRAM's unique speed/energy tradeoff to lower the operating voltage of ZettaRAM even further with little system slowdown.

3.1 Limitations of DRAM Voltage Scaling

A DRAM cell consists of a transistor-capacitor pair, as shown in Figure 3-1. The charge stored in the capacitor determines the logic state of the cell. The cell represents logic "1" if the capacitor is fully charged and logic "0" if the capacitor is fully discharged. To write the cell, a voltage is applied on the bitline and the access transistor is activated. The amount of charge deposited in the capacitor depends linearly on the magnitude of the applied bitline voltage. Therefore, 0 V is applied to write a "0" and a positive voltage V_{write} is applied to write a "1".

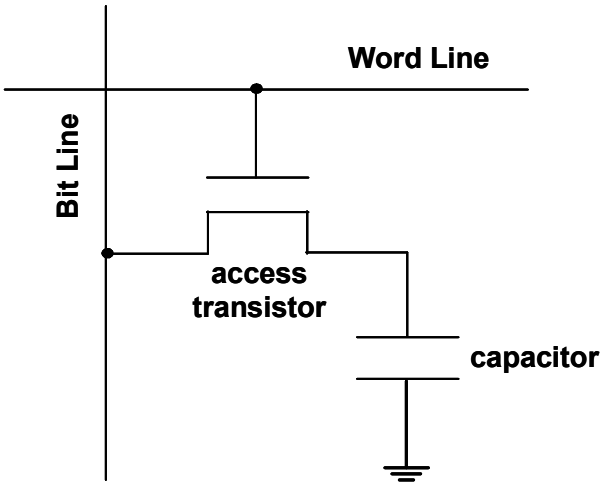


Figure 3-1. 1T-1C DRAM Cell.

To read the cell, the bitline is precharged to an intermediate voltage, typically half the write voltage. Charge flows to/from the DRAM capacitor depending on whether it was initially discharged/charged, respectively, resulting in a small negative/positive change in the bitline voltage, respectively. This small bitline voltage change, ΔV, is sensed by the sense amplifier.

The DRAM read and write voltages are ultimately determined by the voltage differential required by the sense amplifiers to reliably sense a "0" or "1". This voltage differential is expressed by the following equations, derived based on charge redistribution after a read operation. The right and left hand sides of the equation represent the total charge present in the cell capacitor and bitline (modeled as a large capacitor), before and after the read operation, respectively.

ΔV: voltage differential (bitline voltage change during read operation)

Q_{CELL}^{A}: charge in cell capacitor after read operation

Q_{CELL}^{B}: charge in cell capacitor before read operation

Q_{BL}^{A}: charge in bitline after read operation

Q_{BL}^{B}: charge in bitline before read operation

V_{CELL}: voltage over the cell capacitor

V_{PRE}: bitline precharge voltage before the read operation

C_S: cell capacitance

C_{BL}: bitline capacitance

AFTER REDISTRIBUTION BEFORE REDISTRIBUTION

$$Q_{CELL}^{A} + Q_{BL}^{A} = Q_{CELL}^{B} + Q_{BL}^{B} \quad \text{(EQ 3-1)}$$

$$Q_{CELL}^{A} + C_{BL} \cdot (V_{PRE} + \Delta V) = Q_{CELL}^{B} + C_{BL} \cdot V_{PRE} \quad \text{(EQ 3-2)}$$

Rearranging,

$$\Delta V = \frac{Q_{CELL}^{B} - Q_{CELL}^{A}}{C_{BL}} \quad \text{(EQ 3-3)}$$

A minimum voltage differential on the bitline (ΔV) is required for reliable operation of the sense amplifiers in the presence of various noise sources (sense-amplifier V_t mismatch, transient radiation induced charge, leakage, bitline coupling) [45]. Since the 4 Mb DRAM

generation (last decade), the bitline capacitance, C_{BL}, has remained steady at 350 fF [45][60]. The stored charge in the cell capacitor, Q_{CELL}^{B}, must be large enough to generate a bitline voltage change that can be reliably sensed, including compensating for various noise sources: radiation, leakage current, and electrical imbalances between pairs of bitlines. While noise induced by radiation decreases with each generation, leakage current and electrical imbalances remain nearly the same from one generation to the next. Therefore, the minimum charge required in the cell capacitor has not reduced much with each new generation [31].

The DRAM cell capacitor is constrained by conventional charge-voltage coupling, $Q_{CELL} = C_S \cdot V_{CELL}$. Now, if we want to lower DRAM voltage with each new generation, while keeping Q_{CELL} constant, C_S must actually be increased. Yet, DRAM scaling trends show that DRAM cell capacitance will remain steady at around 30-40 fF and will not scale with technology any further [45][60].

Summing up, the confluence of three factors, (i) a fixed minimum charge for reliable sensing, (ii) constant cell capacitance, and (iii) charge-voltage coupling, imposes a fundamental limit on DRAM voltage scaling.

3.2 ZettaRAM Molecular Capacitor

This section provides background on the ZettaRAM molecular capacitor, including basic read/write operation, our novel SPICE device model, and our derivation of charge density as a function of write voltage. The latter derivation provides insight into charge-voltage decoupling, for overcoming the voltage scaling limits of DRAM.

3.2.1 Molecule Description and Reading/Writing the Molecular Capacitor

A ZettaRAM memory cell is formed by replacing the DRAM capacitor with a molecular capacitor, composed of a self-assembled monolayer of charge-storage molecules (e.g., porphyrin molecules) sandwiched between two electrodes. An individual porphyrin molecule is shown in Figure 3-2(a) and the ZettaRAM molecular capacitor is shown in Figure 3-2(b). As shown in Figure 3-2(b), the molecules are attached to the lower metal plate or *working electrode*, via attachment groups called linkers. A linker is shown in detail in Figure 3-2(a) (its length can be customized). The second electrode or *counter electrode*, is interfaced to the molecules via an electrolyte. The electrolyte is conductive (it contains ion dipoles), for forming a closed circuit, but does not conduct electrons.

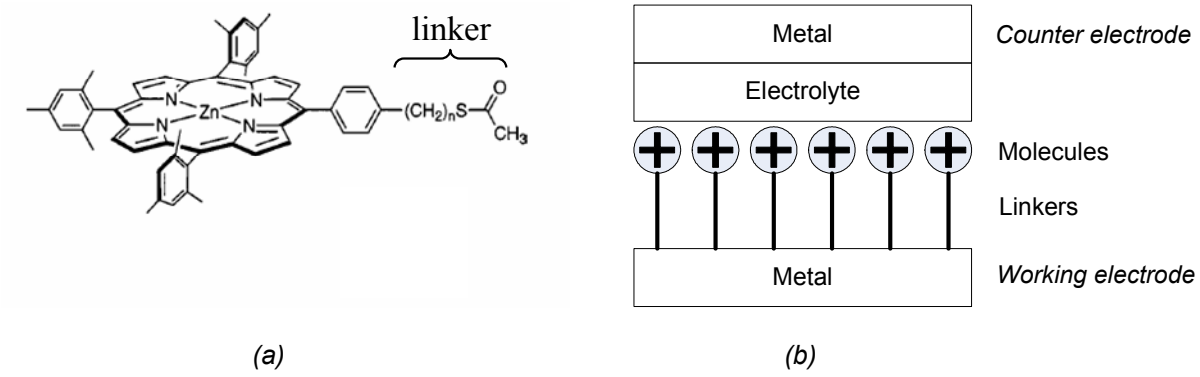


Figure 3-2. (a) Individual porphyrin molecule. (b) ZettaRAM molecular capacitor.

A molecule can be positively charged by removing an electron, referred to as *oxidation*. Oxidation corresponds to "writing" a logic 1. An electron can be added back to the positively charged molecule to return it to the uncharged state, referred to as *reduction*. Reduction corresponds to "writing" a logic 0.

The molecules are oxidized when the voltage applied across the molecules is greater than the oxidation potential (V_{ox}). As shown in Figure 3-3, this is achieved by applying a

positive voltage greater than V_{ox} on the working electrode relative to the counter electrode, causing electrons to tunnel from the molecules to the working electrode across the linkers. The molecules are reduced when the voltage applied across the molecules is lesser than the oxidation potential. As shown in Figure 3-4, this is achieved by applying a voltage less than V_{ox} on the working electrode relative to the counter electrode, causing electrons to tunnel from the working electrode to the molecules across the linkers.

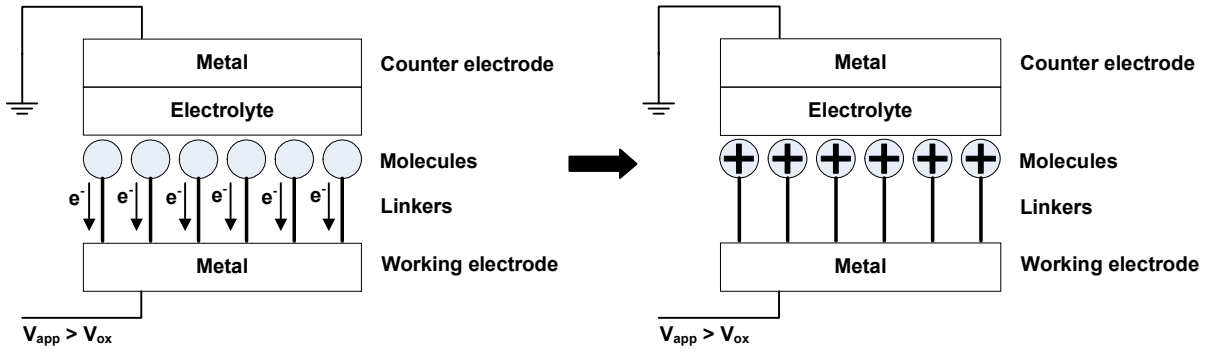


Figure 3-3. Oxidation process in molecular capacitor.

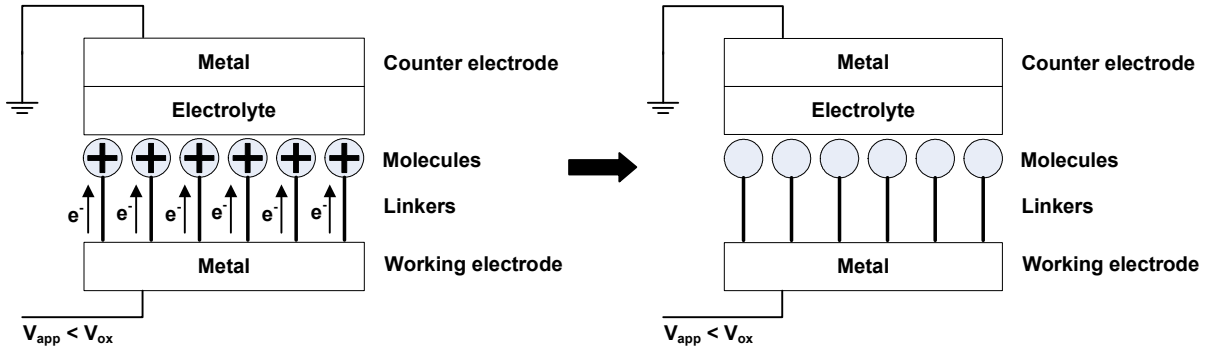


Figure 3-4. Reduction process in molecular capacitor.

A more accurate explanation is that oxidation and reduction are always taking place simultaneously, since any chemical reaction is a combination of forward and reverse reactions. Equilibrium is reached, at which point the rates of the forward and reverse reactions are equal. Although the rates are balanced at equilibrium, the molecules have a

strong tendency towards either the oxidized state or reduced state, depending on the applied voltage (above or below the oxidation potential, respectively). The Butler-Volmer equation in the next subsection expresses the non-equilibrium and equilibrium behavior.

Like reading conventional DRAM, reading ZettaRAM is destructive. To read the state of the molecules in a molecular capacitor, they are discharged (if they are initially charged). This is achieved by reducing them, i.e., the bitline is precharged to a voltage below the oxidation potential. The state of the molecules is sensed by detecting the presence (or absence) of a small voltage change on the bitline as the molecules are discharged (unless neutral), which is procedurally similar to sensing in conventional DRAMs.

An idiosyncrasy of the molecular capacitor, with regard to reading, is that the bitline needs to be precharged to a specific voltage below the oxidation potential called the *open circuit potential* (e.g., OCP = 0.3V for the porphyrin molecule used in section 3.3.3) [58]. The molecular capacitor is an electrochemical cell in which the redox species is the porphyrin molecules. The OCP is a well-known artifact of electrochemical cells. Reading at the OCP prevents discharging of the "double-layer capacitance", which would otherwise drown out discharging of the molecules themselves.

Technological problems that have to be considered when integrating the molecular capacitor in a standard CMOS process include potential degradation of molecules during high-temperature processing steps and potential defects due to non-uniform layers of porphyrin molecules on silicon. These issues have been successfully addressed while integrating the molecular capacitor into a standard CMOS process in small volumes in fabrication labs. K. Roth et al. [56] and Z. Liu et al. [44] showed that molecules can withstand the highest temperature during DRAM processing. K. Roth et al. [57][58] showed

the formation of a uniform layer of porphyrin molecules on the silicon surface. Currently, one of the main technological challenges is volume production, as is often the case with new technologies.

3.2.2 SPICE Model of Molecular Capacitor

The oxidation/reduction reaction is shown below, where A is the porphyrin molecule [56].

$$A \leftrightarrow A^{+} + e^{-} \qquad \text{(EQ 3-4)}$$

In non-equilibrium (charging or discharging), the net rate of oxidation or reduction – i.e., the net current – is exponentially dependent on the difference between the applied voltage and the oxidation potential. This current is expressed by the Butler-Volmer kinetic model [3], shown below, which forms the basis of our SPICE model.

$$I = F \cdot k^{0} \cdot \left([A] \cdot e^{(1-\alpha)\left(\frac{F}{RT}\right)(V - V_{ox})} - [A^{+}] \cdot e^{-\alpha\left(\frac{F}{RT}\right)(V - V_{ox})} \right) \qquad \text{(EQ 3-5)}$$

Parameters above are as follows: k^0 = standard rate constant, α = transfer coefficient, F = Faraday constant, R = gas constant, T = temperature, V = applied voltage, V_{ox} = oxidation potential, [A] = concentration of non-oxidized molecules (in moles per unit area), and $[A^+]$ = concentration of oxidized molecules.

The current I determines the *intrinsic* speed of reading and writing the molecules. In a complete memory circuit, the overall read and write speeds are determined by several interacting components. We build the complete memory circuit using SPICE. The SPICE simulator correctly solves for currents and voltages at all nodes, accurately reflecting the

interaction between the molecular capacitor and the rest of the circuit. The SPICE model and results are presented in Section 3.3.

3.2.3 Highly Non-linear Capacitance: Charge-Voltage Decoupling

The oxidation/reduction reactions shown in EQ 3-4 eventually reach an equilibrium. The net current is zero at this equilibrium. We can derive the amount of charge ($Q_{cell} = [A^+]$) at equilibrium as a function of the write voltage, by substituting I=0 in the Butler-Volmer equation (EQ 3-5). (This gives us the effective capacitance of the molecular capacitor since capacitance expresses Q as a function of V.) Doing so yields the following $Q_{cell}(V)$.

$$Q_{cell}(V) = [A]_0 \cdot \left[\frac{1}{1 + e^{-\frac{F}{RT}(V - V_{ox})}} \right] \qquad \text{(EQ 3-6)}$$

where, $[A]_0$ is the total molecule concentration, equal to the sum of [A] and $[A^+]$. EQ 3-6 is derived in Appendix A. EQ 3-6 is the basis for the unusual charge density graph shown in Figure 3-5 (reproduced from Figure 1-4 in Introduction, Section 1.2.1).

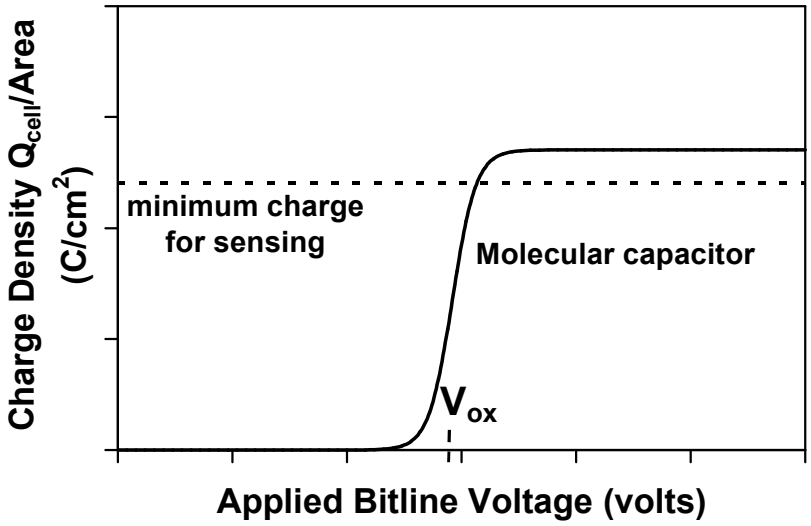


Figure 3-5. Charge density vs. voltage for a molecular capacitor in ZettaRAM.

The exponential term in the denominator becomes negligible as V is increased slightly above V_{ox}. Thus, just above V_{ox}, the molecular capacitor is nearly fully charged. Conversely, the exponential term in the denominator grows large as V is decreased slightly below V_{ox}. Thus, just below V_{ox}, the molecular capacitor is nearly fully discharged. In other words, while voltage still controls charging and discharging, depending on whether the applied voltage is above or below the threashold, V_{ox}, the *amount* of charge is independent of the applied voltage, as shown in Figure 3-5. In contrast, with a conventional DRAM capacitor, voltage has dual roles of controlling charging/discharging and influencing the amount of charge.

3.2.4 Implications of Charge-Voltage Decoupling and Speed-Voltage Dependence

The primary implication of charge-voltage decoupling is that reads/writes can be performed at lower voltages while maintaining the minimum fixed charge for reliable sensing. The write voltage can be lowered from one generation of ZettaRAM to the next, by synthesizing new molecules with lower oxidation potentials via favorable choices of molecule groups and linkers. Precise tuning of molecular attributes provides an inexpensive path to scaling voltage from one generation to the next.

While the fixed charge is independent of voltage, the speed of charging/discharging the molecules depends exponentially on the difference between the applied voltage and oxidation potential (slower with smaller differential), as illuminated by EQ 3-5 in section 3.2.2. The overall speed of ZettaRAM is determined by either the speed of charging/discharging the molecules or the conventional peripheral circuitry used to access the molecular capacitor, whichever is slower. Therefore, to achieve the same performance of conventional DRAM, the write voltage is padded with respect to the oxidation potential. There is room to reduce

the padding by architecturally managing ZettaRAM's unique speed-voltage dependence. We dynamically modulate the write voltage based on the criticality of memory requests, further extending ZettaRAM's voltage scaling advantage with little system slowdown.

3.3 SPICE Results

In this section, SPICE simulations are performed to determine (1) operating voltages and (2) read/write latencies, for both DRAM and ZettaRAM. We use 0.18μ technology and assume a 10:1 ratio between bitline capacitance and cell capacitance [28]. The sense amps are designed accordingly, and are based on designs in the DRAM literature [30].

For the ZettaRAM SPICE experiments presented in this section, we use one particular molecule type that yields similar system performance and energy as conventional DRAM, without our architectural management technique. In Sections 3.6 and 3.7, we consider many different molecules and architectural management of ZettaRAM, respectively.

3.3.1 SPICE Models

Figure 3-6(a) shows the SPICE model of the conventional capacitor integrated into a larger SPICE model of the DRAM architecture, including bitline, wordline, access transistor, and sense amplifier.

Figure 3-6(b) shows the SPICE model of the molecular capacitor. The voltage-controlled current source implements EQ 3-5. The current depends on three variables, [A], [A^+], and V.

Figure 3-6(c) shows the SPICE model of the molecular capacitor integrated into a larger SPICE model of the ZettaRAM architecture, including bitline, wordline, access transistor,

and sense amplifier. The only difference between the ZettaRAM and DRAM SPICE models, is the type of capacitor used inside the cell (molecular vs. conventional, respectively).

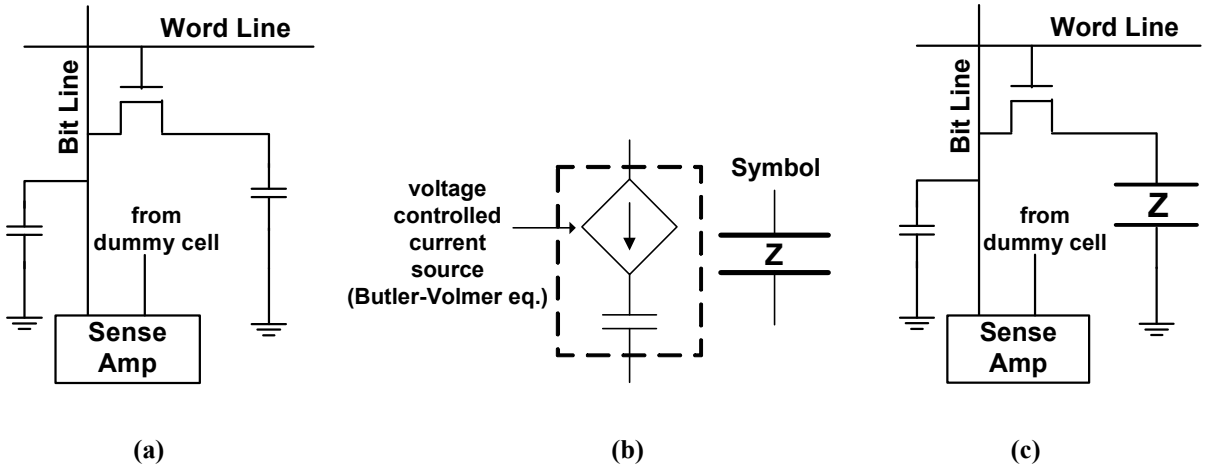


Figure 3-6. (a) DRAM circuit. (b) SPICE device model of molecular capacitor. (c) ZettaRAM circuit.

3.3.2 DRAM Spice Results

The linear relationship between charge and voltage in a conventional capacitor places a lower bound on the DRAM write voltage for writing a "1". Below this voltage, the charge deposited in the capacitor is not enough for the sense amplifier to reliably sense a "1" during a later read operation. We determine this lower bound experimentally and call this write voltage $V_{d\_write\_1}$. Searching in increments of 0.05 V, we determined $V_{d\_write\_1}$ = 1.25 V. The graph in Figure 3-7 shows that writing the DRAM capacitor at 1.2 V causes sensing to fail during a later read operation, since there is not enough charge in the capacitor.

Next, we determine the write latency of DRAM. The graph in Figure 3-8 shows that SPICE produces a DRAM write latency of 8.6 ns for $V_{d\_write\_1}$ = 1.25 V.

Finally, we determine the read latency of DRAM. SPICE produces a read latency of 29 ns (see Figure 3-9).

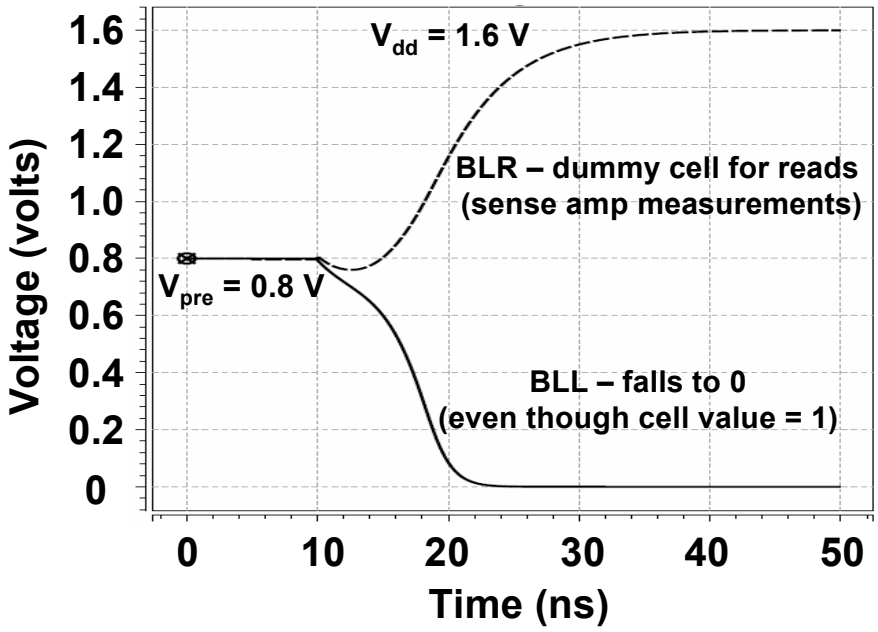


Figure 3-7. Writing DRAM capacitor below 1.25 V causes subsequent read operation to fail.

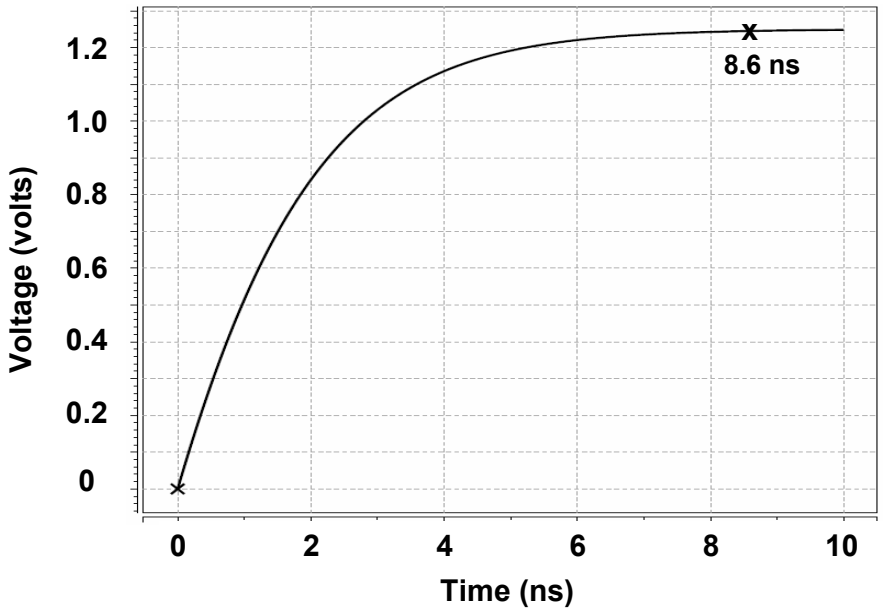


Figure 3-8. Write latency of DRAM is 8.6 ns.

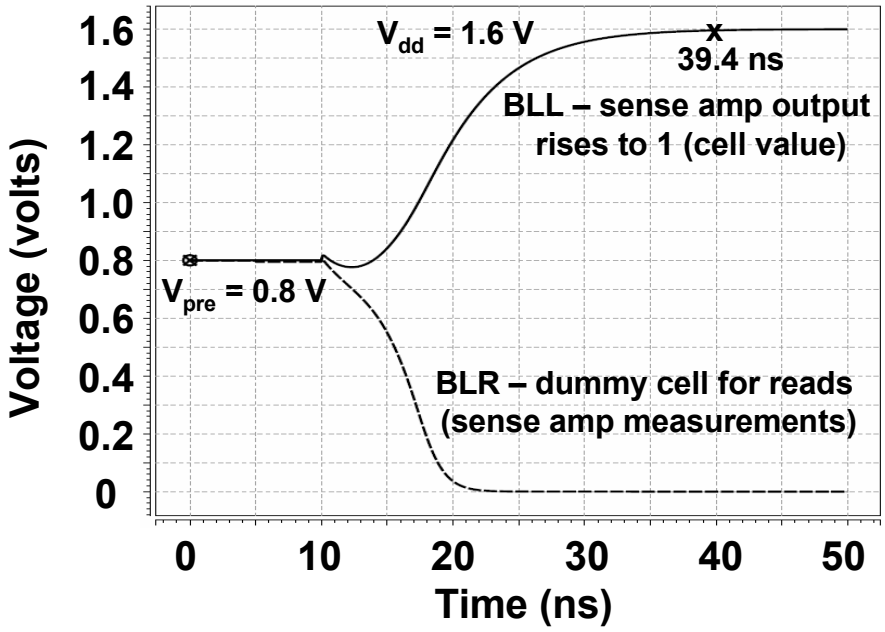


Figure 3-9. Read latency of DRAM is 29 ns.

3.3.3 ZettaRAM Spice Results

The SPICE results for ZettaRAM are shown for one particular molecule type. The molecule type is one of the first synthesized porphyrin molecules. Moreover, this molecule yields similar system performance and energy as conventional DRAM without our architectural management technique.

In the previous subsection, we showed that the conventional capacitor of DRAM is not sufficiently charged below 1.25 V, from the standpoint of correct sensing during a later read operation. On the other hand, writing the molecular capacitor at a voltage as low as 1.0 V (and lower) results in correct sensing during a later read operation, as shown in Figure 3-10.

Next, we determine the write latencies of ZettaRAM as a function of the ZettaRAM write voltage, $V_{z\_write\_1}$. In the first experiment, we use DRAM's minimum write voltage, $V_{d\_write\_1}$ = 1.25 V. The graph in Figure 3-11 shows that the ZettaRAM write latency at this voltage is 8.2 ns, similar to the DRAM write latency (8.6 ns) reported in the previous

subsection. This means that, for $V_{z\_write\_1} = V_{d\_write\_1}$, the conventional peripheral circuitry used to access the molecular capacitor limits the speed, not the intrinsic speed of the molecules.

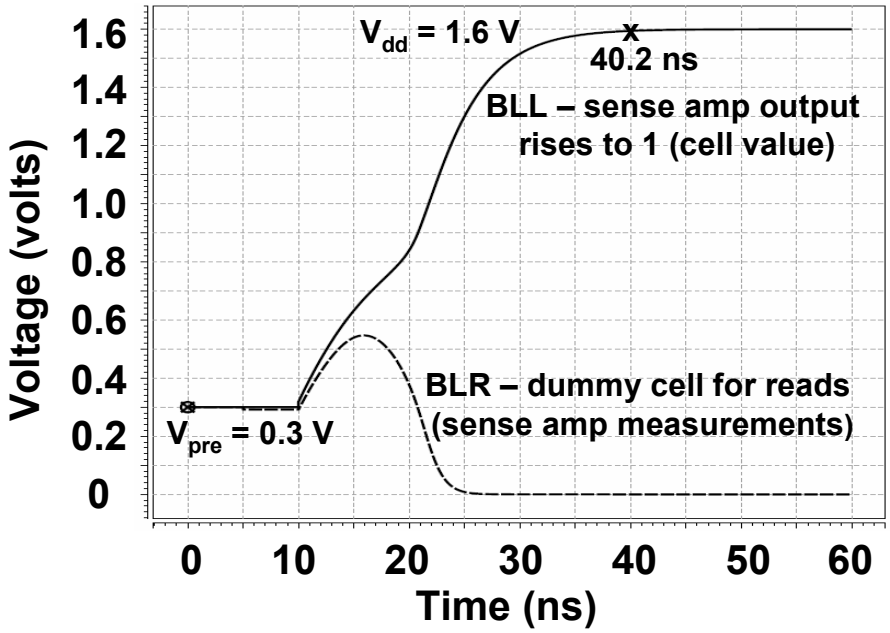


Figure 3-10. Writing molecular capacitor as low as 1.0 V subsequently results in correct sensing.

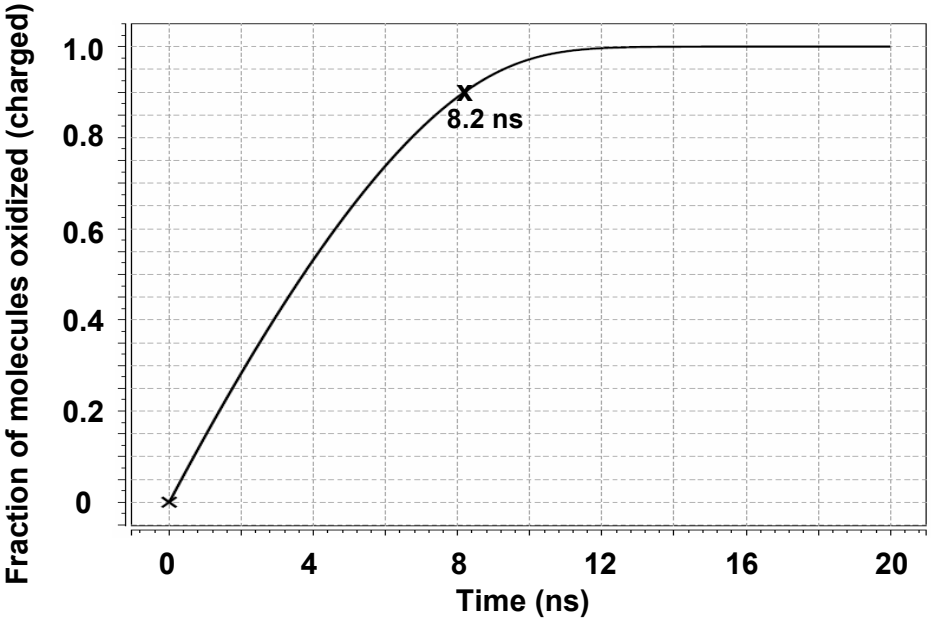


Figure 3-11. Write latency of ZettaRAM is 8.2 ns.

The ZettaRAM molecular capacitor can be reliably written below 1.25 V, although the intrinsic speed of the molecules begins to limit the overall write speed at lower voltages. The SPICE results in Figure 3-12 show exponentially increasing write latency with decreasing write voltage: 9 ns at 1.2 V, 29 ns at 1.1 V, and 166 ns at 1.0 V.

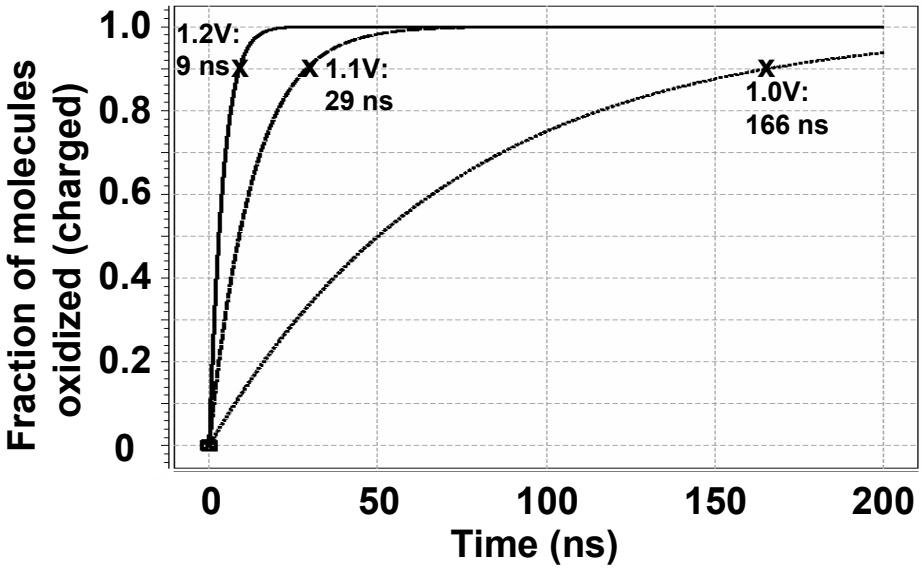


Figure 3-12. ZettaRAM write latency (90% of molecules oxidized) for three applied voltages.

Reading is competitive with conventional DRAM because the read voltage (OCP, Section 3.2.1) is typically sufficiently lower than V_{ox}, such that the molecule current is much faster than the peripheral sensing apparatus, and thus does not limit the speed of reading. Thus, the read latency of ZettaRAM is dictated by the peripheral sensing circuit, common to both DRAM and ZettaRAM. This is confirmed by SPICE simulations. The SPICE result in Figure 3-10 shows that the latency of reading ZettaRAM is 30 ns, similar to the read latency of DRAM (29 ns) measured in section 3.3.2 (Figure 3-9).

Reading the molecular capacitor is tantamount to writing "0", since the read voltage is below V_{ox}, fully discharging the molecular capacitor. So far, we only discussed multiple write voltages for writing a "1". For writing a "0", we consider only a single write voltage

equal to the read voltage. Incidentally, this is a fast write voltage. Bitline operations always alternate between reading (open page) and writing (close page), so keeping the write "0" voltage the same as the read voltage eliminates many bitline transitions altogether. Considering slower write "0" voltages between the read voltage and V_{ox} will only increase the number of bitline transitions, thus increasing energy consumption. This will become clearer in section 3.3.5, where we summarize bitline transitions for ZettaRAM and DRAM.

3.3.4 Retention Time Comparison of DRAM and ZettaRAM

To conclude our SPICE analysis, this subsection compares the retention times of the two technologies, DRAM and ZettaRAM. The retention times of the two technologies are comparable because leakage is an artifact of the access transistor, and the initial stored charge is the same. This is confirmed by the SPICE results shown in Figure 3-13. For example, at 40 ms, the conventional capacitor and molecular capacitor retain 32% and 51% of the initial charge, respectively. The molecular capacitor demonstrates an improved decay curve at the beginning. The retention times of both memories can be improved by applying a negative substrate bias, reducing the leakage current of the access transistor. What we want to demonstrate here is the comparable retention times of the two technologies.

ZettaRAM exhibits an unconventional linear retention time curve. The molecular capacitor's metal plate that is connected to the access transistor (i.e., working electrode) decays sharply to V_{ox} and stays close to this voltage level throughout the decay process. In fact, this is due to the non-linear charge/voltage characteristic described in Section 3.2.3 and depicted in Figure 3-5 (charge drops from fully charged to fully discharged over a narrow voltage interval). Since voltage is nearly constant throughout the decay process, leakage current is nearly constant throughout as well. (Leakage current varies with the voltage at the

drain of the access transistor.) Since leakage is nearly constant, the charge decays linearly with time. We confirmed that the slope corresponds to the leakage current of the access transistor.

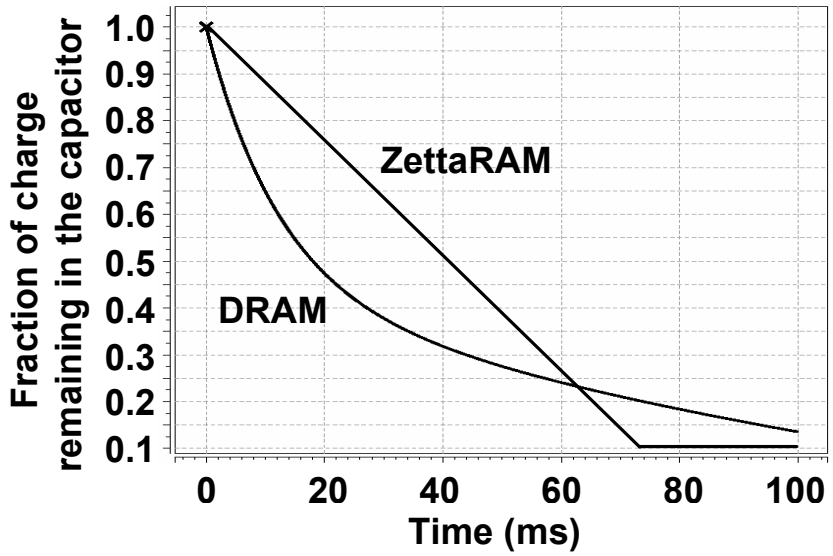


Figure 3-13. Retention times.

3.3.5 ZettaRAM and DRAM: Comparison Summary

Table 3-1 and Table 3-2 summarize the SPICE results for DRAM and ZettaRAM. For ZettaRAM, we include results for two different molecule types. Table 3-1 shows read/write latencies. Table 3-2 shows operating voltages and implied bitline voltage transitions that depend on consecutive memory operations.

Table 3-1. Read/write latencies for DRAM and ZettaRAM, for two example molecules.

Characteristic	DRAM	ZettaRAM	
		Molecule 1 (V_{ox} = 0.73 V)	Molecule 2 (V_{ox} = 0.48 V)
Precharge time (write an entire row)	9 ns	function of applied voltage [9 ns @ 1.2V – 166 ns @ 1V]	function of applied voltage [9 ns @ 0.79V – 166 ns @ 0.59V]
Row access time (read an entire row)	29 ns	30 ns	30 ns

Bitline energy, which can constitute up to 96% of overall energy in DRAM [29], depends on the applied voltage and magnitude of the voltage change ($E_{BL} = C_{BL} \cdot V_{BL} \cdot \Delta V_{BL}$, where E_{BL} is the bitline energy, C_{BL} is the bitline capacitance, V_{BL} is the bitline voltage, and ΔV_{BL} is the bitline voltage change). The actual magnitude of bitline transitions depends on the nature of consecutive operations which cause a voltage change on the bitline – read, write 0, and write 1. We now analyze these individual bitline transitions in depth for both DRAM and ZettaRAM. We observe that both the applied voltage and magnitude of the voltage change can be much lower for ZettaRAM than DRAM, yielding significant energy savings.

Table 3-2 shows operating voltages and bitline voltage transitions for DRAM and ZettaRAM, for two different molecules. Because L2 requests are always serviced from the page held in the row buffer, bitline operations always alternate between reading (open page) and writing (close page). This yields only four valid transitions: read followed by write-0, read followed by write-1, write-0 followed by read, and write-1 followed by read. The first row in the table shows the percentage breakdown of these four transitions. One benchmark from the SPEC2K benchmark suite *mcf* is shown. The other benchmarks show similar breakdowns. The second row shows the DRAM voltage differential for each transition, using the voltages derived in Section 3.3.2. Table entries for positive voltage transitions are highlighted, which we use in the energy accounting. Although the previous SPICE experiments used V_{DD} = 1.6 V due to our available technology files (and a corresponding read precharge voltage of 0.8 V), for energy accounting, we use $V_{DD} = V_{d\_write\_1}$. This adjustment minimizes DRAM energy, by applying a lower voltage differential for the higher percentage write-0→read transitions.

The third and fifth rows show ZettaRAM voltage differentials, using fast writes, for two different molecule types ($V_{z\_write\_1\_fast}$=1.2 V for *Molecule 1* and $V_{z\_write\_1\_fast}$=0.79 V for *Molecule 2*). The fourth and sixth rows show ZettaRAM voltage differentials, using slow writes, for the two molecule types ($V_{z\_write\_1\_slow}$=1.0 V for *Molecule 1* and $V_{z\_write\_1\_slow}$=0.59 V for *Molecule 2*). Because the write-0 and read voltages are the same (Section 3.3.3), two of the transitions incur no voltage change.

Recall that, fast writes match the latency of DRAM writes, and slow writes increase latency for further energy savings. In Section 3.7, we propose a hybrid write policy, wherein fast writes are used to service critical memory requests and slow writes are used to service non-critical memory requests.

Table 3-2. Operating voltages and bitline voltage transitions for DRAM and ZettaRAM, for two example molecules.

		Bitline Transition			
		Read → write 0	Read → write 1	write 0 → read	write 1 → read
% of all transitions, benchmark = *mcf*		28.46%	21.48%	28.48%	21.58%
Conventional DRAM ΔV		$-(V_{DD}/2)$ = -0.625	$+(V_{d\_write\_1}-V_{DD}/2)$ = 0.625	$+(V_{DD}/2)$ = 0.625	$-(V_{d\_write\_1}-V_{DD}/2)$ = -0.625
MOLECULE 1 V_{ox} = 0.73 V, V_{ocp} = 0.3 V	Fast ZettaRAM ΔV ($V_{z\_write\_1\_fast}$ = 1.2 V)	0	$+(V_{z\_write\_1\_fast}-V_{ocp})$ = 0.9	0	$-(V_{z\_write\_1\_fast}-V_{ocp})$ = -0.9
	Slow ZettaRAM ΔV ($V_{z\_write\_1\_slow}$ = 1.0 V)	0	$+(V_{z\_write\_1\_slow}-V_{ocp})$ = 0.7	0	$-(V_{z\_write\_1\_slow}-V_{ocp})$ = -0.7
MOLECULE 2 V_{ox} = 0.48 V, V_{ocp} = 0.17 V	Fast ZettaRAM ΔV ($V_{z\_write\_1\_fast}$ = 0.79 V)	0	$+(V_{z\_write\_1\_fast}-V_{ocp})$ = 0.62	0	$-(V_{z\_write\_1\_fast}-V_{ocp})$ = -0.62
	Slow ZettaRAM ΔV ($V_{z\_write\_1\_slow}$ = 0.59 V)	0	$+(V_{z\_write\_1\_slow}-V_{ocp})$ = 0.42	0	$-(V_{z\_write\_1\_slow}-V_{ocp})$ = -0.42
Baseline DRAM ΔV (V_{read} = $V_{d\_write\_0}$ = $V_{ocp\_molecule1}$=0.3V)		0	$+(V_{d\_write\_1}-V_{read})$ = 0.95	0	$-(V_{d\_write\_1}-V_{read})$ = -0.95

The lack of any write-0→read transitions gives ZettaRAM a substantial energy advantage over conventional DRAM. Conceivably, the same strategy of unifying the read potential and the write-0 potential may be applicable in future DRAMs. To level the playing field, we enhance the DRAM by lowering the read potential from $V_{DD}/2$ and raising the write-0 voltage from 0 V, both to V_{ocp} of *Molecule 1*. (Like ZettaRAM, the enhanced DRAM sense amp senses logic "0" via the absence of a bitline shift.) This enhanced DRAM is the baseline for all architectural experiments performed in the subsequent sections. Voltage differentials for this baseline DRAM are shown in the last row of Table 3-2.

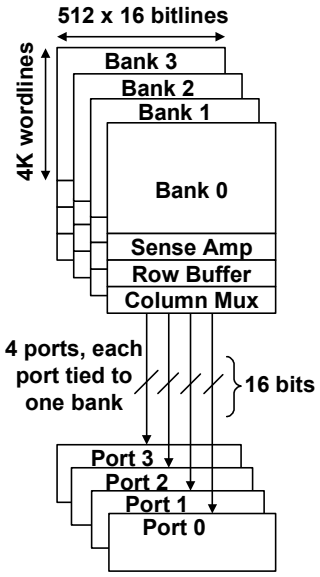
3.4 Experimental Framework

3.4.1 Memory Simulator: Modeling Timing

The memory simulator models the internal state and operation (timing and functionality) of ZettaRAM. The interleaved ZettaRAM memory system, shown in Figure 3-14, is based on the synchronous DRAM (SDRAM) architecture [46].

The ZettaRAM memory system has 4 independent ports, with each port tied to a bank. The memory controller maps physical addresses to memory addresses (bank id, row id, and column id) and schedules pending memory requests. The memory controller maintains a separate queue of pending memory requests for each bank. There are two types of memory requests initiated by the L2 cache, fetch block and writeback block.

Memory access reordering is used by default. Fetch requests that circumvent queued writeback requests need to search the queue for possible read-after-write hazards. Where indicated, we also investigate configurations with memory access reordering disabled.



ZettaRAM configuration

Chip configuration	8 Mb x 16
# chips	4
# banks per chip	4
# bits per column per chip	16
Data bus width	64 bits
Row Addressing	4K (A0 – A11)
Column Addressing	512 (A0 – A8)
Bank Addressing	4 (BA0 – BA1)
Row Access Time - RAS	30 ns
Col. Access Time - CAS	16 ns
Precharge Time - PRE	Variable (voltage dependent)

Figure 3-14. Interleaved ZettaRAM memory system.

A DRAM *page* is a row in memory that is read into the row buffer to service memory requests. The memory controller can use one of two different policies to manage pages – open page policy and close page policy. In the close page policy, a page is immediately written back into its memory array (closed) after servicing the memory request. In the open page policy, a page is held in the row buffer (left open) after reading the page into the row buffer. Thus, the row buffer serves as a cache in DRAM. By keeping the page open, subsequent accesses to the same page do not incur the penalty of opening the page. However, if there is a request to a different page in the same bank, the open page policy incurs the penalty of closing the current page before opening the new page, thus sometimes increasing the wait time of memory requests. Nonetheless, we find that the open page policy significantly outperforms the close page policy. Since we model a high-performance memory system, we consider only the open page policy in our simulations.

The time components in a main memory system are given in Table 3-3. The minimum latency for servicing memory requests and port occupancy times are given in Table 3-4, for

various scenarios in the context of the open page policy. Port occupancy time refers to the time for which the port is tied up by a request.

Table 3-3. Time components in a main memory system.

Component	Description
t_{RAS}	Row Access Strobe (RAS) : Time to read a page into the sense amplifiers/row buffer.
t_{CAS}	Column Access Strobe (CAS): Time to output the requested data (specific column) from the sense amplifiers/row buffer to the memory controller.
t_{PRE}	Precharge (PRE): Time to write a page from the sense amplifiers/row buffer back into the memory array.

Table 3-4. Request latencies and port occupancy times for the open page policy, under various scenarios.

Scenario	Request Latency	Port Occupancy
Requested page is open.	t_{CAS}	t_{CAS}
Requested page is closed, no page currently open.	$t_{RAS} + t_{CAS}$	$t_{RAS} + t_{CAS}$
Requested page is closed, another page is currently open.	$t_{PRE} + t_{RAS} + t_{CAS}$	$t_{PRE} + t_{RAS} + t_{CAS}$

3.4.2 Memory Simulator: Modeling Energy

The memory system can be divided into three major blocks to compute the power consumption: the memory cell array, the decoders (row and column), and the periphery (sense amps). The active power equation is given as follows [29].

$$P = V_{DD} \cdot I_{DD} \quad \text{(EQN 3-7)}$$

$$I_{DD} = m \cdot i_{act} + m \cdot (n-1) \cdot i_{hld} + (n+m) \cdot C_{DE} \cdot V_{INT} \cdot f + C_{PT} \cdot V_{INT} \cdot f + I_{DCP} \quad \text{(EQN 3-8)}$$

V_{DD} is the external supply voltage, m and n are the number of bitlines and wordlines in the memory array, respectively, i_{act} is the effective current of the active or selected cells, i_{hld} is the effective current of the inactive or non-selected cells, C_{DE} is the output node capacitance of each decoder, V_{INT} is the internal supply voltage, C_{PT} is the total capacitance of CMOS logic and driving circuits in the periphery, I_{DCP} is the total static (DC) current of the

periphery, and f is the frequency. The periphery includes the column circuitry, differential amplifiers on the I/O lines, refresh related circuits, and on-chip voltage converters.

The first term in EQ 3-8 is the active current, the second term is the data retention current, the third term is the decoder current, the fourth term is the current in the periphery, and the final term is the static current in the periphery.

At high operating frequencies, I_{DCP} and i_{hld} become relatively small compared to other current components [29]. The decoder charging current is negligibly small in modern DRAMs [29]. The bitline charging current (i_{act}) is $C_{BL} \cdot \Delta V_{BL} \cdot f$ (ΔV_{BL} is the voltage swing on the bitlines). Therefore, EQ 3-8 can be rewritten as,

$$P \approx V_{DD} \cdot [m \cdot C_{BL} \cdot \Delta V_{BL} + C_{PT} \cdot V_{INT}] \cdot f \qquad \text{(EQN 3-9)}$$

The active bitline dissipation power dominates the above equation [29]. So we get,

$$P \approx V_{DD} [m \cdot C_{BL} \cdot \Delta V_{BL}] \cdot f \qquad \text{(EQN 3-10)}$$

Thus, total memory system power is dominated by the bitline energy, i.e., power required to charge the bitlines for opening and closing pages and can constitute up to 96% of the total memory system energy [29].

Thus, in our experiments, we measure bitline energy consumption in the main memory. We track the voltage states of all bitlines in order to measure the energy required to charge the bitlines for a particular memory operation.

Assuming a single voltage supply (V_{DD}), the energy to charge a bitline is $E_{bitline} = C_{BL} \cdot V_{DD} \cdot (\Delta V_{BL}) = C_{BL} \cdot V_{DD} \cdot (V_{z\_write\_1} - V_{ocp})$. Thus, dynamically adjusting the write-1 voltage yields linear energy scaling. If we use a dedicated voltage supply for charging the bitline ($V_{z\_write\_1}$), then $E_{bitline} = C_{BL} \cdot V_{z\_write\_1} \cdot (V_{z\_write\_1} - V_{ocp})$. Now, dynamically adjusting

the write-1 voltage yields quadratic energy scaling. We assume dual voltage supplies for the dual write voltages ($V_{z\_write\_1\_fast}$ and $V_{z\_write\_1\_slow}$) for our dynamic voltage modulation technique. The supplies can be implemented using high-efficiency DC-DC converters [9]. Dual voltages were implemented in drowsy caches and selected in one to two cycles via a MUX to drive the bitlines [17], a technique we borrow. This is illustrated in Figure 3-15.

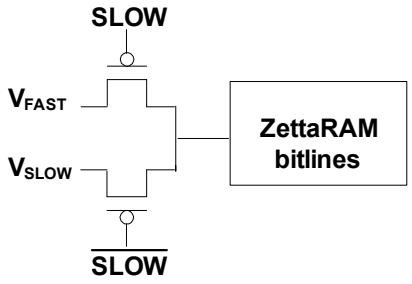


Figure 3-15. Selecting between two voltages to drive the bitlines.

The analytical model $C_{BL} \cdot V_{DD} \cdot (\Delta V_{BL})$ is derived by integrating power across the voltage supply ($V_{DD} \times I$), which yields the overall energy consumed, as opposed to integrating power across only the bitline capacitor ($V_{BL} \times I$). The analytical model was compared against SPICE simulations, and they match exactly.

3.4.3 Cycle-level Simulator

Our memory simulator is integrated with a custom detailed cycle-level processor simulator. The SimpleScalar ISA (PISA) [7] and compiler (gcc-based) are used. The processor configuration is given in Table 3-5. The cache and bus configurations are based on the Pentium® 4 processor [22]. The L1 instruction and data caches each allow up to 32 outstanding misses. The L2 cache allows up to 8 outstanding fetch requests at a time. Increasing the number of L2 MSHRs beyond 8 provided only minor performance benefits.

The maximum number of outstanding L2 writeback requests is only limited by the buffering in the memory controller.

Table 3-5. Processor configuration.

4-issue OOO superscalar, 7-stage pipeline	
Frequency	1 GHz
Reorder buffer	128 entries
Issue queue, LSQ	64 entries
Branch predictor	gshare, 2^{16} entries
Functional units	4, universal
Bus	400 MHz 64-bit
L1 I-cache	8 KB, 4-way, 64 B line size
L1 D-cache	8 KB, 4-way, 64 B line size
L2 cache (unified)	256 KB, 8-way, 128 B line, writeback
Hit latency	L1: 2 ns , L2: 10 ns
MSHRs	L1: 32 , L2: 8

The memory round trip latency for servicing L2 cache miss requests is given in Table 3-6, for various scenarios in the context of the open page policy. The bus latency to fetch a 128-byte L2 cache line from DRAM on a 64-bit wide bus operating at 400 MHz can be computed to be 40 ns ($\frac{128\,bytes}{64\text{ bits}} \times \frac{1}{400\text{ MHz}}$).

Table 3-6. DRAM round trip time for a L2 miss request for open page policy (t_{BUS} = 40 ns, t_{CAS} = 16 ns, t_{RAS} = 30 ns, t_{PRE} = 9 ns).

Scenario	Memory Round Trip Time
Requested page is open.	$t_{BUS} + t_{CAS}$ = 56 ns
Requested page is closed, no page currently open.	$t_{BUS} + t_{RAS} + t_{CAS}$ = 86 ns
Requested page is closed, another page is currently open.	$t_{BUS} + t_{PRE} + t_{RAS} + t_{CAS}$ = 95 ns

3.4.4 Benchmarks

We use eight different integer benchmarks from the SPEC2K benchmark suite with reference inputs. We used SimPoint to determine the appropriate starting simulation point for each benchmark [59]. 100 million instructions are then simulated from this simulation point. The SimPoints chosen for each benchmark are shown in Table 3-7. Table 3-7 also shows the

rates of L1 and L2 cache misses (per 1000 instructions) and L2 writebacks (per 1000 instructions) to main memory for each benchmark.

Table 3-7. SPEC2K benchmarks.

	SimPoint (billions of instr.)	L1 misses*	L2 misses*	writebacks*	writebacks that close page*
bzip	1	84.8	13.3	4.6	2.8
gap	209.5	87.8	4.2	1.8	1.2
gcc	11	98.8	9.6	3.13	2.4
gzip	48.7	97.0	4.7	1.91	1.5
mcf	31.7	208.6	80.3	31.84	23.8
parser	1.7	58.9	5.4	2.12	1.5
twolf	3.2	110.5	22.8	7.61	4.9
vortex	5.8	81.2	7.5	2.9	2.4

* per 1000 instructions

3.5 Baseline DRAM Energy and Performance

Figure 3-16 shows (a) bitline energy consumption and (b) execution times, for DRAM operating at 1.25 V. The pending request queue size for each bank is fixed at 4 entries. Memory access reordering is used in the baseline unless otherwise indicated. Since 1.25 V is the lowest reliable write voltage for DRAM, we use this system as our baseline and all ZettaRAM performance and energy measurements are normalized with respect to this baseline.

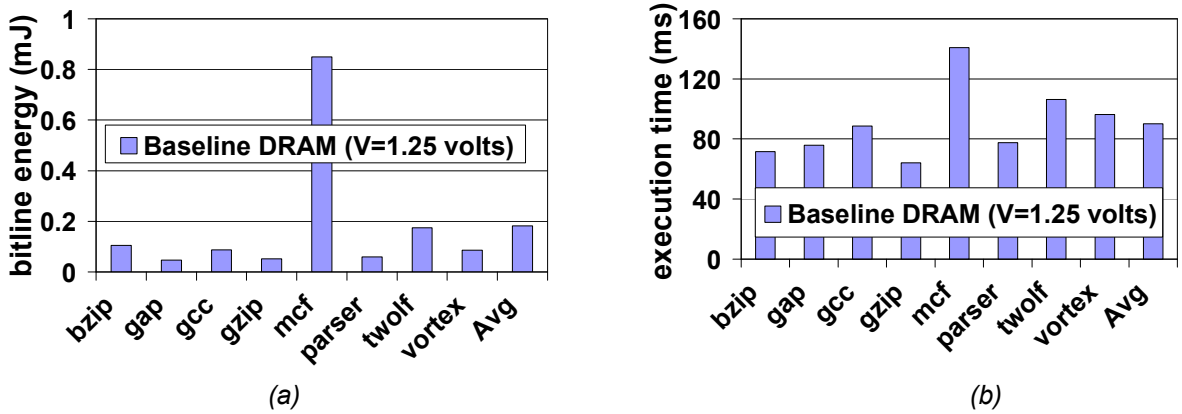


Figure 3-16. (a) Bitline energy consumption and (b) execution times, for DRAM operating at 1.25 V.

The next two sections describe in depth the two power scaling opportunities enabled by the charge-voltage decoupling characteristic of the molecular capacitor in ZettaRAM. Section 3.6 describes how molecular attributes can be precisely tuned to lower the ZettaRAM write voltage from one generation to the next. Section 3.7 describes how the molecular capacitor's speed/energy tradeoff can be architecturally managed by dynamically modulating write voltage for different memory requests, to lower the write voltage even further with little system slowdown.

3.6 Exploiting Molecular Engineering for Long-Term Power Scalability

Synthetic chemists can precisely tune key properties of the molecules through the choice of molecular "groups" and "linkers", such as the oxidation potential, electron transfer rate, and surface concentration (charge density). Many molecules have been synthesized and characterized by ZettaCore. Among these, we identify molecules that yield ZettaRAMs comparable to or better than DRAM in all respects – density (as determined by *cell area*), performance (as determined by *write latency*), and power (as determined by *voltage*).

To compare these ZettaRAMs with DRAM, we fix two of the variables, *cell area* and *write latency*, in order to focus on the third variable, *voltage*. Since cell area is fixed, we consider only those molecules with charge density greater than or equal to that of DRAM, so that the minimum amount of charge for reliable sensing is available. For fast writes, we target the same write latency as DRAM. Targeting faster intrinsic molecular speeds is of no use because, as mentioned before, the conventional peripheral circuitry used to access the molecular capacitor limits write latency anyway.

Having pinned down the cell area and write latency, voltage is the only unknown variable. In this situation, voltage is controlled by three key molecular attributes – oxidation potential, electron transfer rate, and surface concentration. We now give insight as to how these three molecular attributes affect the write voltage.

(i) *Oxidation potential (V_{ox}):* Write latency is determined by the current, which in turn is exponentially dependent on the difference between the applied voltage and the oxidation potential, as seen in EQ 3-5 of Section 3.2.2. Since write latency, hence current, are fixed, the difference $V-V_{ox}$ must be fixed. Thus, a decrease in oxidation potential results in an equal decrease in the write voltage.

(ii) *Electron transfer rate constant (k^0):* The current also depends on the electron transfer rate constant. A higher rate constant implies that the same current can be generated at a lower write voltage. Therefore, an increase in rate constant results in a decrease in write voltage. However, the relationship is non-linear because the rate constant is a coefficient of the Butler-Volmer exponential term whereas voltage is in the exponent.

(iii) *Surface concentration:* Since cell area is fixed, a higher surface concentration yields more molecules in the cell. However, the molecular capacitor needs to be charged only to the minimum charge that is required by the sense amplifiers for reliable sensing, i.e., the target number of charged molecules is fixed. Therefore, a higher surface concentration implies a correspondingly smaller fraction of the total molecules needs to be charged. Due to the nature of the Butler-Volmer equation, a smaller fraction can be charged faster than a larger fraction, even if the absolute number of charged molecules is the same in both cases (analogous to radioactive half-life principle – the fraction is what matters). Since we want to fix write latency, we can now offset the higher speed of charging a

smaller fraction by lowering the write voltage (slowing it back down to the target write latency). To sum up, we can exploit a higher surface concentration of molecules to lower the write voltage, thus saving energy. However, as with rate constant, the relationship is non-linear because concentration is a coefficient of the Butler-Volmer exponential term whereas voltage is in the exponent. In our analyses, Q_0 refers to the total number of molecules present in the molecular capacitor.

The relationship between write voltage and the three molecular parameters – V_{ox}, k^0, and Q_0 – is expressed indirectly by EQ 3-11 (derived in Appendix B), which shows the intrinsic molecule write latency t_{mol_write} as a function of the molecular parameters V_{ox}, k^0, and Q_0. We fix t_{mol_write} and then numerically solve for write voltage, given parameters V_{ox}, k^0, and Q_0 for a particular molecule.

$$t_{mol\_write} = \frac{1}{(k_O + k_R)} \cdot \ln\left\{ \frac{k_O}{k_O - (k_O + k_R) \cdot \left(Q_f / Q_0 \right)} \right\} \qquad \text{(EQ 3-11)}$$

$$k_O = k^0 \cdot e^{(1-\alpha)\left(\frac{F}{RT}\right)(V - V_{ox})} \qquad k_R = k^0 \cdot e^{-\alpha\left(\frac{F}{RT}\right)(V - V_{ox})}$$

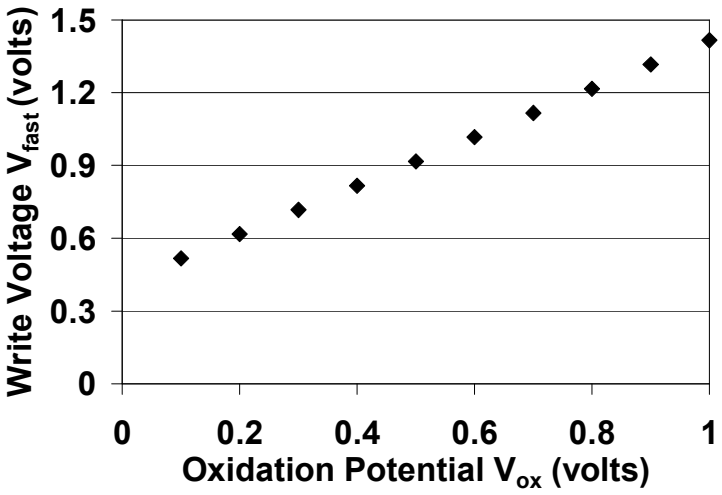
Parameters above are as follows: k_O = oxidation rate constant, k_R = reduction rate constant, Q_f = final charge in the molecular capacitor determined by the minimum charge required by the sense amplifiers for reliable sensing, Q_0 = total molecules in the molecular capacitor, equal to the surface concentration of the molecules multiplied by the fixed cell

area, k^0 = standard rate constant, α = transfer coefficient, F = Faraday constant, R = gas constant, T = temperature, V = applied voltage, V_{ox} = oxidation potential.

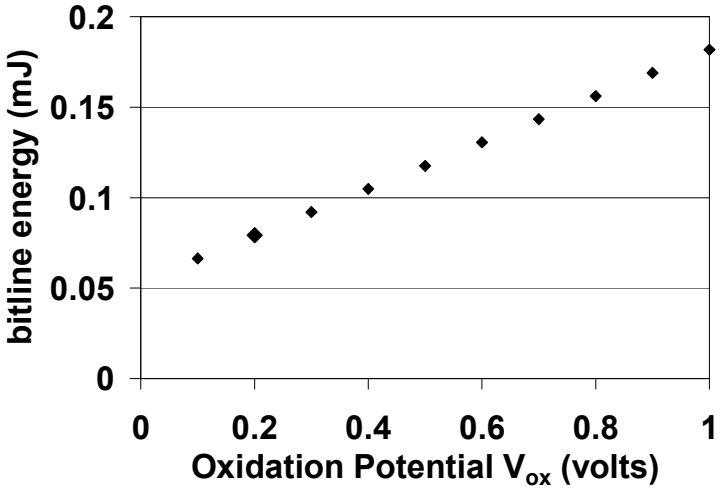
In Section 3.6.1, we study the sensitivity of write voltage and energy to changes in each of the three molecular attributes – V_{ox}, k^0, and Q_0. In Section 3.6.2, we evaluate the write voltage (for uniformly fast writes) and energy of numerous molecules based on their attributes obtained from the literature and identify molecules that yield ZettaRAMs with lower write voltages and energy consumption than DRAM.

3.6.1 Sensitivity of Voltage and Energy to Molecular Attributes – V_{ox}, k^0, and Q_0

First, we study the effect of increasing the oxidation potential on write voltage and bitline energy. In order to study the effect of oxidation potential in isolation, we fix surface concentration and rate constant to be the same as that of the first porphyrin molecule used in our SPICE experiments (28×10^{-11} moles/cm^2 and 7.5×10^4 s^{-1}, respectively). Figure 3-17 shows the effect of increasing V_{ox} on (a) write voltage and (b) bitline energy. We observe that write voltage changes by the same magnitude as V_{ox}, for example, decreasing V_{ox} by 0.9 V results in a 0.9 V decrease in write voltage. Thus, the write voltage is highly sensitive to changes in oxidation potential. Interestingly, the relationship between oxidation potential and bitline energy is also linear, as shown in Figure 3-17(b), even though bitline energy is proportional to both the write voltage V_{BL} and the voltage swing ΔV_{BL} ($E_{BL} = C_{BL} \cdot V_{BL} \cdot \Delta V_{BL}$). This is because the write voltage changes by the same amount as the oxidation potential, and, as a result, ΔV_{BL} remains constant. Nonetheless, in Figure 3-17, a 10X decrease in oxidation potential, which corresponds to a 64% decrease in write voltage, reduces energy consumption by 64%.



(a)

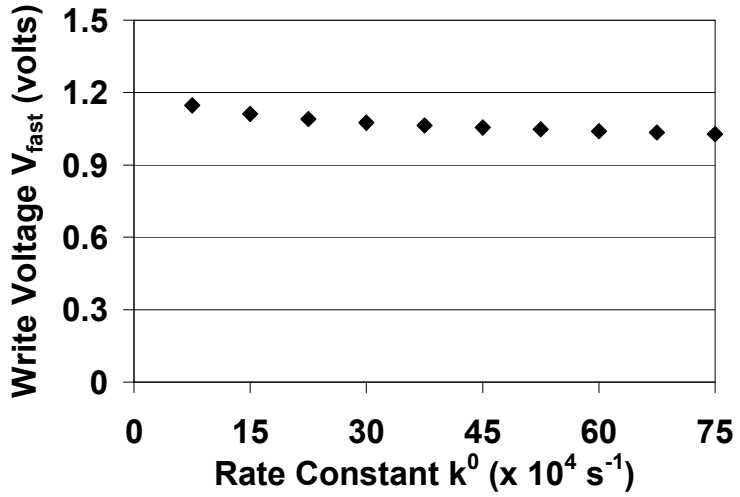


(b)

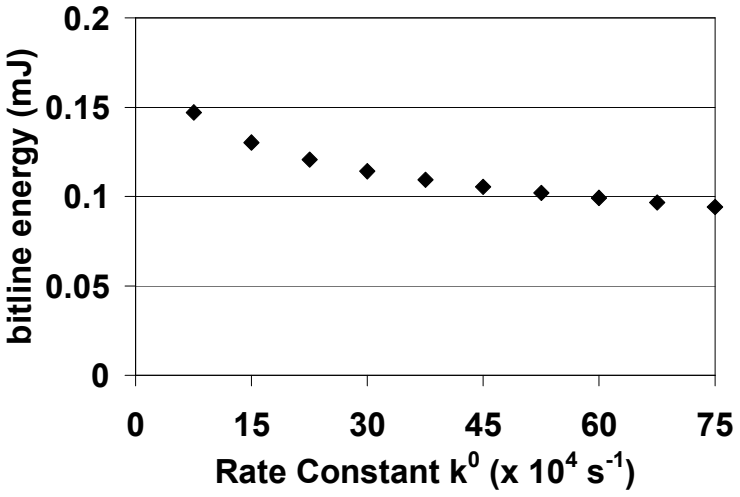
Figure 3-17. Effect of oxidation potential V_{ox} on (a) write voltage and (b) bitline energy.

Next, we study the effect of rate constant k^0 on the write voltage and bitline energy. Here, we fix surface concentration and oxidation potential to be the same as that of the first porphyrin molecule used in our SPICE experiments (28 x 10^{-11} moles/cm^2 and 0.73 V, respectively). Figure 3-18 shows the effect of increasing the rate constant k^0 on (a) write voltage and (b) bitline energy. We observe that a 10X increase in rate constant results in only a 10% decrease in write voltage. However, this corresponds to a more substantial decrease in

bitline energy of 36%, due to its quadratic dependence on write voltage. We conclude that write voltage and bitline energy are mildly sensitive to changes in rate constant.



(a)

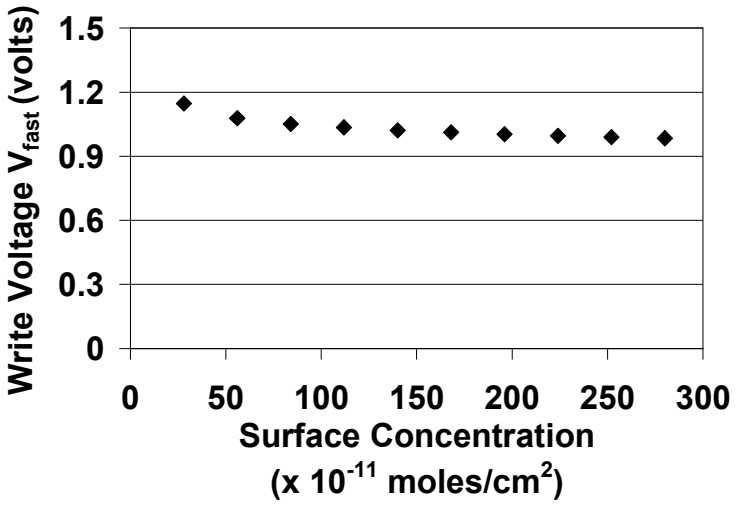


(b)

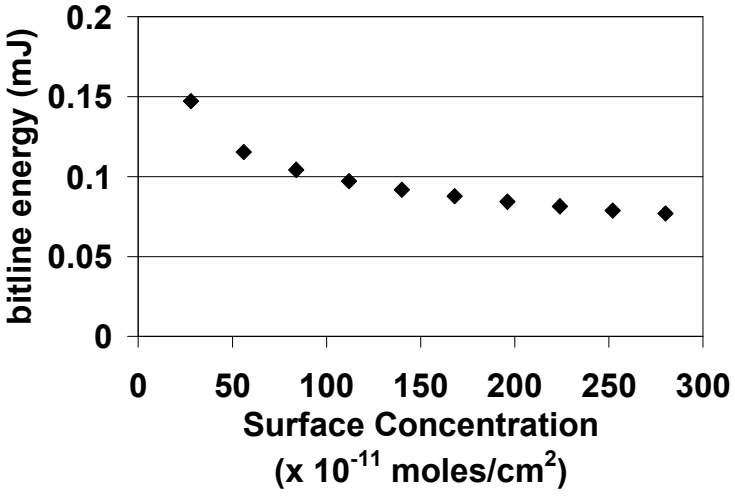
Figure 3-18. Effect of rate constant k^0 on (a) write voltage and (b) bitline energy.

Finally, we study the effect of surface concentration on the write voltage and bitline energy. Here, we fix rate constant and oxidation potential to be the same as that of the first porphyrin molecule used in our SPICE experiments (7.5 x 10^4 s^{-1} and 0.73 V, respectively).

Figure 3-19 shows the effect of increasing the surface concentration on (a) write voltage and (b) bitline energy. We observe that a 10X increase in surface concentration, i.e., a 10X increase in charge density, results in only a 14% decrease in write voltage, which corresponds to a more substantial (48%) decrease in bitline energy. We conclude that write voltage and bitline energy are mildly sensitive to changes in surface concentration.



(a)



(b)

Figure 3-19. Effect of surface concentration on (a) write voltage and (b) bitline energy.

From the above analysis, we observe that among the three molecular attributes, oxidation potential has the largest effect on write voltage. However, oxidation potential, surface concentration, and rate constant have similar effects on bitline energy. Bitline energy only depends linearly on the write voltage, when V_{ox} is varied. Consequently, a 64% decrease in write voltage results in a 64% decrease in energy consumption. On the other hand, bitline energy depends on the square of the write voltage, when surface concentration and rate constant are varied. As a result, a 10%-14% decrease in write voltage is magnified to a 36%-48% decrease in energy consumption.

3.6.2 Analysis of Molecules

The selected molecules and their attributes are shown in Table 3-8. Also shown are the fast and slow write voltages, calculated using the methodology derived earlier in Section 3.6. The fast voltage is used to determine energy with uniformly fast writes. Both the fast and slow voltgaes are sued to determine the energy with our hybrid write policy (dynamic voltage modulation), described in Section 3.7.

Table 3-8. Molecules with comparable or better charge density than DRAM.

Molecules	Surface Concentration ($\times 10^{-11}$ mol/cm^2)	Rate Constant k^0 ($\times 10^4$ s^{-1})	Oxidation Potential V_{ox} (V)	V_{fast} (V)	V_{slow} (V)
(1) Triple decker O2 2/3+	96	9.7	0.3	0.65	0.45
(2) Triple decker O2 3/4+	420	4.5	0.48	0.79	0.59
(3) TD-Tpd(TD-2/3+)	35	10	0.31	0.73	0.53
(4) Triple decker O2 1/2+	25	16	0.24	0.73	0.53
(5) TD-Tpd(TD-3/4+)	96	8.7	0.77	1.13	0.93
(6) TD-Phenylethynylphenyl Linker state 3	27	7	0.52	1	0.8
(7) TD-Phenylethynylphenyl Linker state 3	27	2.2	0.39	0.93	0.73
(8) TD-Tpd(TD-2/3+)	35	8.5	0.76	1.19	0.99
(9) PM-1	28	7.5	0.73	1.2	1
(10) TD-Tpd(TD-3/4+)	96	8.8	1.23	1.59	1.39
(11) ZnP-Tpd(ZnP-1/2+)	25	10.7	0.68	1.19	0.99
(12) Triple decker T1 3/4+	96	8	1.44	1.8	1.6
(13) ZnP-Tpd(ZnP-1/2+)	25	5.5	0.71	1.25	1.05
(14) Triple decker D1 3/4+	64	0.9	0.9	1.4	1.2
(15) Triple decker M1 3/4+	35	7.9	1.23	1.66	1.46
(16) Triple decker M2 2/3+	27	5.8	0.99	1.48	1.28
(17) Fc-ZnP-Tpd (ZnP-1/2+)	25	9.8	1.13	1.64	1.44
(18) TD-Phenylethynylphenyl Linker state 4	35	5.4	1.48	1.93	1.73
(19) Fc-ZnP-Tpd (ZnP-1/2+)	25	4.5	1.03	1.58	1.38
(20) TD-Phenylethynylphenyl Linker state 4	64	0.9	1.3	1.8	1.6
(21) Triple decker T1 2/3+	27	5.4	1.46	1.95	1.75
(22) Triple decker M1 2/3+	25	7.3	1.34	1.87	1.67
(23) Triple decker D1 2/3+	27	4.5	1.68	2.18	1.98

3.6.3 Energy Savings with Uniformly Fast Writes

Figure 3-20 shows ZettaRAM energy consumption normalized to that of DRAM, for each molecule type. All write operations are performed at a voltage V_{fast}. In other words, at this voltage, the peripheral circuitry, rather than intrinsic molecule speed, is the overall performance limiter. This voltage V_{fast} is different for each molecule and is given in Table 3-8. We observe that, among the 23 molecules chosen, 13 operate with higher energy consumption than DRAM. An unfavorable combination of oxidation potential, electron transfer rate, and surface concentration yields a higher write voltage, and consequently higher energy consumption. 10 molecules operate with lower energy consumption than DRAM.

This is because a favorable combination of molecule attributes yields a lower write voltage, resulting in significantly lower energy consumption than DRAM. Energy savings of up to 61% can be obtained by using a single, fixed write voltage. The porphyrin molecule used in the earlier SPICE simulations is highlighted via the arrow in Figure 3-20. Eight different molecules yield lower energy than it.

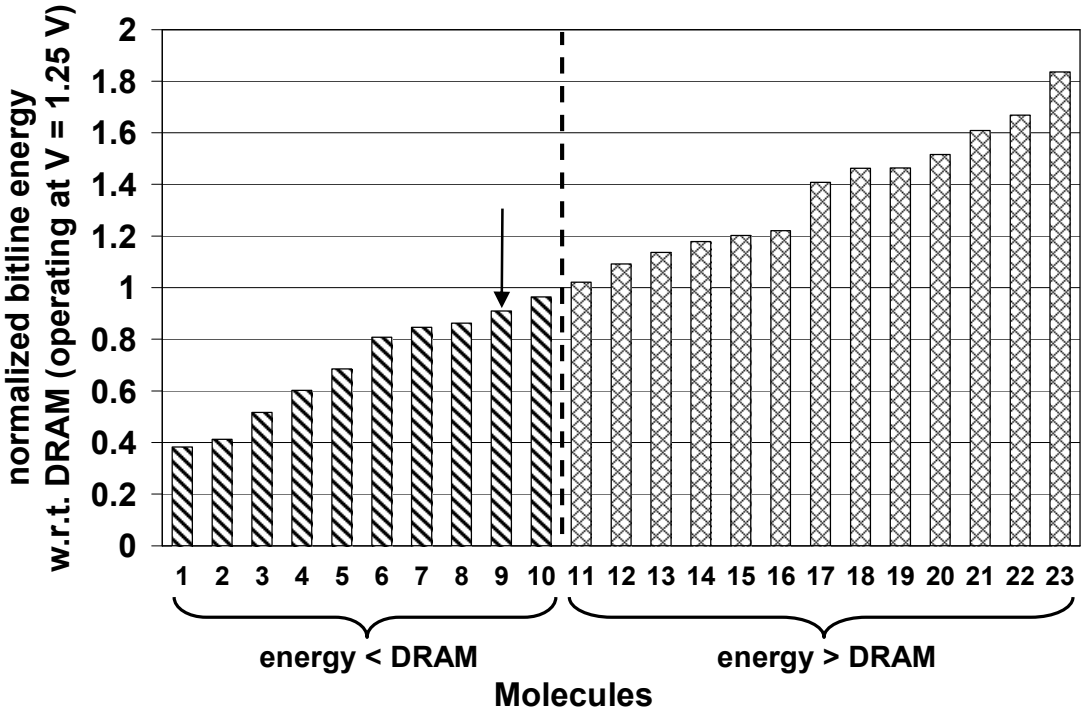


Figure 3-20. ZettaRAM with uniformly fast writes.

3.7 Intelligent Management of ZettaRAM

ZettaRAM write speed depends on the difference between applied voltage and oxidation potential (faster with larger differential). Therefore, the write voltage is padded with respect to the oxidation potential to achieve the same speed as DRAM and avoid any performance

degradation with respect to DRAM. Thus, there is some room to lower voltage even further by intelligently managing this performance-energy tradeoff.

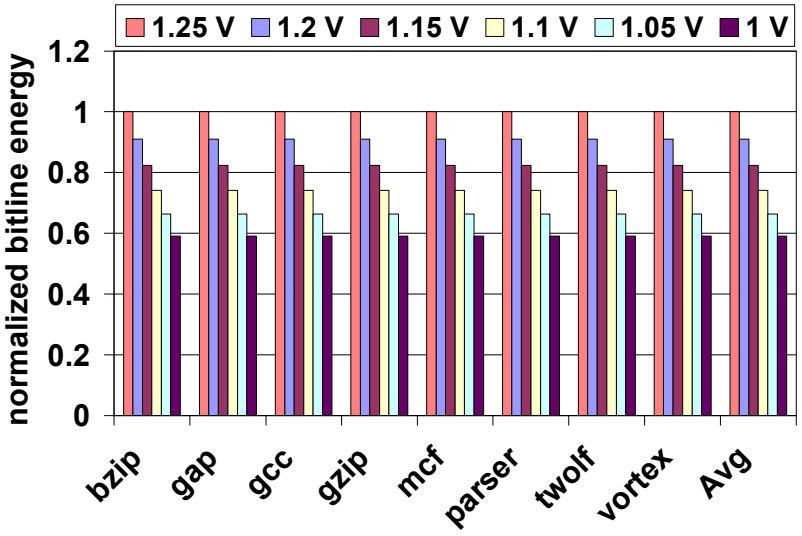
Specifically, we describe a novel hybrid write policy, which dynamically modulates the write voltage for different memory requests to maximize energy savings potential of ZettaRAM with little system slowdown. The proposed dynamic voltage modulation further extends ZettaRAM's power scaling advantage over DRAM.

First, we explain dynamic voltage modulation and demonstrate its energy benefits in the context of one particular molecule type (porphyrin molecule used in earlier SPICE simulations, V_{ox} = 0.73 V). Then, the energy benefits of dynamic voltage modulation are shown for all 23 molecules. The results presented in subsequent graphs in this section are averaged over the eight SPEC2K benchmarks, and normalized with respect to the baseline DRAM, unless stated otherwise.

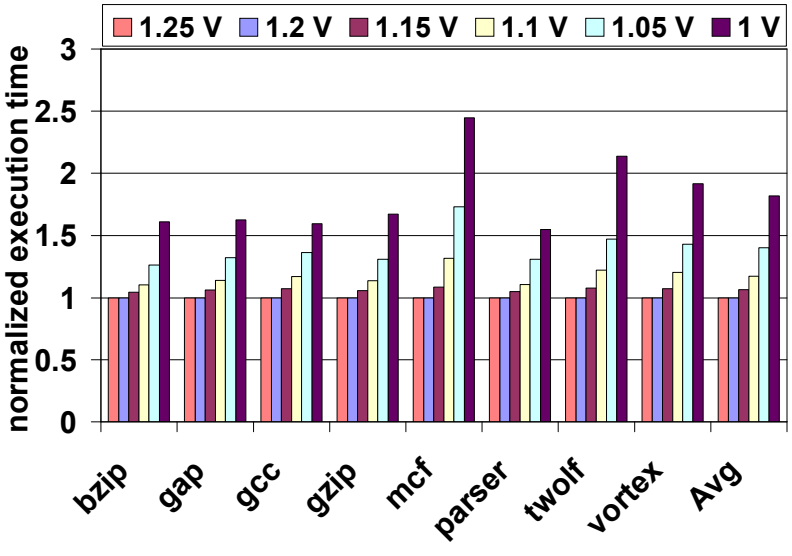
3.7.1 Tradeoff Between Bitline Energy and System Performance

We first quantify the tradeoff between system performance and bitline energy as the ZettaRAM write voltage is changed. Figure 3-21 shows (a) bitline energy consumption and (b) execution times, for ZettaRAM operating at fixed write voltages of 1.0 V through 1.25 V in 0.05 volt increments. At 1.25 V and 1.2 V, the execution times for ZettaRAM and the baseline DRAM are equal because the intrinsic speed of the molecules is fast enough above 1.2 V, such that the write latency is dictated by the peripheral circuitry. However, at lower voltages, overall write latency and thereby system performance is mainly determined by the intrinsic speed of the molecules.

From Figure 3-21(a), lowering the write voltage from 1.25 V to 1.0 V reduces bitline energy by 41%. However, as expected, execution time increases by 81%, as shown in Figure 3-21(b). This is because write latency increases exponentially with decreasing write voltage. Thus, writes done at 1.2 V favor performance, whereas writes done at 1.0 V favor energy savings.



(a) bitline energy w.r.t. DRAM



(b) execution time w.r.t. DRAM

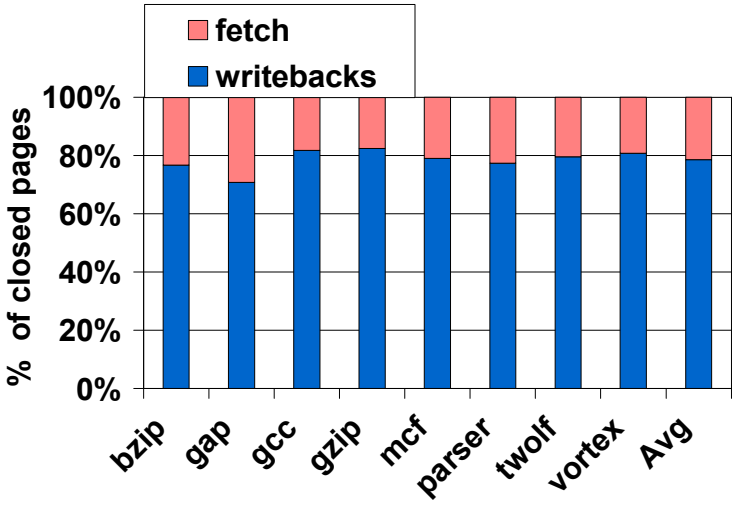
Figure 3-21. ZettaRAM with various write voltages.

Although the voltage drop from 1.25 V to 1 V may seem modest, in the 1V regime, it is significant. Moreover, the potential energy savings are much higher (41%) due to a quadratic dependence on voltage, as described in Section 3.4.2.

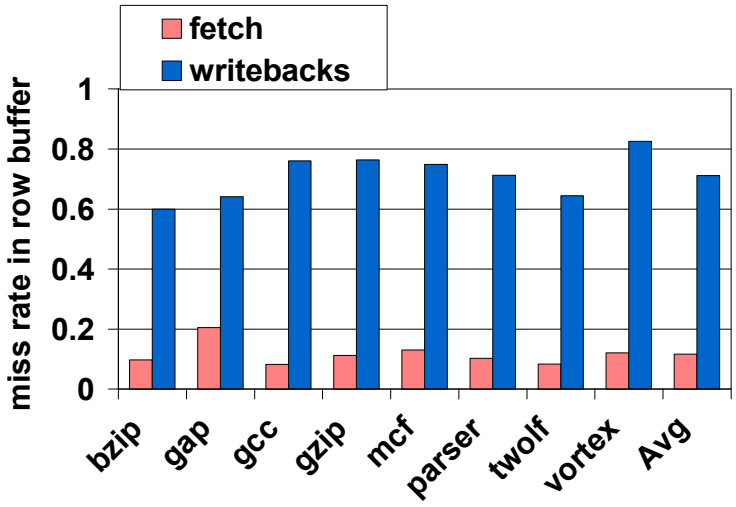
3.7.2 Hybrid Write Policy

All L2 requests to memory are serviced from the row buffer. If the requested page is not in the row buffer, the current page is closed (written back to array) and the requested page is opened (read from array). Closing a page corresponds to a write operation, and can be done with either a fast write or slow write, favoring either performance or energy, respectively. Opening a page corresponds to a read operation, and is always performed at the OCP. Thus, our hybrid write policy applies to closing a page.

Two types of memory requests are initiated by the L2 cache, fetch block and writeback block. The graph in Figure 3-22(a) shows that 71-82% (79% on average) of all closed pages are closed because of writebacks that miss in the row buffer. Only 18-29% (21% on average) of all closed pages are due to fetches that miss in the row buffer. Writebacks exhibit significantly lower locality than fetches, with respect to the row buffer. Figure 3-22(b) shows that fetches miss only 10-20% of the time, whereas writebacks miss 60-82% of the time (71% on average). Because writebacks cause most of the closed pages, they constitute most of the energy savings potential. Therefore, we can tap most of the energy savings potential of ZettaRAM by focusing on writebacks. Moreover, writebacks are not timing critical and do not stall the processor directly, thereby offering scheduling flexibility. In contrast, fetch requests are timing critical and typically stall the processor because of the high memory latency. The processor scheduling window is not large enough to tolerate the high memory latency, even with out-of-order execution.



(a)



(b)

Figure 3-22. (a) Percentage of closed pages that are closed due to writebacks vs. fetches. (b) Row buffer miss rates for writebacks and fetches.

Accordingly, we propose a hybrid write policy, where fetches and writebacks are handled differently. If a critical fetch request causes a row buffer miss, the current page is closed using a fast write (high energy), whereas, if a non-critical writeback request causes a row buffer miss, the current page is closed using a slow write (low energy). Applying slow writes to writebacks taps most of the energy savings potential, and applying fast writes to fetches eliminates most of the performance degradation. The pie chart in Figure 3-23

illustrates the hybrid write policy. Fast and slow writes are done at 1.2 V and 1.0 V, respectively, for this particular molecule type.

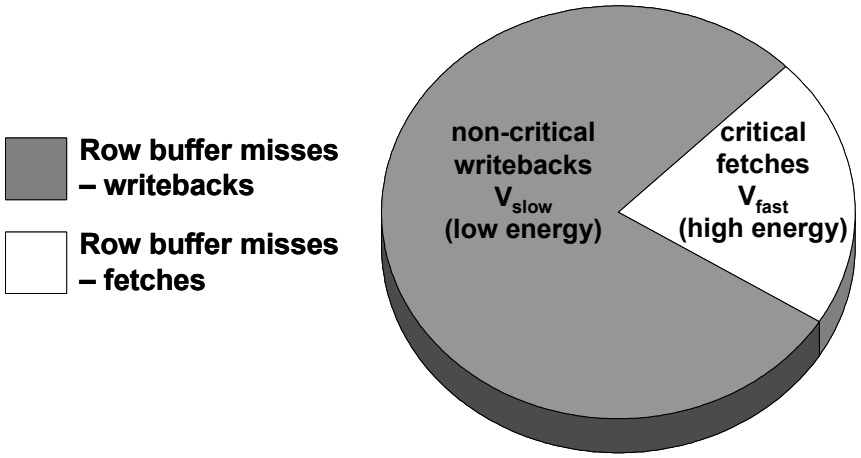
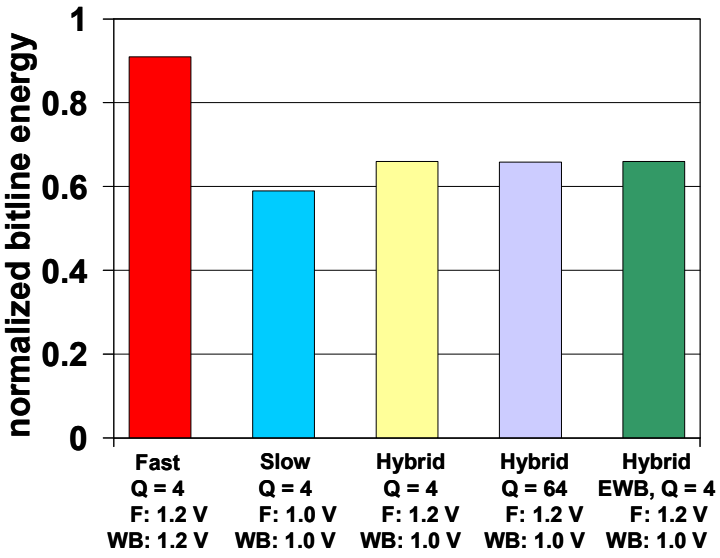


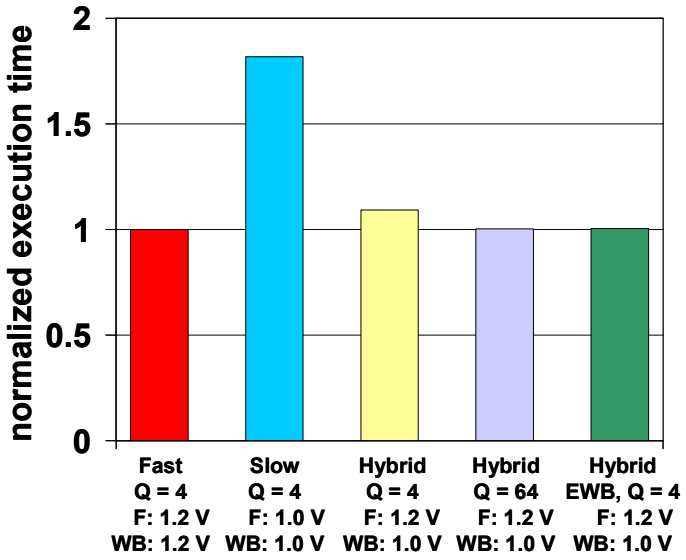
Figure 3-23. Hybrid write policy.

Figure 3-24 shows (a) bitline energy consumption and (b) execution time, for ZettaRAM with different write policies. The first bar ("Fast") correponds to all writes done at 1.2 V. Similarly the second bar ("Slow") correponds to all writes done at 1.0 V. These two bars are reproduced in Figure 3-24 from Figure 3-21 to facilitate comparisons with the hybrid write policies. The third bar in Figure 3-24(a) shows that the hybrid write policy, as predicted, taps most of the energy savings potential. Hybrid writes achieve 34% energy savings, out of a possible 41% energy savings with uniformly slow writes (second bar in Figure 3-24(a)). Moreover, hybrid writes increase execution times by only 10% (third bar in Figure 3-24(b)), as opposed to 81% with uniformly slow writes (second bar in Figure 3-24(b)). The performance degradation has been reduced from an average slowdown of 81% with uniformly slow writes to only 10% with hybrid fast/slow writes. Thus, the hybrid write

policy couples the performance of uniformly fast writes with the energy savings of uniformly slow writes.



(a) *bitline energy w.r.t. DRAM*



(b) *execution time w.r.t. DRAM*

Figure 3-24. ZettaRAM with various write policies.

3.7.3 Eliminating Residual Slowdown

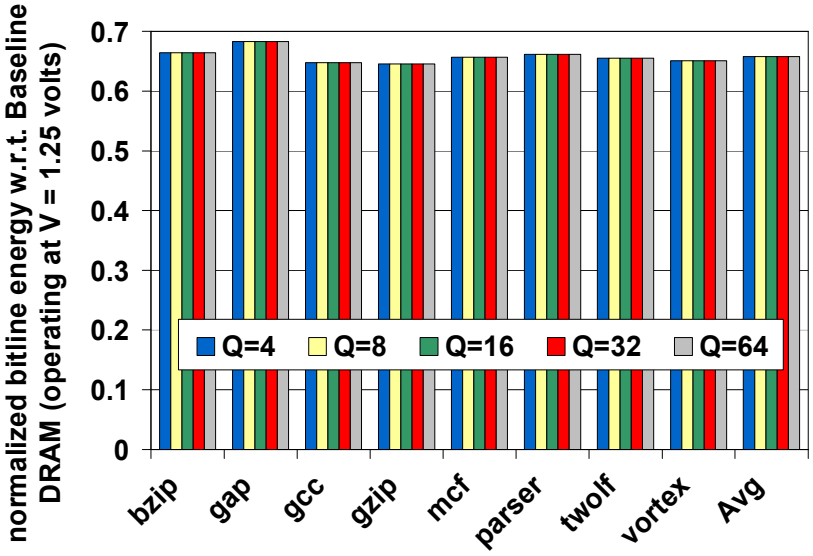
The residual 10% performance degradation still exists because, although deferred writebacks do not directly stall the processor, they may fill up the request queues in the memory controller, eventually stalling critical fetch requests.

3.7.3.1 Hybrid Write Policy Coupled with Large Queues and Access Reordering

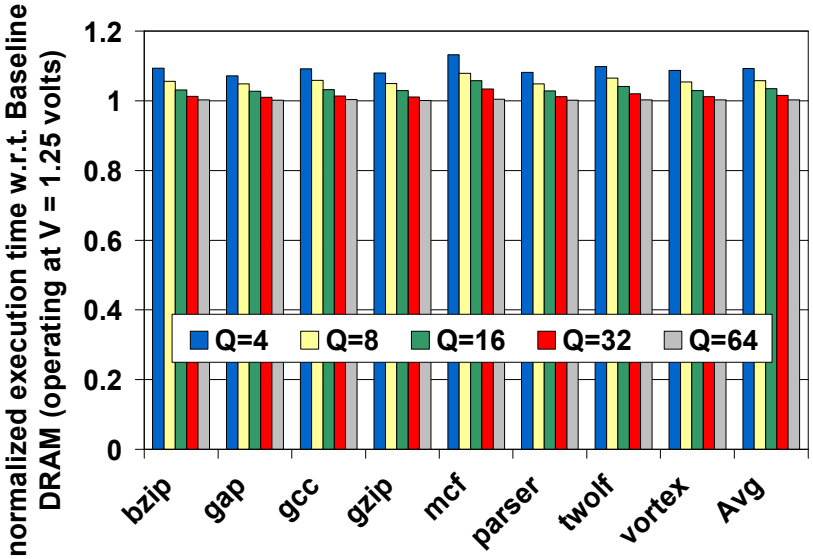
A natural approach to avoid any queue-full stalls, the reason behind the residual slowdown, is to increase the request queue size. The default policy of memory access reordering (fetch requests bypass queued writeback requests) is used. Figure 3-25 shows (a) bitline energy consumption and (b) execution times, for ZettaRAM using the hybrid write policy and queue sizes of 4, 8, 16, 32, and 64 entries. As expected, the residual slowdown is reduced as the queue size is increased. The residual performance degradation is reduced to less than 1% when the request queues are increased from 4 to 64 entries, as shown in Figure 3-25(b) for all benchmarks. The average slowdown is less than 1%, as shown in the fourth bar of Figure 3-24(b). Therefore, a ZettaRAM memory system employing slow writes for pages closed by non-critical requests (writebacks) achieves 34% bitline energy savings with less than 1% performance degradation, if the pending request queue contains 64 entries and memory access reordering is used.

However, enlarging the queues increases system cost. Each entry contains an entire cache block, thus four 64-entry queues costs 31 KB more than four 4-entry queues. Enlarging the queues also increases complexity. Fetch requests that circumvent queued writeback requests must first search the queue for possible address matches to avoid any read-after-write hazards. The complexity of searching the entire queue for these conflicts increases with queue size. An option to avoid this complexity is to disable memory access reordering. In

Section 3.7.4, we measure the performance impact of not reordering memory accesses. But first, we explore eager writebacks as an alternative to large queues in the next subsection.



(a) *normalized bitline energy*



(b) *normalized execution time*

Figure 3-25. ZettaRAM with hybrid write policy and various request queue sizes.

3.7.3.2 Hybrid Write Policy Coupled with L2 Cache Eager Writeback

To avoid the cost and complexity of larger queues in the memory controller, we propose employing the eager writeback policy in the L2 cache [41] to evenly spread out writeback

requests from the L2 cache, reducing the frequency of queue-full stalls. In the eager writeback policy, a writeback is issued as soon as a dirty block becomes the LRU block in its set, instead of waiting for the block to be evicted.

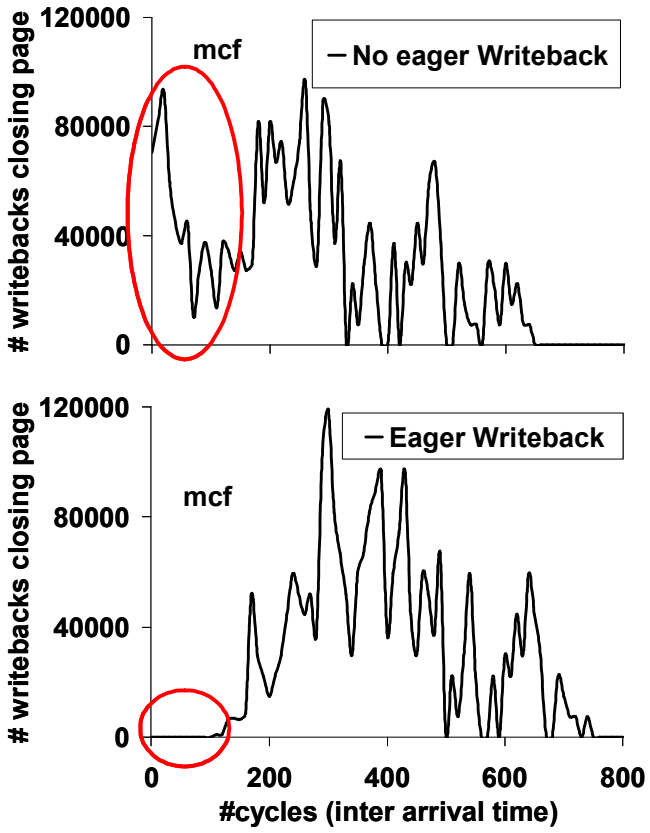


Figure 3-26. Arrival time (in cycles) of next request after a writeback request starts closing a page.

Figure 3-26 shows the arrival time (in cycles) of the next request to a bank after a writeback request starts closing a page, for the hybrid write policy with 4 queue entries (top graph) and the hybrid write policy with 4 queue entries coupled with the eager writeback policy in the L2 cache (bottom graph). The measurements are for *mcf* (other benchmarks show similar patterns). We can see that the L2 eager writeback policy de-clusters the writeback requests to memory. For example, once a writeback request starts closing a page,

the next request does not arrive for at least 100 cycles. In comparison, without eager writeback, about a quarter of all next requests arrive between 0 and 100 cycles.

Thus, with eager writebacks, we can probably do well with a small queue, in spite of delaying writebacks in the memory controller. *Effectively, issuing the writeback early from the L2 cache compensates for delaying it in the memory controller.*

For results in the remainder of this section, the L2 cache implements eager writeback for all configurations including the baseline DRAM system, to which energy and performance of the hybrid write policy coupled with eager writeback are normalized. We first explore the impact of eager writebacks on the baseline system. Figure 3-27 shows that L2 eager writebacks improve performance of the baseline DRAM system by only a negligible amount for these benchmarks, 0.6-1.3%, and bitline energy consumption is unaffected.

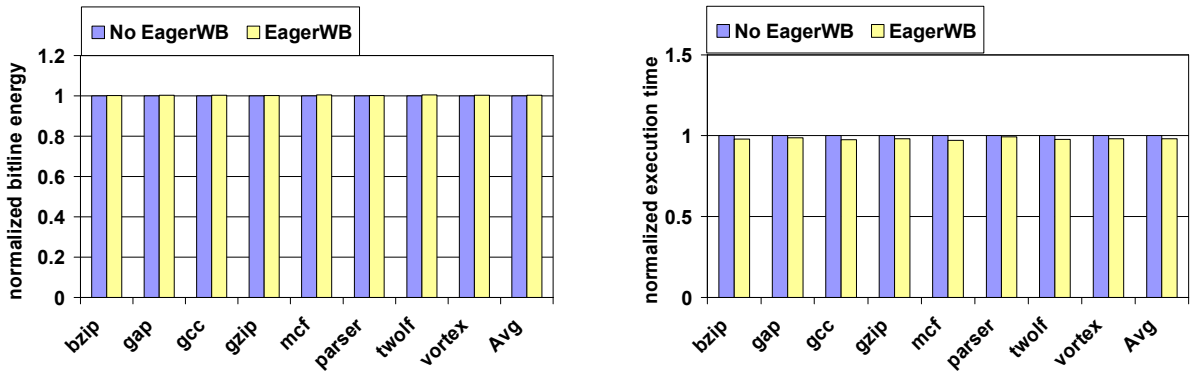
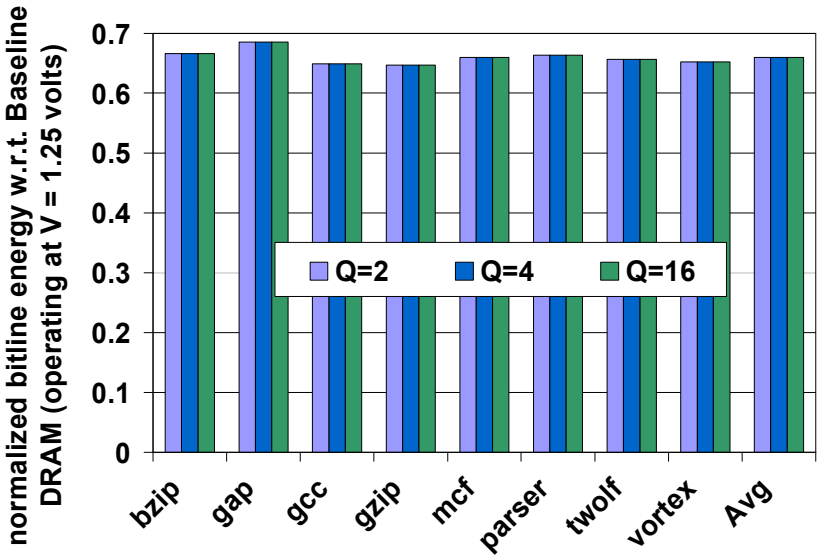


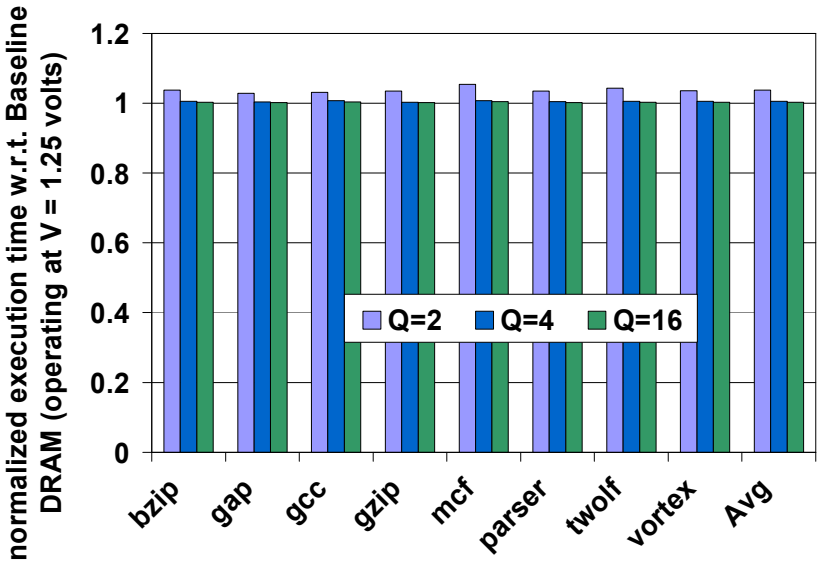
Figure 3-27. Effect of L2 cache eager writebacks on (a) Bitline energy consumption and (b) execution times, for the baseline DRAM system.

In contrast, eager writebacks are highly beneficial for ZettaRAM with hybrid fast/slow writes, as predicted. Figure 3-24(b) (final bar) and Figure 3-28(b) show that eager writebacks are effective for eliminating performance degradation, otherwise caused by delayed writebacks in the memory controller. A ZettaRAM memory system employing the hybrid write policy in conjunction with the L2 cache eager writeback policy yields 34% energy

savings with less than 1% performance loss. This is achieved without enlarging the request queue size with respect to the baseline system (4 entries). This is significant in terms of keeping the cost and complexity of the memory controller the same as the baseline. Results are also presented for queue sizes of 2 and 16 in Figure 3-28. The queue size of 2 entries degrades performance by 4%, whereas 16 entries performs only slightly better than 4 entries.



(a) *normalized bitline energy*



(b) *normalized execution time*

Figure 3-28. ZettaRAM (hybrid write policy, various queue sizes) in conjunction with L2 eager writebacks.

Thus, eager writebacks are remarkably synergistic with ZettaRAM for eliminating performance degradation otherwise caused by delayed writebacks.

The eager writeback policy may increase the number of L2 writeback requests, by occasionally writing back a dirty block before the final store has occurred to the block before eviction. Fortunately, it is rare. Figure 3-29 shows the percentage increase in the number of L2 writeback requests and L2 writeback requests that close a page in memory, because of premature writebacks. On average, there is only a 0.84% increase in the number of writeback requests and a 0.16% increase in the number of writeback requests that close a page. The extra writebacks are accounted for in the primary results in Figure 3-24 and Figure 3-28.

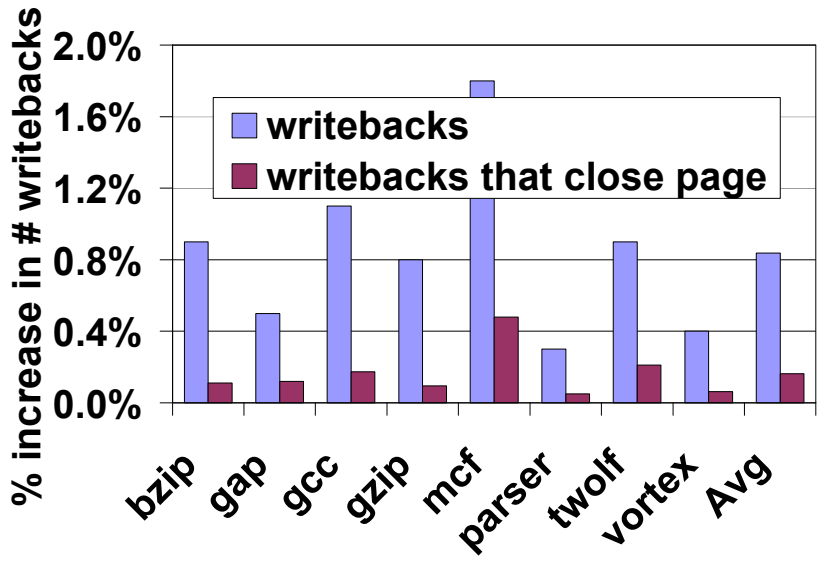


Figure 3-29. Percentage increase in L2 writebacks.

3.7.4 Effect of Memory Access Reordering

From subsection 3.7.3, we arrive at two competing alternatives for achieving high performance and low energy with a ZettaRAM memory system: (1) hybrid write policy with a large queue vs. (2) hybrid write policy with a small queue coupled with L2 eager writebacks. In all previous experiments, memory access reordering was used by default. We

now measure the impact of disabling memory access reordering for these two competing alternatives. The increase in execution time when reordering is disabled is shown in Figure 3-30 ("ZettaRAM, large Q, no reorder" and "ZettaRAM, small Q, eager WB, no reorder"). As expected, memory access reordering is performance-critical for the ZettaRAM that achieves latency tolerance via the large queue (execution time increases by 5-12% without reordering), but not performance-critical for the ZettaRAM that achieves latency tolerance via eager writebacks (execution time increases by less than 1.2% without reordering).

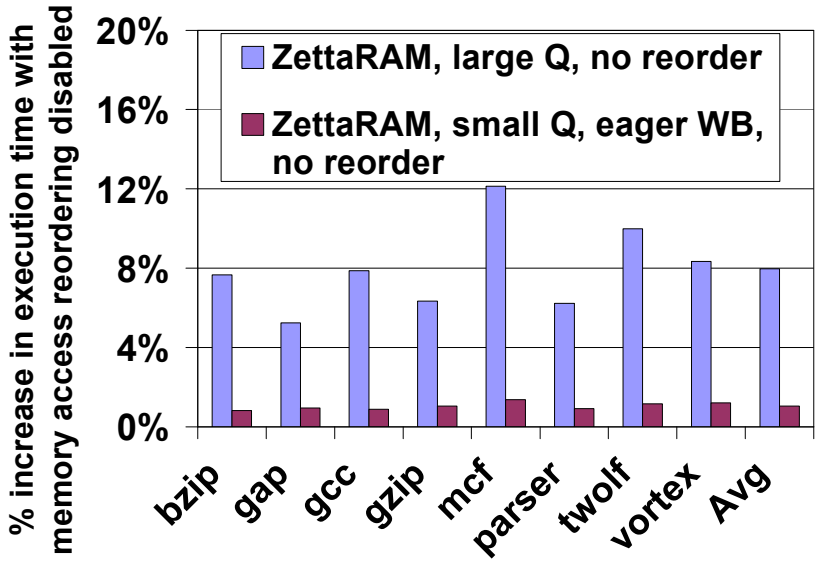


Figure 3-30. Impact of disabling memory access reordering for two ZettaRAM configurations.

To sum up, hybrid fast/slow writes and L2 eager writebacks lower the ZettaRAM write voltage even further with little performance loss and the least complexity in the memory controller. Managed ZettaRAM provides the same performance as DRAM and in addition can operate with much lower power.

3.7.5 Energy Savings with Hybrid Write Policy for Different Molecules

Now, we apply our hybrid write policy (coupled with L2-cache eager writebacks) in the context of each molecule. Recall, in the hybrid write policy, fast writes are applied to critical

fetch requests and slow writes are applied to non-critical writeback requests. Fast and slow writes are performed at voltages V_{fast} and V_{slow}, respectively. The voltages V_{fast} and V_{slow} for each molecule are given in Table 3-8. Figure 3-31 shows ZettaRAM energy consumption normalized to that of DRAM, for each molecule type. As expected, we get further energy savings using our architectural technique. The best molecule now yields energy savings of 77% (previously 61%). As before, the original molecule is highlighted with an arrow. On average, the hybrid write policy extends ZettaRAM's energy advantage by an additional 25-40%. Interestingly, 6 of the molecules that yielded higher energy than DRAM with uniformly fast writes, now yield lower energy when the hybrid write policy is employed.

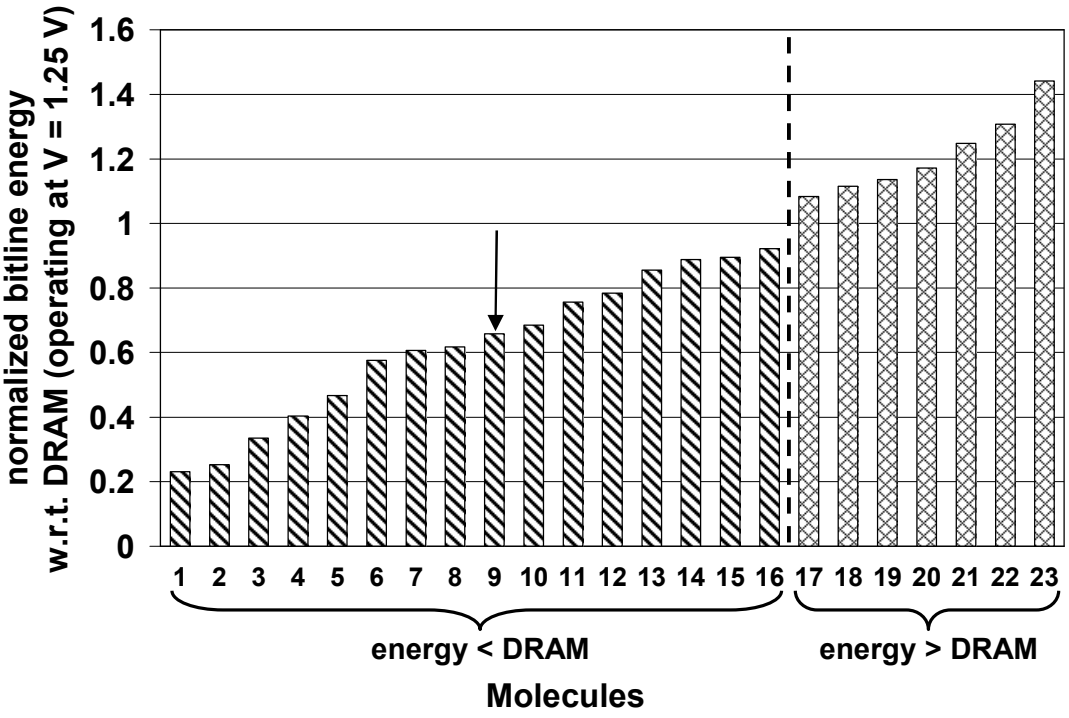


Figure 3-31. ZettaRAM with hybrid write policy.

3.8 ZettaRAM Summary

To summarize this chapter, ZettaRAM overcomes voltage scaling limits of DRAM via charge-voltage decoupling. Charge-voltage decoupling provides two power-scaling opportunities.

(1) Static Voltage Scaling: Precise engineering of molecular attributes provides an inexpensive path to voltage scaling from one memory generation to the next.

(2) Dynamic Voltage Scaling: The speed-voltage dependence requires the applied voltage to be padded with respect to the oxidation potential, in order to be as fast as DRAM. The padded applied voltage can be further lowered by architectural management of the speed-voltage dependence.

Chapter 4

Related Work

4.1 DRAM Scaling Trends

According to the DRAM density scaling trends published by ITRS, DRAM densities quadruple with each new generation [4].

Itoh et al. [31], Mandelman et al. [45], and Teng et al. [60] have investigated DRAM scaling trends and concluded that DRAM cell capacitance will remain steady at around 30 fF to 40 fF and will not scale with technology [45][60]. The charge in the capacitor must be large enough to generate a bitline voltage change that can be reliably sensed, including compensating for various noise sources (radiation, leakage current, and electrical imbalances between pairs of bitlines). While noise induced by radiation decreases with each generation, leakage current and electrical imbalances remain nearly the same from one generation to the next. Therefore, the required charge has not reduced much with each new generation [31][45]. The above factors coupled with the linear relationship between charge and voltage in DRAMs has resulted in only a small reduction in both charge and voltage with each generation.

4.2 Memory Energy Consumption

Itoh et al. [28][29] quantify energy consumption in DRAMs and conclude that bitline energy consumption is the main component of the total memory system energy consumption.

Viredaz et al. [65] characterize the ITSY prototype and provide detailed analysis of the power consumption of various components during different power modes. Their work provides valuable insights on the power distribution among various components in next-generation mobile devices. They show that DRAM consumes 25-50% of total system power while running representative applications like MPEG-1 and DECtalk on the ITSY. Future feature-rich applications will likely call for more memory capacity and increase memory system activity [18][27]. So, main memory will consume an increasing fraction of the energy budget. Therefore, techniques for reducing memory energy consumption will become increasingly important.

Mobile devices typically go through short bursts of activity followed by long periods of idle/sleep time [26][34][35]. Efficient energy management is paramount during both active and idle periods. There have been a lot of studies on energy management of main memories (DRAMs) during both standby and active operation.

4.3 Reducing DRAM Energy Consumption in Standby Mode

DRAM refresh is the largest component of power consumption when the device is in standby mode [34][35][68]. Therefore, there has been a lot of work done at the device level and circuit level to reduce DRAM refresh power.

Most of the techniques at the device level focus on decreasing the leakage current. Two example techniques include varying dopant concentrations [42][43] and controlling the trap densities and electric fields at the junction [49][62]. Lower leakage currents increase the retention times of individual DRAM cells, thereby reducing refresh power.

The rest of this section focuses on circuit level techniques to reduce refresh power.

4.3.1 Self-Tuning Refresh Period

Many circuit level techniques have been proposed to automatically tune the refresh period of a single DRAM chip. This approach has several advantages. First, automatically detecting the refresh period may obviate the need for off-line data retention testing. Second, the refresh period can be automatically and dynamically adjusted for temperature. Third, these techniques naturally exploit inter-die process variation, i.e., variation across different DRAM chips. However, these techniques do not exploit retention-time variations among cells within a single DRAM chip, so the shortest-retention cell among all cells still limits the safe refresh period.

Murotani [50] and Pelley et al. [55] propose modifications to DRAM refresh circuitry, to automatically tune a single refresh period based on temperature. Burgan [6] proposes monitoring the leakage of test cells representative of the entire memory, for automatically detecting a suitable refresh period for the DRAM as a whole. Test cells monitor leakage of both logic “1” and logic “0”.

Hsu et al. [10] and Cho et al. [11] propose monitoring the leakage of four test cells, one for each bank, thus automatically selecting a refresh period for each bank of the DRAM. The shortest-retention cell in a bank still limits the refresh period of the bank as a whole. That is, the granularity is only slightly finer than that of the whole DRAM, and it is not clear if there will be much separation among per-bank refresh periods.

Limited support for temperature compensated refresh (TCR) is now being deployed in low power and mobile DRAMs [8][64]. However, TCR does not exploit retenton-time variations among different memory cells, i.e., all cells are still treated uniformly with a single temperature dependent refresh period, determined by the worst cell.

4.3.2 Exploiting Retention-time Variations among Cells

Process variations cause different retention times among memory cells, rows (pages), or arbitrary blocks of cells within a single DRAM chip. Therefore, different cells retain information for longer or shorter periods of time. Several researchers have recognized the opportunity presented by retention-time variations, proposing custom DRAM support for refreshing different cells or blocks of cells at different refresh rates.

4.3.2.1 Characterization and Modeling of Retention-time Variations

Hamamoto et al. [19] characterize retention-time variations among individual cells, although no techniques to exploit retention-time variations are proposed. Hamamoto et al. [19] show that retention-time variations among individual cells in DRAM follow a normal distribution, and propose an analytical model that provides insight into the contributing factors that cause retention-time variations. Our retention time measurements corroborate those of Hamamoto et al. We extend their results by showing significant variations even at the page granularity, an important result for practical exploitation of retention-time variations at the level of pages instead of cells, since pages are the natural refresh granularity. In addition, we also provide a software testing method to characterize the retention times of pages, and discuss the factors that affect the run time, power, and precision of our testing strategy.

Jin et al. [32] propose a comprehensive model of retention time distribution using Monte Carlo methods. This model calculates the retention time distribution for any DRAM device, given the device parameters like doping profile, stress profile, and bias conditions.

4.3.2.2 Dual Period Refresh

Yanagisawa [70], in a patent, classifies DRAM cells into two types – cells with long retention times and cells with short retention times. Modifications to the conventional counter-based refresh circuitry are proposed to refresh cells with long retention times less frequently than cells with short retention times.

Idei et al. [25] propose a dual period self refresh scheme. One refresh period corresponds to the default refresh period based on the worst cell and the other refresh period is a longer refresh period. The longer refresh period is a multiple of the default refresh period. A PROM mode register records the refresh period of each DRAM page to be either short or long, based on the page retention time measurements. A hardware circuit is proposed to refresh each DRAM page according to its refresh period.

4.3.2.3 Multiperiod Refresh

Kim and Papaefthymiou [34][35] and Ohsawa et al. [52] propose custom hardware support in the DRAM chip itself, to refresh different cells at different refresh rates and thereby exploit retention-time variations among memory cells. Kim and Papaefthymiou [34][35] propose a block-based multiperiod refresh approach, where a custom refresh period is selected for each block, i.e., group of cells. They also propose an algorithm to compute the optimal number of blocks. Ohsawa et al. [52] exploit retention-time variations of DRAM pages and propose a hardware technique, where each DRAM page is refreshed at a tailored refresh period that is a multiple of the shortest refresh period among all pages. They also propose another technique which builds on top of this by refreshing only occupied pages.

Our approach is unique, in that it does not require custom hardware modifications to the DRAM. The underlying reason is that a single refresh period is used at any given time.

RAPID software populates longer-retention pages before shorter-retention pages, enabling a single refresh period at any given moment that correlates with the shortest-retention page among only populated pages rather than the shortest-retention page overall. A single refresh period, selected and possibly adjusted by RAPID as DRAM utilization changes, can be used to refresh contemporary off-the-shelf DRAMs for practical exploitation of retention-time variations. We compare our RAPID techniques with the custom hardware refresh techniques proposed by Ohsawa et al. [52] in terms of refresh energy savings.

Kai et al. [33], in a patent, propose a variety of refresh circuit optimizations, including selective refresh of occupied pages and per page refresh periods, in the context of mixed DRAM/Logic chips. They also propose a combined hardware and software approach to place data in DRAM according to page retention times and apply per page refresh periods or a single refresh period accordingly. They use a flag to indicate which rows are occupied and need to be refreshed, and only these rows are refreshed. In contrast, RAPID is a software-only solution, where all DRAM pages are refreshed with a single extended refresh period. In addition, the technique proposed by Kai et al. is in the context of merged DRAM/Logic chips, whereas RAPID is a software-only approach, proposed mainly in the context of exploiting off-the-shelf DRAMs.

Le et al. [39], in a patent, propose a hardware technique for remapping DRAM banks according to their retention times, disabling unused banks, and then setting either a single refresh period or refresh period per bank to refresh the banks that are not disabled. Mori et al. [48], in a patent, propose a hardware circuit to identify and select memory blocks with long retention times.

4.4 Reducing DRAM Energy Consumption in Active Mode

There has been a plethora of work in the last decade to reduce power consumption of DRAMs during active mode. Banked DRAMs have been the focus of several recent efforts to reduce DRAM energy consumption. Hardware techniques, software techniques, and combined hardware-software techniques have been proposed to exploit multiple banks and multiple low-power modes in DRAM.

Delaluz et al. [14] propose a compiler-based approach for reducing power consumption of individual DRAM banks by predicting idle periods for each DRAM bank at compile time. Compiler-based approaches have certain limitations in that not all information is available at compile time. They also propose self-monitored run-time approaches that require hardware support. The idea is that the memory system automatically transitions into lower power modes based on information captured by the supporting hardware. The supporting hardware for checking bank usage incurs energy overhead, which may reduce the energy benefits. As a result, alternative operating system based approaches have been proposed.

Fan et al. [16] and Lebeck et al. [40] propose energy management schemes to exploit the different power modes available in some of the commodity DRAMs (active, standby, nap and power down). Lebeck et al. [40] exploit low-level support for transitioning DRAM chips among four power modes. Their power-aware page allocation polices attempt to maximize use of lower power modes while minimizing performance overhead of transitioning back to higher power modes. The underlying idea is that an application's pages are allocated into a minimum number of banks and unused banks are put in low power modes during the execution of an application.

Delaluz et al. [13] propose a scheduler-based approach for transitioning the unused memory banks to the lowest power mode. This approach uses a bank usage table, managed by the operating system, that resets the power modes of all banks at context switching time. These operating system based methods can be used with hardware methods if hardware cost (energy overhead) is minimal.

Huang et al. [23] propose a mechanism to actively remap memory accesses in the operating system to coalesce idle periods, so that various power management schemes like the one proposed by Lebeck et al. [40] can be used more efficiently.

Huang et al. [24] propose and implement a power-aware virtual memory (PAVM) manager in a linux operating system. The memory usage patterns of all processes in a linux operating system are characterized. This allows PAVM to effectively manage all system memory including kernel memory, dynamically loaded libraries, user process's own private pages, and their interactions and reduce the energy of each process in the system.

Kobayashi et al. [36] propose a software-only technique to reduce DRAM power by efficiently managing DRAM banks. A memory management library is proposed which forecasts the areas of DRAM that are not likely to be touched for extended periods of time and puts these areas into different standby power consumption states.

AbouGhazaleh et al. [1] exploit the cached-DRAM architecture to reduce DRAM power consumption. Cached-DRAM integrates a SRAM cache into the DRAM module to improve performance. The SRAM cache stores most recently accessed DRAM pages. The average DRAM access latency is reduced by accessing data through the fast SRAM cache rather than the slower DRAM. AbouGhazaleh et al. [1] also propose a Power-Aware Cached-DRAM (PA-CDRAM) technique, in which, DRAM banks are completely powered off or transitioned

into low power modes after data is transferred to SRAM cache. An optimal cache configuration is chosen to minimize accesses to DRAM banks.

Ozturk et al. [53] propose replicating data in DRAMs to lower DRAM power in the context of banked memories. Compiler-directed data replication is effective in avoiding re-activating an otherwise idle bank. Algorithms are proposed based on heuristics and profiling to optimally place and replicate data in DRAM banks, yielding significant power savings.

Zhou et al. [71] exploit miss ratio curves to understand the dynamic memory behavior of applications. The miss ratio curves for SPEC2000 benchmarks show that the miss ratio does not improve if more than a certain number of pages are allocated to an application. This information can be used to either improve performance or conserve energy. Performance can be improved by allocating only the required number of pages to multiple applications. Power savings can be achieved by keeping only the required number of pages active and powering down the rest of the memory.

To reduce active power, we exploit the lower operating voltage of ZettaRAM. ZettaRAM is based on DRAM architectures. Therefore, all of the above power-saving techniques, based on transitioning to different power modes, are applicable to ZettaRAM. Also, in all of these power modes, refresh is always present, and RAPID techniques can be applied to reduce refresh power.

4.5 DRAM Data Retention Time Test Patterns and Testing Methods

Data retention time tests work as follows. A specific test pattern is written to a predetermined number of memory cells and read out after a wait time corresponding to the refresh period. The test is then repeated until the entire DRAM chip is tested. Almost two-

thirds of the overall DRAM testing time is because of data retention time tests [65][66][67]. Therefore, a lot of research has been done to reduce the data retention testing time of DRAM [38][20].

Hashimoto et al. [20] propose a circuit technique to reduce data retention test times by a an order of magnitude, by using a charge offset, realized by operating a dummy cell reference voltage. Koike et al. [38] propose a Built-In-Self-Test (BIST) scheme for data retention tests in the context of LSI memories to reduce DRAM testing time. The data retention test patterns used in this work are checkerboard (alternating rows of 1010… and 0101…) and inverse checkerboard.

Van de Goor et al. [65][66][67] and Vollrath [69] describe the standard DRAM tests that are performed in the industry. Multiple test patterns are used in the DRAM industry to identify data retention failures with high coverage. Diverse test patterns are required to identify potential cross-cell retention time degradation and other interactions. The most common test patterns used to test data retention times are the checkerboard and inverse checkerboard [38][20]. Other common test patterns are all 1s, all 0s, alternating 1s and 0s and alternating 0s and 1s (i.e., 1010… and 0101…) [65][66][67] [69].

In our retention time measurements, the six industry-standard test patterns are used to test the ISSI DRAM chip. Statistical inference is applied to safely bound error of testing. We also propose an end-user software testing method to characterize retention times of DRAM pages, providing insights on run-time and power overheads of on-line testing.

4.6 Architectures Influenced by Technology Variation

RAPID can be classified among techniques that exploit the principle of better-than-worst-case design [2][15][63]. Due to future variation-limited technology, better-than-worst-case design has recently received significant attention in the context of microprocessors [2][15][63], for example. Variation-oriented low-power memory techniques such as RAPID can complement these variation-oriented low-power microprocessor techniques.

Chapter 5

Summary

This dissertation addresses the fundamental power scaling challenges facing DRAM. Firstly, DRAM requires continuous refresh because of charge leakage, resulting in power consumption even during standby operation. As density quadruples with each new generation of DRAM, standby power increases correspondingly and will become a significant fraction of overall system power in the future. Secondly, we are fast approaching a lower limit on DRAM voltage scaling because charge in a DRAM capacitor depends directly on voltage, and a minimum amount of charge is needed for reliable operation.

To overcome the DRAM power scaling challenge, our method of attack has been to identify unique properties of the underlying technology and exploit them at the system level.

Scalable Standby Power via RAPID

To achieve favorable standby power scaling, we proposed a software technique that nearly eliminates refresh power. This technique exploits the fact that charge leakage is typically low in most DRAM cells, and frequent refreshes are because of a small number of atypical leaky cells. Intelligent placement of data by the operating system can avoid the use of atypical leaky cells, which can be exploited to reduce the frequency of refreshes by orders of magnitude.

Accordingly, we proposed Retention-Aware Placement in DRAM (RAPID), novel software approaches that can exploit off-the-shelf DRAMs to reduce refresh power to

vanishingly small levels. RAPID-1 is a static approach, where the samll fraction of outlier pages (1%) are mapped out, enabling an extended refresh period and corresponding refresh power savings of 83% relative to ideal temperature-compensated refresh (TCR) at 25°C. RAPID-2 is a dynamic technique, where the key idea is to favor longer-retention pages over shorter-retention pages, when allocating DRAM pages to application programs. A single refresh period is then selected based on the shortest-retention page among populated pages, instead of the shortest-retention page overall. RAPID-2 yields 93% refresh power savings at 25°C. RAPID-3 is basically RAPID-2, with the added ability to continuously reconsolidate data into the longest-retention pages. RAPID-3 yields 95% refresh power savings at 25°C. Both RAPID-2 and RAPID-3 are nearly as effective as idealized techniques that require custom DRAM support. Summing up, RAPID reduces DRAM standby power to near-zero levels, yielding favorable standby power scaling despite quadrupling of DRAM capacity from one generation to the next.

Scalable Voltage via ZettaRAM

To overcome DRAM voltage scaling limits, we explored the long-term power scalability of ZettaRAM™, an emerging DRAM technology based on molecular electronic components. ZettaRAM preserves the classic 1T-1C DRAM cell structure, except the conventional capacitor is replaced with a molecular capacitor. Unlike a conventional capacitor, a fixed charge is deposited on the molecular capacitor, the magnitude of which does not depend on the applied voltage. Charge-voltage decoupling holds the key to viable voltage scaling from one generation to the next.

Charge-voltage decoupling enables read/writes to be performed at lower voltages, while still maintaining the minimum amount of charge needed for reliable sensing. Voltages can be

lowered in ZettaRAM from one generation to the next by synthesizing new molecules with lower threshold voltages. Key properties of the molecules can be precisely tuned through the choice of molecular "groups" and "linkers", such as the oxidation potential (V_{ox}), electron transfer rate (k^0), and surface concentration (charge density). We provided novel analyses to explain how these three parameters influence the operating voltage either directly or indirectly.

We explored 23 molecules documented in the literature, to confirm our hypothesis that ZettaRAM provides an inexpensive path to scaling power in the long term, i.e., molecular attributes can be tuned to scale voltage (hence power) from one memory generation to the next. Among these, we identified molecules that yield ZettaRAMs comparable to or better than DRAM in all respects – density, performance, and power. Our studies revealed that the best molecule yields a ZettaRAM voltage of 0.65V, whereas the charge-voltage dependence of the conventional capacitor limits DRAM voltage to 1.25V.

Intelligent Management of ZettaRAM

We also proposed dynamically modulating the ZettaRAM voltage to reduce power even more. While the fixed charge is independent of applied voltage, the speed of charging/discharging the molecules depends on the difference between the applied voltage and the oxidation potential (faster with larger differential). Thus, for good performance, the applied voltage is padded with respect to the oxidation potential. Thus, there is some room to lower voltage further, if performance can be dealt with by some other means.

We systematically applied architectural principles to manage this speed/energy tradeoff and showed that the write voltage can be dynamically modulated to nearly achieve ZettaRAM's full power savings potential with little system slowdown.

A key architectural insight is that almost 80% of the bitline activity is caused by L2 cache writebacks. This is significant because writebacks do not directly stall the processor and thereby offer scheduling flexibility to tolerate higher molecule latencies at lower voltages. Hence, we can tap most of the energy savings potential by focusing on L2 cache writebacks. Accordingly, we proposed a hybrid write policy, in which slow writes (lower voltage) are applied to non-timing-critical writebacks and fast writes (higher voltage) are applied to timing-critical fetches. We characterized the additional energy savings yielded by the hybrid write policy, for one of the first synthesized porphyrin molecules: if fast and slow writes are done at 1.2 V and 1.0 V, respectively, then the hybrid write policy yields 34% energy savings (out of a possible 41% with uniformly slow writes) with only a 10% increase in execution time (as opposed to 81% with uniformly slow writes).

The residual 10% performance loss is caused by delayed writebacks filling up request queues in the memory controller, eventually stalling critical fetches. These indirect stalls can be prevented by (i) increasing the request queue sizes or (ii) employing L2 cache eager writebacks. Larger queues increase both cost and complexity. Eager writebacks are remarkably synergistic with low-power ZettaRAM: initiating writebacks early in the L2 cache compensates for delaying them at the memory controller. For SPEC2K benchmarks, applying hybrid fast/slow writes coupled with small queues and the L2 cache eager writeback policy yields energy savings of 34% (out of a possible 41% with uniformly slow writes), with less than 1% performance degradation.

Thus, the combination of hybrid fast/slow writes and L2 cache eager writebacks nearly achieves the lowest energy possible with ZettaRAM, with little performance loss and the least complexity in the memory controller. Moreover, 6 of the molecules that yield higher

energy than DRAM with uniformly fast writes yield lower energy when the hybrid write policy is applied.

RAPID and ZettaRAM

In conclusion, the combination of RAPID and ZettaRAM leads to a power-scalable form of DRAM, extending the roadmap of this important memory technology.

Chapter 6

Directions for Future Research

This dissertation addresses the power scaling challenges of DRAM. We proposed RAPID for favorable standby power scaling and explored the superior power scaling characteritics of ZettaRAM to overcome voltage scaling limits of DRAM. However, there are many interesting facets of both RAPID and ZettaRAM that are yet to be explored.

In the area of RAPID software, it may be interesting to look at modifications that would be required for handling differing O/S and DRAM pages sizes. Another interesting aspect would be to explore the benefits of sub-page retention-aware placement (e.g., padding data structures to skip over subpar cells). We proposed a binning strategy in RAPID-2/RAPID-3 based on dividing bins into equal time intervals. Other strategies may yield more optimal page distributions in terms of lowering frequency of refreshes or migrations.

There are opportunities to explore novel interactions between RAPID and ECC, for example, defining the retention time of a page according to its second-worst bit and leveraging ECC to repair the worst bit.

Another interesting exploration would be to implement RAPID on a real system, e.g., PDA, cellular phone, or pocket PC. Energy savings could then be measured while running application programs on a real system and compared with the theoretical results obtained for RAPID in this dissertation.

Retention times decrease with increasing temperature. Performing either on-line or off-line retention-time tests for different temperature ranges may not be feasible in practice. Fortunately, an analytical model describing both the effects of process variation and temperature on retention times was proposed by Hamomoto et al. [18]. This model can be leveraged to couple RAPID with TCR. In this case, retention-time tests only need to be performed at one temperature. Retention times can then be scaled for any temperature using the temperature scaling factor provided by Hamomoto et al. [19]. Also, if tests are performed for numerous temperatures ranges, the storage cost is high to store the retention time data for each temperature range. The approach of analytically coupling RAPID with TCR can reduce this storage cost, as we need to store only retention times for one temperature.

Bibliography

[1] N. AbouGhazaleh, B. Childers, D. Mosse, R. Melhem. Near-memory caching for improved energy consumption. International Conference on Computer Design, Oct. 2005, pp. 105 – 108.

[2] T. Austin, D. Blaauw, T. Mudge, K. Flautner. Making Typical Silicon Matter with Razor. IEEE Computer, Volume 37, Number 3, March 2004.

[3] A. J. Bard and L. R. Faulkner. Electrochemical Methods: Fundamentals and Applications, John Wiley and Sons, 2001, pp. 92-96.

[4] R. Barth. ITRS Commodity Memory Roadmap. International Workshop on Memory Technology, Design and Testing. July 2003, pp. 61-63.

[5] S. Borkar. Getting Gigascale Chips: Challenges and Opportunities in Continuing Moore's Law. ACM Queue, Vol. 1, No. 7, Oct. 2003.

[6] J. Burgan. Variable Refresh Control for a Memory. US Patent #6,778,457, Aug. 2004.

[7] D. Burger, T. Austin, and S. Bennett. Evaluating Future Microprocessors: The Simplescalar Toolset. Tech. Report CS-TR-96-1308, CS Department, Univ. of Wisc.-Madison, July 1996.

[8] CellularRAM™ – Micron website:

http://download.micron.com/pdf/flyers/cellularram_flyer.pdf.

[9] A. P. Chandrakasan and R. W. Brodersen. Minimizing Power Consumption in Digital CMOS Circuits. In Proc. of the IEEE, 83(4):498-523, April 1995.

[10] L.L. Chen Hsu, G. Frankowsky, O. Weinfurtner. Dynamic DRAM Refresh Rate Adjustment Based on Cell Leakage Monitoring. US Patent #6,483,764, Nov. 2002.

[11] H.Y. Cho, J.K. Oh. Self-Refresh Apparatus for a Semiconductor Memory Device. US Patent # 6,229,747, May 2001.

[12] D.C. Choi, Y.R. Kim, et al. Battery operated 16M DRAM with Post Package Programmable and Variable Self Refresh. 1994 Symposium on VLSI Circuits Digest of Technical Papers, 1994, pp. 83-84.

[13] V. Delaluz, A. Sivasubramaniam, M. Kandemir, N. Vijaykrishnan, and M. J. Irwin. Scheduler-based DRAM Energy Management. Design Automation Conference, June 2002.

[14] V. Delaluz, M. Kandemir, N. Vijaykrishnan, A. Sivasubramaniam, and M. J. Irwin. DRAM Energy Management Using Software and Hardware Directed Power Mode Control, International Symposium on High Performance Computer Architecture (HPCA-7), pp. 159-169, Jan. 2001.

[15] D. Ernst, N. S. Kim, S. Das, S. Pant, R. Rao, T. Pham, C. Ziesler, D. Blaauw, T. Austin, K. Flautner, T. Mudge. Razor: A Low Power Pipeline Based on Circuit-Level Timing Speculation. MICRO-36, IEEE CS Press, 2003, pp.7-18.

[16] X. Fan, C. S. Ellis, and A. R. Lebeck. Memory Controller Policies for DRAM Power Management. International Symposium on Low Power Electronics and Design, Aug. 2001.

[17] K. Flautner, N. S. Kim, S. Martin, D. Blaauw, and T. Mudge. Drowsy Caches: Simple Techniques for Reducing Leakage Power. International Symposium on Computer Architecture, July 2002.

[18] S. Fyffe. DRAMatic Cell Phone Showdown. Electronic News, 2/12/2001. M. Kanellos, B. Charny. Phone makers copy PCs. CNET News.Com, 4/2/2002. Dram Makers Prep Multichip Cell Phone Memories. ElectroSpec, 1/3/2003.

[19] T. Hamamoto, S. Sugiura, S. Sawada. On the Retention Time Distribution of Dynamic Random Access Memory (DRAM). IEEE Transactions on Electron Devices, Volume 45 Issue 6, June 1998, pp.1300-1309.

[20] M. Hashimoto, I. Uchida, S. Hatakoshi. Over an Order of Magnitude DRAM Test Time Reduction by Charge Offset. In Proc. of IEEE Custom Integrated Circuits Conference, May 1996, pp. 483-486.

[21] M. Hempstead, N. Tripathi, P. Mauro, G. Wei, D. Brooks. An Ultra Low Power System Architecture for Sensor Network Applications. 32nd International Symposium on Computer Architecture, June 2005.

[22] G. Hinton, D. Sager, M. Upton, D. Boggs, D. Carmean, A. Kyker, P. Roussel. The Microarchitecture of the Pentium® 4 Processor. Intel Technology Journal Q1, 2001.

[23] H. Huang, K. Shin, C. Lefurgy, T. Keller. Improving Energy Efficiency by making DRAM Less Randomly Accessed. ISPLED'05, pp. 393-398, Aug. 2005.

[24] H. Huang, P. Pillai and K. Shin. Design and Implementation of Power-Aware Virtual Memory. USENIX Annual Technical Conf., pp. 57–70, 2003.

[25] Y. Idei, K. Shimohigashi, M. Aoki, H. Noda, H. Iwai, K. Sato, T. Tachibana. Dual-period Self-refresh Scheme for Low-power DRAM's with On-chip PROM Mode Register. IEEE Journal of Solid-State Circuits. Volume 33, Issue 2, 1998, pp. 253-259.

[26] Infineon – Mobile-DRAM, Specialty DRAMs. Application Note, V1.1, Feb. 2002.

[27] Innovative Mobile Products Require Early Access to Optimized DRAM Solutions. A Position Paper by Samsung Semiconductor, Inc., 2003.

[28] K. Itoh, K. Sasaki, Y. Nakagome. Trends in Low-Power RAM Circuit Technologies. In Proc. of the IEEE, 83(4):524-543, April 1995.

[29] K. Itoh. VLSI Memory Chip Design, Springer Series in Advanced Microelectronics, 2001, pp. 117-123.

[30] K. Itoh. VLSI Memory Chip Design, Springer Series in Advanced Microelectronics, 2001, p. 403.

[31] K. Itoh, Y. Nakagome, S. Kimura, T. Watanabe. Limitations and Challenges of Multigigabit DRAM Chip Design. IEEE Journal of Solid-State Circuits. 32(5):624-634, May 1997.

[32] S. Jin, J-H Yi, J. H. Choi, D. G. Kang, Y. J. Park, H. S. Min. Prediction of Data Retention Time Distribution of DRAM by Physics-Based Statistical Simulation. IEEE Transactions on Electron Devices. Volume 52, No. 11, Nov. 2005.

[33] K. Kai, T. Ohsawa, K. Murakami. Semiconductor circuit and method of controlling the same. US Patent #6,446,159, Sep. 2002.

[34] J. Kim, M. Papaefthymiou. Block-Based Multiperiod Dynamic Memory Design for Low Data-Retention Power. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol. 11, No. 6, Dec. 2003, pp.1006-1018.

[35] J. Kim, M. Papaefthymiou. Block-Based Multi-Period Refresh for Energy Efficient Dynamic Memory. 14th ASIC/SOC Conference, Sept. 2001, pp 193-197.

[36] S. Kobayashi, A. Fukuda. Design of a library for DRAM power reduction in an embedded multi-task kernel. IEEE Pacific Rim Conference on Communications, Computers and signal Processing, PACRIM, Volume 1, pp. 5-8, Aug 2003.

[37] T. Komatsu. Self-Refreshing of Dynamic Random Access Memory Device and Operating Method Therefor. US Patent #4,943,960, July 1990.

[38] H. Koike, T. Takeshima, M. Takada. A BIST scheme using microprogram ROM for large capacity memories. In Proc. of International Test Conference, Sept. 1990, pp. 815-822.

[39] T. Le, S. Bowyer. Bank Address Mapping According to Bank Retention Time in Dynamic Random Access Memories. US Patent #6,920,523, July 2005.

[40] A. Lebeck, X. Fan, H. Zeng, C. Ellis. Power Aware Page Allocation. 9th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS), Nov. 2000.

[41] H. S. Lee, G. S. Tyson, M. K. Farrens. Eager Writeback – a Technique for Improving Bandwidth Utilization. 33rd International Symposium on Microarchitecture, pp. 11-21, 2000.

[42] J. Lee, D. Ha, and K. Kim. Novel Cell Transistor Using Retracted Si_3N_4-Liner STI for the Improvement of Data Retention Time in Gigabit Density DRAM and Beyond. IEEE Transactions on Electron Devices. Volume 48, No. 6, June 2001.

[43] J. W. Lee, Y. S. Kim, J. Y. Kim, Y. K. Park, S. H. Shin, S. H. Lee, J. H. Oh, J. G. Lee, J. Y. Lee, D. I. Bae, C. Lee, C. S. Lee, C. J. Yun, C. H. Cho, K. Y. Jin, Y. J. Park, T. Y. Chung, K. Kinam. Improvement of data retention time in DRAM using recessed channel array transistors with asymmetric channel doping for 80 nm feature size and beyond. In Proc. of 34th European Solid-State Device Research Conference, ESSDERC 2004.

[44] Z. Liu, A. A. Yasseri, J. S. Lindsey, D. F. Bocian. Molecular Memories That Survive Silicon Device Processing and Real-World Operation. Science Volume 302, 1543-1545, Nov. 2003.

[45] J. A. Mandelman, R. H. Dennard, G. B. Bronner, J. K. DeBrosse, R. Divakaruni, Y. Li, C. J. Radens. Challenges and Future Directions for the scaling of Dynamic Random-access Memory (DRAM). IBM J. Res. and Dev. Vol. 46, No. 2/3, Mar/May 2002.

[46] Micron SDRAM 8Mx16x4 Part No. MT48LC32M16A2TG-75, 2003.

[47] D. S. Moore, G. P. McCabe. Introduction to the Practice of Statistics. Third Edition, W. H. Freeman and Co., 1999.

[48] M. Mori, T. Sunaga, S. Watanabe. DRAM and Refresh Method Thereof. US Patent #6,977,857, Dec. 2005.

[49] Y. Mori, Y. Takeda, S. Kimura, K. Ohyu, H. Uchiyama, R. Yamada. A new experimental method for evaluating electric field at the junctions of DRAM cell transistors and the effect of electric field strength on the retention characteristics of DRAM. 42nd IEEE International Reliability Physics Symposium Proceedings, April 2004.

[50] T. Murotani. Temperature Responsive Refresh Control Circuit. US Patent #4,393,477, June 1983.

[51] L. Nazahandali, B. Zhai, J. Olson, A. Reeves, M. Minuth, R. Helfand, S. Pant, T. Austin, D. Blaauw. Energy Optimization of Subthreshold-Voltage Sensor Network Processors. 32nd International Symposium on Computer Architecture, June 2005.

[52] T. Ohsawa, K. Kai, K. Murakami. Optimizing the DRAM Refresh Count for Merged DRAM/Logic LSIs. ISPLED98, pp. 82-87, August 1998.

[53] O. Ozturk, M. Kandemir. Data Replication in Banked DRAMs for Reducing Energy Consumption. 7th International Symposium on Quality Electronic Design, 2006. ISQED '06, March 2006.

[54] C. Park, J. Seo, D. Seo, S. Kim and B. Kim. Cost-Efficient Memory Architecture Design of NAND Flash memory Embedded Systems. In Proc. of the 21st International Conference on Computer Design (ICCD '03), Oct. 2003.

[55] P. Pelley, J. Burgan. Memory Having Variable Refresh Control and Method Therefor. US Patent #6,781,908, Aug. 2004.

[56] K. M. Roth, D. T. Gryko, P. C. Clausen, J. Li, J. S. Lindsey, W. G. Kuhr, D. F. Bocian. Comparison of Electron-Transfer and Charge-Retention Characteristics of

Porphyrin-Containing Self-Assembled Monolayers Designed for Molecular Information Storage. J. Phys. Chem. B, 106, 8639-8648, 2002.

[57] K. M. Roth, N. Dontha, R. B. Dabke, D. T. Gryko, C. Clausen, J. S. Lindsey, D. F. Bocian, W. G. Kuhr. Molecular Approach Toward Information Storage Based on the Redox Properties of Porphyrins in Self-Assembled Monolayers. J. Vac. Sci. Technology B, 18, 2359–2364, 2000.

[58] K. M. Roth, J. S. Lindsey, D. F. Bocian, W. G. Kuhr. Characterization of Charge Storage in Redox-Active Self-Assembled Monolayers. Langmuir, 18, 4030–4040, 2002.

[59] T. Sherwood, E. Perelman, G. Hamerly and B. Calder. Automatically Characterizing Large Scale Program Behavior. 10th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS), Oct. 2002.

[60] C. W. Teng. DRAM Technology Trend. Proc. of the 1995 International Symposium on VLSI Technology, Systems and Applications, 1995, p. 295-299.

[61] Ubicom. http://www.ubicom.com

[62] S. Ueno, Y. Inoue, M. Inuishi. Scaling guideline of DRAM memory cells for maintaining the retention time. IEEE Symposium on VLSI Technology, Digest of Technical Papers, June 2000.

[63] A. Uht. Going Beyond Worst-Case Specs with TEAtime. IEEE Computer, Volume 37 Number 3, March 2004.

[64] UtRAM™ - Samsung website

http://www.samsung.com/Products/Semiconductor/SRAM

[65] A. J. Van de Goor, J. de Neef. An Industrial Evaluation of DRAMs Tests. In Proc. of IEEE Design and Test of Computers, Volume 21, Issue 5, Sept-Oct 2004, pp. 430-440.

[66] A. J. Van de Goor, J. de Neef. Industrial Evaluation of DRAMs Tests. In Proc. of Design Automation and Test Conference in Europe, March 1999, pp. 623-630.

[67] A. J. Van de Goor, A. Paalvast. Industrial Evaluation of DRAMs SIMM Tests. In Proc. of International Test Conference, Oct 2000, pp. 426-435.

[68] M. Viredaz, D. Wallach. Power Evaluation of a Handheld Computer: A Case Study. Compaq-WRL Research Report 2001/1, 2001. ITSY Link:

http://www.hpl.hp.com/research/papers/2003/handheld.pdf

[69] J. E. Vollrath. Testing and Characterization of SDRAMs. IEEE Transcations on Design and Test of Computers, Volume 20, Issue 1, Jan-Feb 2003, pp. 42-50.

[70] K. Yanagisawa. Semiconductor Memory. US Patent #4,736,344, April 1988.

[71] P. Zhou, V. Pandey, J. Sundaresan, A. Raghuraman, Y. Zhou. Dynamic Tracking of Page Miss Ratio Curve for Memory Management. 12th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS), Oct. 2004.

[72] ZettaCore - http://www.zettacore.com

Appendices

Appendix A

Equilibrium Charge in Molecular Capacitor

In this appendix, we derive the equilibrium charge present in the molecular capacitor as a function of write voltage.

The oxidation/reduction reaction is shown below, where A is a porphyrin molecule [56].

$$A \leftrightarrow A^{+} + e^{-} \qquad \text{(EQ A-1)}$$

In non-equilibrium (charging or discharging), the net rate of oxidation or reduction, i.e., the net current is expressed by the Butler-Volmer kinetic model [3], shown below. The current I determines the *intrinsic* speed of reading and writing the molecule.

$$I = F \cdot k^{0} \cdot \left([A] \cdot e^{(1-\alpha)\left(\frac{F}{RT}\right)(V - V_{ox})} - [A^{+}] \cdot e^{-\alpha\left(\frac{F}{RT}\right)(V - V_{ox})} \right) \qquad \text{(EQ A-2)}$$

Parameters above are as follows: k^0 = standard rate constant, α = transfer coefficient, F = Faraday constant, R = gas constant, T = temperature, V = applied voltage, V_{ox} = oxidation potential, [A] = concentration of non-oxidized molecules (in moles per unit area), and $[A^+]$ = concentration of oxidized molecules.

The oxidation/reduction reactions shown in EQ A-1 eventually reach an equilibrium. The net current is zero at this equilibrium. We can derive the amount of charge ($Q_{cell} = [A^+]$) at equilibrium as a function of the write voltage, by substituting I=0 in the Butler-Volmer equation (EQ A-2).

$$[A]\cdot e^{(1-\alpha)\left(\frac{F}{RT}\right)(V-V_{ox})} = [A^+]\cdot e^{-\alpha\left(\frac{F}{RT}\right)(V-V_{ox})} \qquad \text{(EQ A-3)}$$

Let $[A]_0$ be the total molecule concentration.

$$[A]_0 = [A^+]+[A] \qquad \text{(EQ A-4)}$$

Therefore, substituting EQ A-4 in EQ A-3,

$$\left\{[A]_0 - [A^+]\right\}\cdot e^{(1-\alpha)\left(\frac{F}{RT}\right)(V-V_{ox})} = [A^+]\cdot e^{-\alpha\left(\frac{F}{RT}\right)(V-V_{ox})}$$

Rearranging, $$[A^+]\cdot\left\{e^{-\alpha\left(\frac{F}{RT}\right)(V-V_{ox})} + e^{(1-\alpha)\left(\frac{F}{RT}\right)(V-V_{ox})}\right\} = [A]_0\cdot e^{(1-\alpha)\left(\frac{F}{RT}\right)(V-V_{ox})}$$

$$[A^+] = [A]_0\cdot\left\{\frac{e^{(1-\alpha)\left(\frac{F}{RT}\right)(V-V_{ox})}}{e^{-\alpha\left(\frac{F}{RT}\right)(V-V_{ox})} + e^{(1-\alpha)\left(\frac{F}{RT}\right)(V-V_{ox})}}\right\}$$

$$[A^+] = [A]_0\cdot\left\{\frac{1}{\frac{e^{-\alpha\left(\frac{F}{RT}\right)(V-V_{ox})}}{e^{(1-\alpha)\left(\frac{F}{RT}\right)(V-V_{ox})}}+1}\right\}$$

$$Q_{cell}(V) = [A^+] = [A]_0\cdot\left\{\frac{1}{1+e^{-\left(\frac{F}{RT}\right)(V-V_{ox})}}\right\} \qquad \text{(EQ A-5)}$$

EQ A-5 is the basis for the unusual charge density graph shown in Figure 1-4 in Introduction, Section 1.2.1. EQ A-5 gives us the effective capacitance of the molecular capacitor since capacitance expresses Q as a function of V.

Appendix B

Intrinsic Molecule Write Latency

In this appendix, we derive the intrinsic molecule write latency, t_{mol_write} as a function of molecular parameters V_{ox}, k^0, and Q_0.

The oxidation/reduction reaction is shown below, where A is a porphyrin molecule [56].

$$A \leftrightarrow A^+ + e^- \qquad \text{(EQ B-1)}$$

In non-equilibrium (charging or discharging), the net rate of oxidation or reduction, i.e., the net current is expressed by the Butler-Volmer kinetic model [3], shown below. The current I determines the *intrinsic* speed of reading and writing the molecule.

$$I = F \cdot k^0 \cdot \left([A] \cdot e^{(1-\alpha)\left(\frac{F}{RT}\right)(V - V_{ox})} - [A^+] \cdot e^{-\alpha\left(\frac{F}{RT}\right)(V - V_{ox})} \right) \qquad \text{(EQ B-2)}$$

Parameters above are as follows: k^0 = standard rate constant, α = transfer coefficient, F = Faraday constant, R = gas constant, T = temperature, V = applied voltage, V_{ox} = oxidation potential, [A] = concentration of non-oxidized molecules (in moles per unit area), and $[A^+]$ = concentration of oxidized molecules.

EQ B-2 can be rewritten as,

$$I = k_O \cdot F \cdot [A] - k_R \cdot F \cdot [A^+] \qquad \text{(EQ B-3)}$$

$$k_O = k^0 \cdot e^{(1-\alpha)\left(\frac{F}{RT}\right)(V-V_{ox})} \qquad k_R = k^0 \cdot e^{-\alpha\left(\frac{F}{RT}\right)(V-V_{ox})}$$

where, k_O = oxidation rate constant, and k_R = reduction rate constant.

Let $Q_0 = F\cdot[A]_0$ be the total number of molecule charge in the molecular capacitor, equal to the surface concentration of the molecules multiplied by the cell area. Let $Q = F\cdot[A^+]$.

$$Q_0 = Q + F\cdot[A] \qquad \text{(EQ B-4)}$$

Substituting EQ B-4 in EQ B-3 and expressing *current I* as dQ/dt

$$\frac{dQ}{dt} = k_O \cdot (Q_0 - Q) - k_R \cdot Q$$

Rearranging,
$$\frac{dQ}{dt} = k_O \cdot Q_0 - Q \cdot (k_O + k_R)$$

$$dt = \frac{dQ}{k_O \cdot Q_0 - Q \cdot (k_O + k_R)}$$

$$t = \int \frac{dQ}{k_O \cdot Q_0 - Q \cdot (k_O + k_R)} \qquad \text{(EQ B-5)}$$

Integrating EQ B-5,

$$t = -\frac{\ln(k_O \cdot Q_0 - Q \cdot (k_O + k_R))}{k_O + k_R} + C \qquad \text{(EQ B-6)}$$

The integration constant in EQ B-6 can be determined by using the following boundary condition: at t = 0, Q = 0. Substituting this into EQ B-6,

$$C = \frac{\ln(k_O \cdot Q_0)}{k_O + k_R} \qquad \text{(EQ B-7)}$$

Therefore, EQ B-6 can be rewritten as,

$$t = \frac{1}{k_O + k_R} \cdot \{ \ln(k_O \cdot Q_0) - \ln(k_O \cdot Q_0 - Q \cdot (k_O + k_R)) \}$$

$$t = \frac{1}{(k_O + k_R)} \cdot \ln\left\{ \frac{k_O \cdot Q_0}{k_O \cdot Q_0 - (k_O + k_R) \cdot Q} \right\}$$

$$t = \frac{1}{(k_O + k_R)} \cdot \ln\left\{ \frac{k_O}{k_O - (k_O + k_R) \cdot \left(Q / Q_0 \right)} \right\} \qquad \text{(EQ B-8)}$$

Let Q_f be the final charge in the molecular capacitor determined by the minimum charge required by the sense amplifiers for reliable sensing, and t_{mol_write} be the intrinsic molecule write latency. Substituting $t = t_{mol\_write}$ and $Q = Q_f$ in EQ B-8,

$$t_{mol\_write} = \frac{1}{(k_O + k_R)} \cdot \ln\left\{ \frac{k_O}{k_O - (k_O + k_R) \cdot \left(Q_f / Q_0 \right)} \right\} \qquad \text{(EQ B-9)}$$

$$k_O = k^0 \cdot e^{(1-\alpha)\left(\frac{F}{RT}\right)(V - V_{ox})} \qquad k_R = k^0 \cdot e^{-\alpha\left(\frac{F}{RT}\right)(V - V_{ox})}$$

EQ B-9 shows the intrinsic molecule write latency t_{mol_write} as a function of the molecular parameters V_{ox}, k^0, and Q_0. The relationship between write voltage and the three molecular parameters – V_{ox}, k^0, and Q_0 – is expressed indirectly by EQ B-9. We fix t_{mol_write} and then

numerically solve for write voltage, given parameters V_{ox}, k^0, and Q_0 for a particular molecule.