

# Analyzing Power Variability of DDR3 Dual Inline Memory Modules

Mark Gottscho<sup>1</sup>, Dr. Puneet Gupta<sup>2</sup>, Abde Ali Kagalwalla<sup>1</sup>  
NanoCAD Lab, UCLA Electrical Engineering  
<sup>1</sup>{mgottscho | abdeali }@ucla.edu, <sup>2</sup>puneet@ee.ucla.edu

**Abstract**—Variability in modern digital integrated circuits is emerging as an important area of research as it can cause profound impacts on chip performance and power consumption. We found that memory power consumption in a mainstream Atom-based computer can be as much as 18.5% of the total system power. We investigated the power consumption of DDR3 SDRAM DIMMs, finding that power usage in memory is heavily dependent both on operation type (write, read, and idle) as well as data. A low-level benchmark was constructed for DIMM power consumption to analyze power variability in mainstream memory products from various vendors and suppliers. Temperature had little effect (on the order of 1-3%) across the -50C to 50C range. Variations between specimens of the same model and different models of the same vendor were on the order of 5-15%. In the scope of all tested modules, deviations were up to approximately 20% in write and idle power. Furthermore, we found that capacity has a direct effect on power consumption of a DIMM. These power deviations could be exploited by an operating system to improve overall system power efficiency, particularly for memory-bottlenecked systems and applications. We propose several possible approaches to implement power and variability-aware modifications to a kernel.

**Keywords**—DDR3; DRAM; DIMM; variability; variability-aware; power-aware; kernel

## I. INTRODUCTION

Modern digital integrated circuits (ICs) exhibit significant variability as a consequence of imperfections in the fabrication processes [1], [2]. Often, the market compensates by partitioning functional dice by performance, but there is also variation within each of these groups. A typical computer is not “aware” of the actual power and performance characteristics of its particular hardware, knowing only its reported specifications. It is possible to design a software solution (in an operating system kernel) that is aware of its particular hardware instance, able to adapt to subtle variations in each component. Software has the advantage of global instead of local control offered by dedicated hardware. It also allows for updates and new techniques as research in this area progresses. One application of a variability-aware kernel could exploit power variability in dynamic random-access memory (DRAM) to reduce overall memory power consumption. This technique could be useful in any system where memory consumes a considerable proportion of power – for example, in certain supercomputing architectures, memory may consume up to 48% of total power [3]. Using the PARSEC Canneal

benchmark on a low-power Intel Atom platform, we have observed up to 18.5% of total CPU and memory power consumed purely by the memory [4], [5], [6]. Previous work in this area developed power-aware virtual memory systems [7]. While these designs reduced power consumption of main memory by minimizing the active bank utilization, they did not take into account hardware variability; instead, they assumed all banks to be of equal performance and efficiency. An initial variability-aware implementation might work at the dual-inline memory module (DIMM) level of granularity, but these techniques could be extended to individual DRAMs or even banks within DRAMs for fine control.

To find out if DRAM hardware variations are significant enough for the development of a variability-aware kernel modification, we analyzed the write, read, and idle power consumption of several mainstream DDR3 DIMMs, comprised of parts from several vendors and suppliers.

## II. MEMORY SYSTEM BACKGROUND

### A. Overview

Today’s general purpose computers and servers utilize Dynamic Random Access Memory (DRAM) for the primary memory architecture. In a modern hierarchical memory system, DRAM lies between one or more high-performance, small, and expensive caches and slow, large, and cheap non-volatile storage (i.e. hard drives and flash). Currently, DRAM is the ideal tradeoff between capacity, performance, and price for main memory. Due to its random access capabilities, data and instructions can be stored at any location within the storage arrays.

In a typical DRAM-based memory system, a memory controller interfaces with one or more *DIMMs* across one or more memory bus *channels*, each of which operate in parallel. Each DIMM typically consists of one or two *ranks*, which consist of eight (for DDR3) DRAM devices. Each DRAM is divided into several *banks*, each of which contains a storage *array*. These arrays are accessed through rows and columns decoded from the memory address; the number of rows,  $m$ , is typically much greater than the number of columns,  $n$ . Finally, each column consists of  $b$  bits. The finest level of granularity is thus  $b$  bits; for any given access, the memory always must work with this

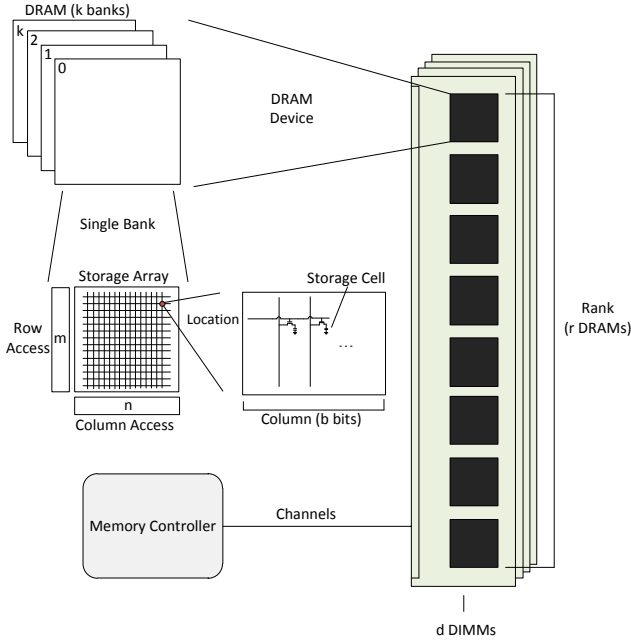


Figure 1. Memory System Components Breakdown [8]

fundamental data size. A diagram of the main memory components is depicted in Figure 1.

A bit in DRAM is maintained as charge on a capacitor, accessed through an NMOS transistor connected to a *bit line*. A full charge of the cell capacitor at  $V_{DD}$  represents a 1, while a discharged capacitor represents a 0. The NMOS transistor acts as a switch for the charge, and is off when the *word line* is low. This maintains the bit stored in the cell. When the word line is asserted (pulled up to  $V_{DD} + V_T$ ), the NMOS transistor turns on and allows writes or reads from the cell.

There are three primary stages in a simplified DRAM operation. Firstly, all of the bit lines in a bank must be *pre-charged*, where the voltage is set to  $V_{ref} = V_{DD}/2$  [9]. Next, an *access* occurs on the word line corresponding to the decoded row address. This causes all the cells along the selected word line to share their charges with their bit lines. In the third stage, a *sense* operation occurs, which is then followed by *restoration* of the accessed cells and either a *read* or *write* on the selected column. Figure 2 illustrates the primary components involved in a memory operation.

### B. Pre-Charge

A DRAM cell typically has a much smaller capacitance than the bit line to which it is connected; in a simple array, a bit line might be connected to all  $m$  rows in the bank [10]. For this reason, the bit line must be pre-charged to  $V_{ref}$ , as

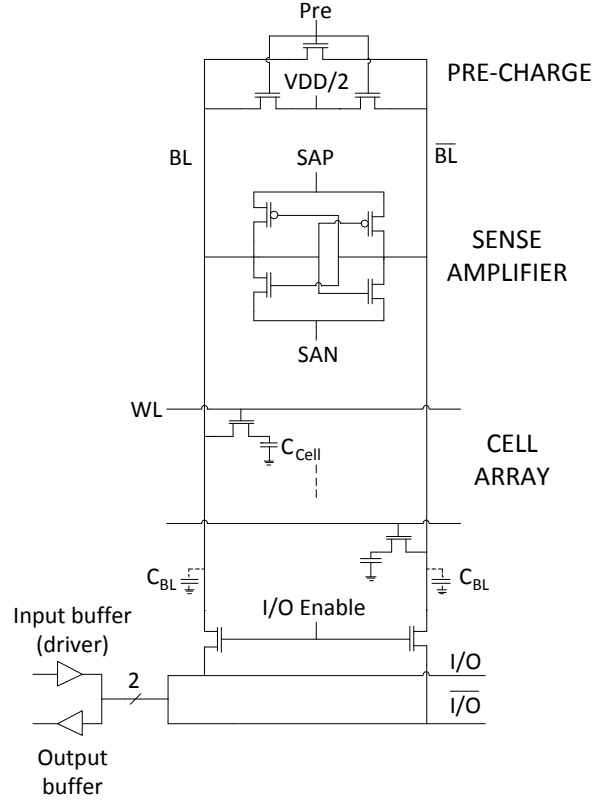


Figure 2. Cell Access Components [10]

mentioned above, so that the value stored in the cell may be detected in the sense and restore phase (see Access as well as Sense and Restore below). This is done by asserting a signal *Pre*, as seen in Figure 2, that turns on three NMOS transistors to connect *BL* to  $\overline{BL}$ . This effectively shorts the two bit lines together such that they stabilize at  $V_{ref}$  (after a sense and restore operation and before pre-charge, one bit line will be at  $V_{DD}$  and the other at  $GND$ ).

### C. Access

In the access stage, a word line (*WL* in Figure 2) is used to select an entire row of cells, by turning all of them on when asserted. A column, consisting of several bit lines, selects which section of that row to access. When a cell is accessed, it shares its charge with the bit line, contaminating the data within the cell. Assuming the pre-charge stage has occurred and the bit line voltage is initially  $V_{ref}$ , the bit line voltage is perturbed by a small signal [10]

$$v_s = V_{ref} \cdot \frac{C_{cell}}{C_{bitline} + C_{cell}}$$

If the cell stored a 1, then let  $V_{ref}^+ = V_{ref} + v_s$ . Otherwise, let  $V_{ref}^- = V_{ref} - v_s$ . After an access, both the bit line and the cell settle at either  $V_{ref}^+$  or  $V_{ref}^-$ , depending on whether

the cell stored a 1 or 0, respectively.

#### D. Sense and Restore

Because cell accesses are fundamentally destructive of data, each cell along the selected word line must have its charge restored. Because an access only causes a small perturbation of the bit line voltage, the difference must be amplified such that the cell voltage can be restored to its initial value. Each bit line,  $BL$ , has a corresponding “dummy” bit line,  $\overline{BL}$ , that serves as its data complement (see Figure 2). Like the main bit line,  $\overline{BL}$  is also pre-charged to  $V_{ref}$ . When an access occurs, only  $BL$  shares charge with the cell. During the sense and restore stage, a *sense amplifier* amplifies the differential voltage  $v_s$  between  $BL$  and  $\overline{BL}$ . The sense amplifier then holds the bit lines to their saturated values until the selected cell on each line is charged up (restored) to the initial value. The sense amplifier circuit is depicted in Figure 2 along with its input control signals  $SAP$  and  $SAN$ .

For example, if a cell that containing a 1 is accessed,  $BL$  goes to  $V_{ref}^+$  and  $\overline{BL}$  stays at  $V_{ref}$ . The sense amplifier, through the use of positive feedback, amplifies the difference between the two bitlines ( $v_s$ ).  $BL$  is pulled up to  $V_{DD}$  and  $\overline{BL}$  is pulled down to  $GND$ . Similarly, for accessing a cell containing a 0,  $BL$  is pulled down to  $GND$  and  $\overline{BL}$  is pulled up to  $V_{DD}$ .

After accessing, sensing, and restoring a row, it is open for input and output. Only one row within a bank may be open at any time. Once the cell restoration is complete, any combination of read and write operations can be performed on the open row. When all operations on the row are complete, the row must be closed by a pre-charge operation.

1) *Read*: A read operation may follow a sense and restore by simply multiplexing a (fully driven) column onto an input/output bus. When the bit lines are saturated and the accessed cells fully restored, the bit lines in the selected column can then be read out through an I/O bus to an output buffer. A simple control signal controls access from each bit line to the I/O bus (Figure 2). The other accessed cells on the word line that were not requested for a read do not place their data on the I/O bus lines. An example involving a read 1 operation follows, along with a graphical depiction in Figure 3:

- *Pre-charge*:  $Pre$  is asserted and the bit lines are ready for an access, with voltages equal to  $V_{ref}$ .
- *Access*:  $WL$  is asserted to  $V_{DD} + V_T$  to account for the threshold voltage of the NMOS cell transistor. Charge sharing occurs between  $C_{cell}$  and  $C_{BL}$ , bringing the voltage on both to  $V_{ref}^+$ . Other bit lines also undergo charge sharing.

- *Sense and Restore*: The sense amplifier pulls  $BL$  and  $C_{cell}$  to  $V_{DD}$  and  $\overline{BL}$  to  $GND$ . Other cells on the row are concurrently sensed and restored.
- *Read*: Once  $BL$  and  $\overline{BL}$  are stable,  $I/O\_enable$  is asserted and column data is transferred to the output buffer. Other reads or writes may follow on the open row.
- *Pre-charge*:  $Pre$  is asserted once again and the row is closed by a pre-charge operation.

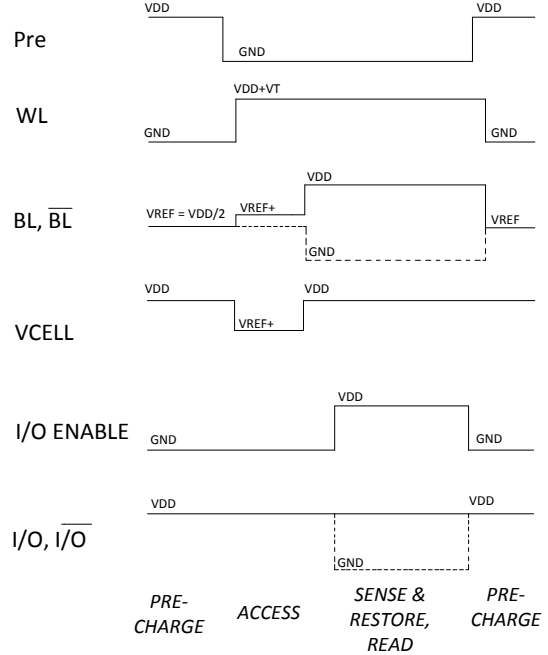


Figure 3. Read 1 Waveform [10] – Assume no rise/fall time

2) *Write*: Due to the nature of the DRAM array architecture, a write operation in DRAM is also preceded by a sense and restore operation on the selected row, because the non-requested cells must have their charges replenished [10]. After the sense and restore stage is complete,  $BL$  and the open cell are driven to the input voltage, and  $\overline{BL}$  to its complement, and the write of the new value is completed. Note that this may cause a bit line to be charged to  $V_{DD}$  or  $GND$  during the restoration phase, only to be forced to fully charge or discharge in the opposite direction during a write. An example involving a write 0 over 1 operation follows, along with a graphical depiction in Figure 4:

- *Pre-charge*:  $Pre$  is asserted and the bit lines are ready for an access, with voltages equal to  $V_{ref}$ .
- *Access*:  $WL$  is asserted to  $V_{DD} + V_T$  to account for the threshold voltage of the NMOS cell transistor. Charge sharing occurs between  $C_{cell}$  and  $C_{BL}$ , bringing the voltage on both to  $V_{ref}^+$ . Other bit lines also undergo charge sharing.
- *Sense and Restore*: The sense amplifier pulls  $BL$  and

$C_{cell}$  to  $V_{DD}$  and  $\overline{BL}$  to  $GND$ . Other cells on the row are concurrently sensed and restored.

- **Write:**  $I/O$  and  $\overline{I/O}$  are set to the input value and its complement, respectively.  $I/O\_enable$  is asserted.  $BL$  and  $C_{cell}$  are pulled down to  $GND$ , while  $\overline{BL}$  is pulled up to  $V_{DD}$ . Other cells in the column are simultaneously written with the input data. Other reads or writes may follow on the open row.
- **Pre-charge:**  $Pre$  is asserted once again and the row is closed by a pre-charge operation.

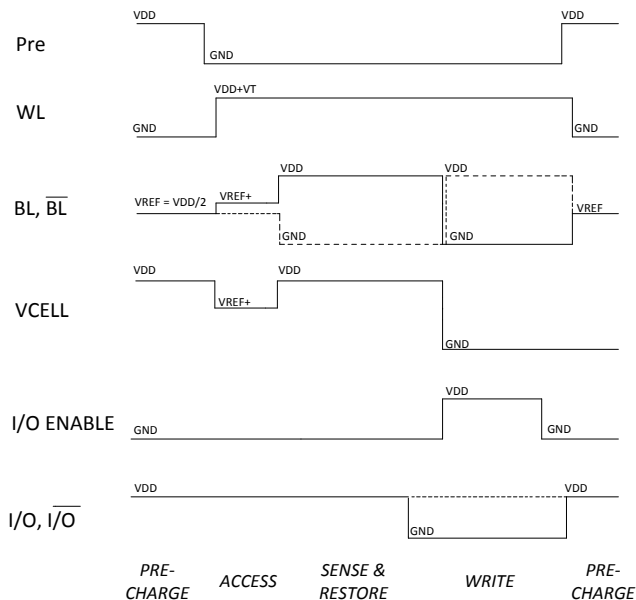


Figure 4. Write 0 over 1 Waveform [10] – Assume no rise/fall time

### E. Refresh Operation

The *refresh* operation is required to maintain data integrity over periods of inactivity. Because data is represented as charges on capacitors, they leak over time. Because of this, cells will eventually discharge and corrupt the original data. To counter this problem, the DRAM must be periodically refreshed at a rate sufficient to maintain data integrity in each cell in the device. A refresh operation is essentially implemented as an access, sense, and restore on each row in the array, and is detrimental to performance.

## III. TEST METHODOLOGY

### A. Memory Equipment

Our DIMMs were comprised of several models from four vendors (see Table I). The Vendor 1 DIMMs sourced their memory chips from Supplier 1 (with the possible exception of Model 4). The Vendor 3 modules sourced chips from Supplier 3, and the Vendor 2 used Supplier 2 parts. For some DIMMs, we could not identify the DRAM

chip suppliers (hence UNKNOWN in Table I). Most models were 1 GB DDR3 modules, rated at 1066MHz (except for the Vendor 4 models, rated for 1800 MHz) with a supply voltage of 1.5V. We also included three 2GB specimens to see if capacity had any direct effect on power consumption.

Table I  
DIMM SELECTION

ID	Manuf.	Model	Supplier
A	Vendor 1	Model 1 (1GB)	Supplier 1
B	Vendor 1	Model 1 (1GB)	Supplier 1
C	Vendor 1	Model 1 (1GB)	Supplier 1
D	Vendor 1	Model 2 (1GB)	Supplier 1
E	Vendor 1	Model 2 (1GB)	Supplier 1
F	Vendor 2	Model 1 (1GB)	Supplier 2
G	Vendor 2	Model 1 (1GB)	Supplier 2
H	Vendor 1	Model 4 (1GB)	UNKNOWN
I	Vendor 1	Model 4 (1GB)	UNKNOWN
J	Vendor 3	Model 1 (1GB)	Supplier 3
K	Vendor 3	Model 1 (1GB)	Supplier 3
L	Vendor 3	Model 1 (1GB)	Supplier 3
M	Vendor 4	Model 1 (1GB)	UNKNOWN
N	Vendor 4	Model 1 (1GB)	UNKNOWN
O	Vendor 4	Model 1 (1GB)	UNKNOWN
P	Vendor 3	Model 2 (1GB)	Supplier 3
P	Vendor 3	Model 2 (1GB)	Supplier 3
Q	Vendor 3	Model 2 (1GB)	Supplier 3
R	Vendor 1	Model 1 (1GB)	Supplier 1
S	Vendor 1	Model 1 (1GB)	Supplier 1
T	Vendor 1	Model 3 (2GB)	Supplier 1
U	Vendor 1	Model 3 (2GB)	Supplier 1
V	Vendor 1	Model 3 (2GB)	Supplier 1

### B. Test Platform & Data Acquisition

The test platform utilized a dual-core Atom D525 CPU running at 1.80 GHz. Only one DIMM was installed at a time on the motherboard, and all other hardware was identical for all tests. No peripherals were attached to the system except for a keyboard, VGA monitor, and a USB flash drive containing the custom test routines. Temperature was regulated within the chamber.

To measure power consumption of the DIMM, a  $2\Omega$  resistor was inserted between the module and the motherboard slot [11]. Using an Agilent 34411A digital multimeter, we sampled the voltage across the resistor at approximately 10 ksamples/sec, and wrote the data to files for post-processing with custom MATLAB scripts.

Because we required fine control over all memory I/Os on the testbed, we developed custom modifications to the Memtest86 v3.5b open source software, which is typically

used to diagnose memory faults [12]. The advantage of using this software as a foundation was the lack of any other processes or an operating system with virtual memory, which granted us the flexibility to utilize memory at a low level.

Version 3.5b was chosen because of several existing tests that could be modified to suit our needs. The primary test of interest was the Address Write/Read test, which iterated through the entire memory sequentially, writing each memory location with its own address, and then reading it back on the next pass to check for faults. However, in order to separate and control write and read operations individually, the test was split into two independent functions.

We created a write function which only wrote memory sequentially with a specified bit pattern, and never read it back. The code was derived from the optimized assembly code from the original authors, which eliminated any possibility of the C compiler diluting the experiment. The loop was written to maximize memory utilization and hence memory power consumption for write operations.

Similarly, a read function was created which only read memory sequentially, but never used it. Like the write function, it also utilized the optimized read loop assembly code from the original test to avoid C compiler effects. The loop maximized memory utilization and memory power consumption for read operations. Each word location in memory was initialized with an arbitrary pattern before the read process executed.

Lastly, the bit fade test – which is normally used to detect bit errors over a period of inactivity – was modified to serve as an idle power test, where absolutely no memory I/Os were issued from the software. This allowed for measurement of background power.

For all tests, the cache was enabled to allow for maximal memory bus utilization. With the cache disabled, we observed dramatically lower data throughput and were unable to distinguish power differences between operations. All tests were run on a single CPU core. Table II summarizes the important test environment parameters.

#### IV. DATA DEPENDENCE OF POWER CONSUMPTION

In order to construct a reasonable method of benchmarking DIMMs for power consumption, experiments were run to determine data and operation dependencies in power usage. The goal was to develop a process to measure DIMM write, read, and idle power independently to learn how power is used in memory at a high level, and what, if any, effects the

Table II  
TESTBED AND MEASUREMENT PARAMETERS

Parameter	Value
Testbed CPU	Intel Atom D525 @ 1.8 GHz
Number of CPU Cores Used	1
Cache Enabled	Yes
DIMM Capacities	1 GB, (2GB)
DIMM Operating Clock Freq.	400 MHz
Effective DDR3 Clock Freq.	800 MHz
DIMM Supply Voltage	1.5V
Primary Ambient Temp	30C
Secondary Ambient Temps	-50C, -30C, -10C, 10C, 40C, 50C
Memory Test Software	Modified Memtest86 v3.5b
Custom Test Routines	Seq. Write Pattern, Seq. Read, Idle
Digital Multimeter	Agilent 34411A
Sampling Frequency	10 ksamples/sec
Reading Range	100 mV
Reading Accuracy	approx. 0.06 mV for typ. reading
Number of Samples	200000

actual memory data had on the results. Figure 5 depicts the power signature of idle DIMM J for comparison. For write and read operations, there are six basic combinations of data I/O on an individual data cell:

- *Read 0*: Read a cell containing 0.
- *Read 1*: Read a cell containing 1.
- *Write 0 over 1*: Write 0 to a cell containing 1.
- *Write 0 over 0*: Write 0 to a cell containing 0.
- *Write 1 over 1*: Write 1 to a cell containing 1.
- *Write 1 over 0*: Write 1 to a cell containing 0.

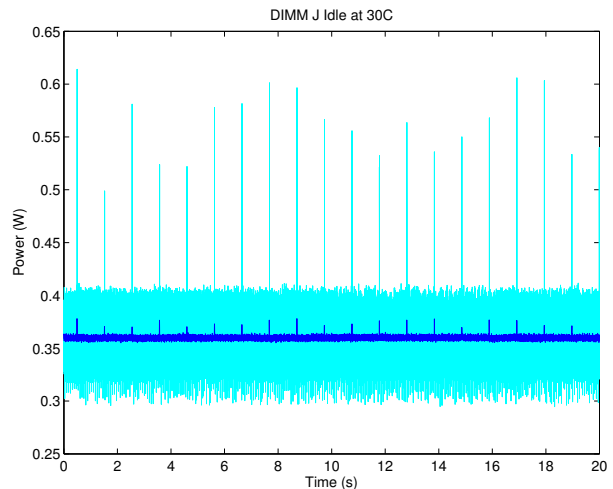


Figure 5. DIMM J Idle Power Waveform

Considering that the voltage waveforms for the bit lines and cells are dependent on the type of operation (read or write) as well as the data being read or written, we

hypothesized that the power consumption would vary between each of the above cases. Note that the background, pre-charge, and access power consumed in a DRAM should have no dependence on the data [13]. To observe how power varies between the above six cases, we ran measurements for each scenario. All six tests were performed on DIMM J (see Table I) at 30C. All trends noted in the subsections below were verified for other DIMMs by different vendors, suppliers, and models; only the data for DIMM J is described in particular. The spikes in the plots are due to program overhead between passes on the memory, and we ignore those deviations for simplicity. *Note: Any statements made about causes of the power variations in this section are purely speculation. These explanations are intended to provide background intuition on data and operation dependence of DRAM power consumption.*

### A. Read-Only Power Consumption

The first test continually read across the DIMM initialized to all 0s. The data was never overwritten after initialization and no computations were performed on the data. Figure 6 illustrates the processed power waveform. The average power consumed in this test was 0.475W. In all plots, the light blue lines are the raw data; the dark blue lines are moving averages.

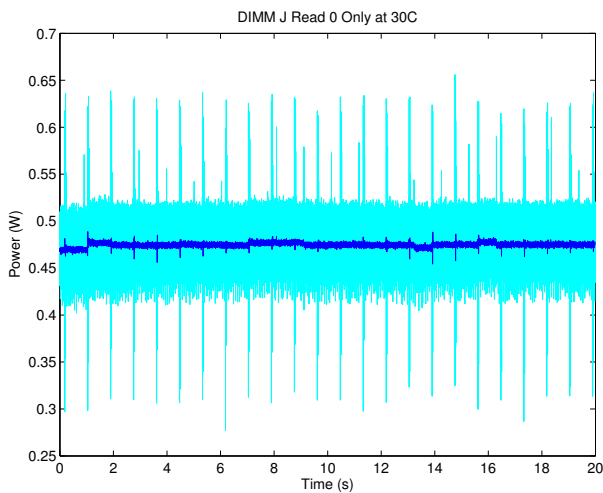


Figure 6. DIMM J Read 0 Power Waveform

The second test continually read the DIMM that was initialized to all 1s. The test was otherwise performed identically to the Read 0 test. Figure 7 depicts the power waveform for this experiment. The average power consumed in this test was 0.676W. Since the difference between the Read 0 test and the Read 1 test is the data in memory, we can attribute the additional power consumed to read 1s over

0s as purely a function of data.

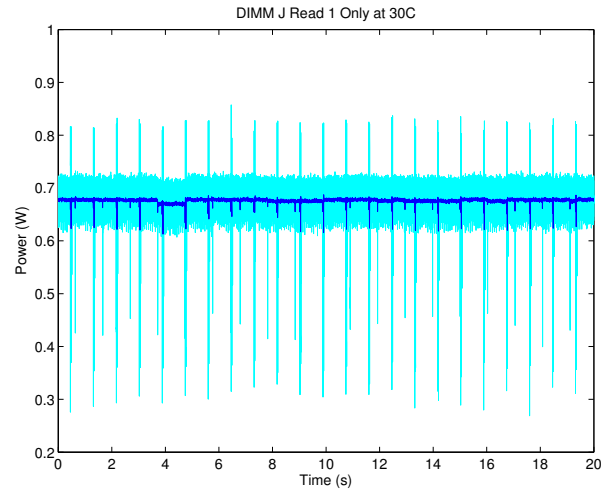


Figure 7. DIMM J Read 1 Power Waveform

To understand why this might be the case, consider the voltage waveforms for reading a 0 out of a DRAM cell. If we disregard any activity caused by pre-charge and access (because they should be independent of reading, writing, and data), and examine only positive voltage swings driven by supply current, the total positive swing on both bit lines (additive) is  $V_{ref}$ , while there is no positive swing on the cell capacitor. On the other hand, when one considers the voltage waveforms for reading a 1 out of a DRAM cell (see Figure 3), one finds by inspection that the total positive swings are  $V_{ref} - v_s$  on the bit lines (additive) and  $V_{ref} - v_s$  on the cell capacitor. Recall that during a sense and restore operation, each bit line pair on the entire row must experience these voltage patterns since each cell contains identical data (due to the intentional design of the test), whether it be all 0s or all 1s. Therefore, we can reasonably infer that the increase in average power between reading all 0s and reading all 1s is due to the additional current required to charge each cell capacitor to  $V_{DD}$  during a restore operation.

### B. Write-Only Power Consumption

There are four basic data scenarios in write operations. First, we will consider writing 0s over 0s across memory, and then writing 1s over 1s. This allows us to examine the difference in power consumed by read and write operations with the exact same data.

In the first case, writing 0s over 0s, the voltage waveforms in the array should be nearly identical to that of the read 0 case. The exception is that there is an additional delay induced after the sense and restore stage, when the write

data on the I/O lines must be transferred to the column of interest. However, because the data being written is the same as the data already on the bit lines and cells, there should be no additional array current penalty invoked over the read 0 case. Figure 8 displays the power waveform obtained by writing 0s over 0s continuously. The mean power was 1.102W. We speculate that the significant difference in DIMM power (0.627W) is caused by the DRAM peripheral circuitry used to handle data I/O, while the current consumed in the array should remain the same.

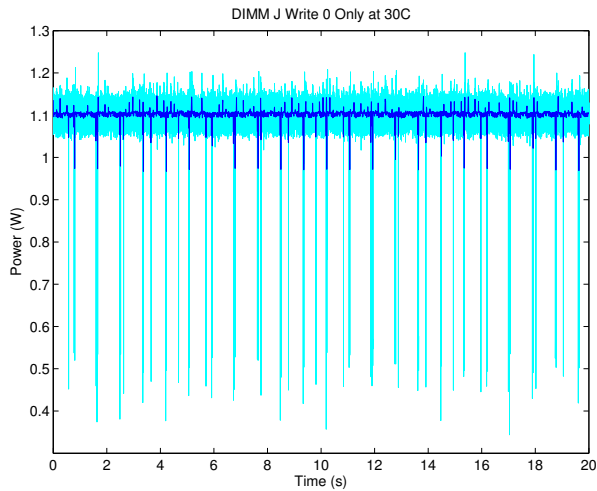


Figure 8. DIMM J Write 0 over 0 Power Waveform

Now consider writing 1s over 1s. Like the previous case, the array voltage waveforms should be nearly identical to the read 1 scenario, with any overall power difference likely being due to the peripheral and I/O circuitry on the DRAM devices. The power waveform for this test can be seen in Figure 9. The mean power was 1.199W. Note that the difference of 0.523W between this case and the read 1 case is less than the difference between writing 0s over 0s and reading 0s. It is not clear why this might be the case.

In order to test the remaining two cases of 0 over 1 as well as 1 over 0, we performed the write test with an alternating pattern of all 1s followed by all 0s, switching indefinitely. Thus in each pass over the memory, we will be covering one of the remaining cases. Figure 10 depicts the resulting waveform when used with the alternating write pattern. It is clear that there are two average power levels corresponding to each case.

Referring to Figure 4, writing 0s over 1s should theoretically draw more current in the array than the 0 over 0 case – because of a greater total positive voltage swing – and thus cause greater power consumption. However, the average power was *lower*, at 1.008W. This can be observed

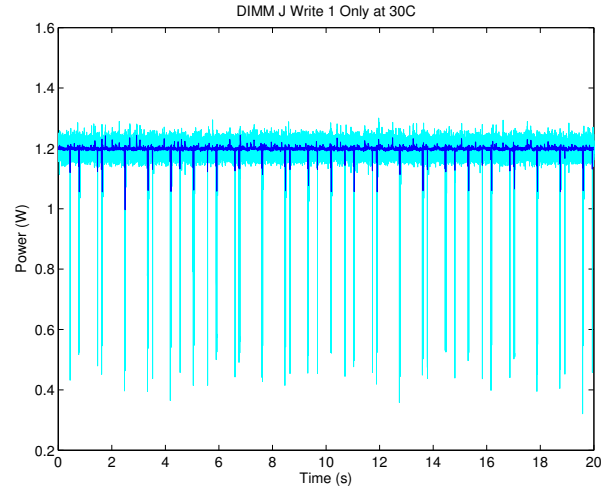


Figure 9. DIMM J Write 1 over 1 Power Waveform

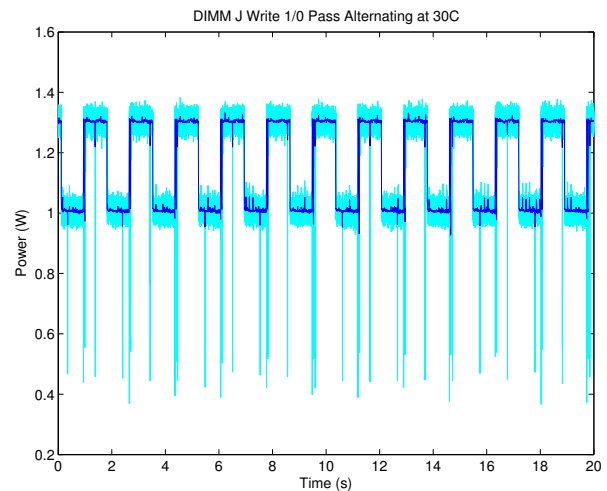


Figure 10. DIMM J Write 1/0 Alternating Each Pass Power Waveform

as the lower power threshold in Figure 10. It is not apparent why this is the case. We presume that the DRAMs might be using a special power-saving technique to conserve the initial charges on the cells to reduce the current draw required to pre-charge the bit lines.

The last case, writing 1s over 0s, should theoretically consume greater power than the 1s over 1s case, due to the necessity of fully discharging and charging the bit lines to overwrite the initial values. Our intuition is confirmed by Figure 10, where the upper bound of 1.305W is consumed when overwriting 0s with 1s.

The 0s over 1s case *decreased* the power consumption compared to 0s over 0s, whereas the 1s over 0s iterations *increased* the power compared to the 1s over 1s case, *both by the same delta of approximately 0.1W*. While we are



presently unable to explain all the data dependencies in write power consumption, it is clear that there are consistent and symmetric trends at play here, and we have verified our general hypothesis that DRAM power consumption is affected both by the type of operation (read or write) as well as the data being read or written.

## V. TEST RESULTS

### A. Choice of Memory Access Patterns

As described earlier in Section III, our Memtest86-based benchmark solution allowed us to write and read an arbitrary pattern to each memory location. Having found in Section IV that the power consumed in write and read operations is heavily data-dependent, we required data that would not be biased in favor of 1s nor 0s. We decided to use memory addresses as the data to write and read for all subsequent tests, because over the entire address space, there are approximately equal quantities of 1s and 0s. This scheme would produce power waveforms that averaged between all 1s and all 0s and also represented typical data that might be stored in memory. The same approach was applied to the three 2GB DIMMs (T, U, and V in Table I). Although the address space expanded compared to the 1GB models, resulting in longer durations per pass, this should not have any effect on instantaneous power consumption. *NOTE: Because the sample size of DIMM specimens was small, no statistical conclusions can be made about the memory population in general. All quantitative observations and inferences are merely about the specimens at hand, while statements about the population are speculative.*

### B. Temperature Effects

To determine if temperature had any effect on power consumption for write, read, or idle on a DIMM, we tested four DIMMs, one from each vendor: DIMM F, J, M, and R (see Table I). Each DIMM was tested for write, read, and idle power variations at ambient temperatures of -50C, -30C, -10C, 10C, 30C, 40C, and 50C. Testing above 50C was not practical as it caused hardware failure on the testbed.

A graph of the average write, read, and idle power for each of the four DIMMs at each temperature is depicted in Figure 11. It is clear that temperature had a negligible effect on power consumption even across a large temperature range. Closer views of the temperature dependencies for each operation are depicted in Figures 12, 13, and 14. Table IV summarizes the maximum relative variations per DIMM. Because no DIMM exhibited more than 3.61% variation across 100C of temperature range, all further tests were performed at an ambient temperature of 30C.

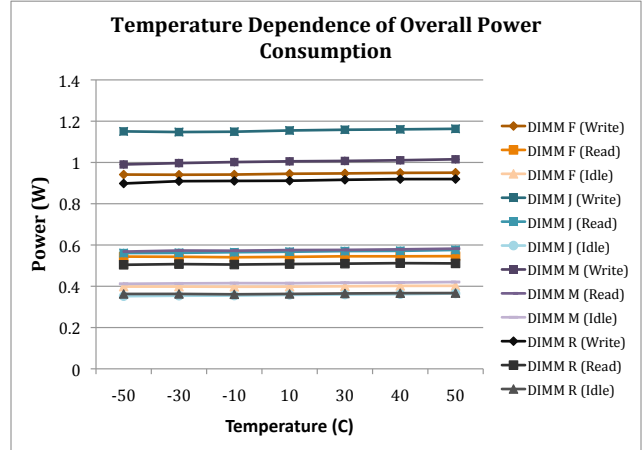


Figure 11. Temperature Dependence of Power Consumption

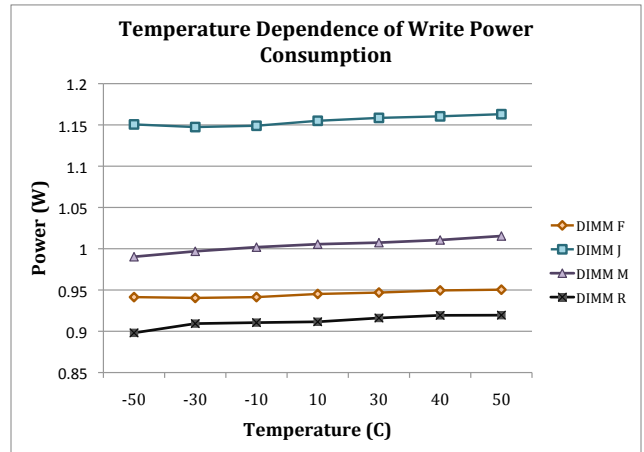


Figure 12. Temperature Dependence of Power Consumption - Write

### C. DIMM Power Variations

Having established negligible temperature dependence of DIMM power consumption, we performed the write, read, and idle tests for all the DIMMs in Table I at 30C. A plot of the resulting power numbers is depicted in Figure 15. Upon inspection, it appears that there was significant variation across all DIMMs, particularly between models and vendors. Furthermore, there seems to be a trend of power dependence on DIMM capacity.

1) *Variability Within DIMMs of the Same Model (1GB):* Consider V1S1M1 (Vendor 1, Supplier 1, Model 1), which has the largest number of specimens. There is moderate variation among this model, although most of this appears to be between two different groups. This may be because DIMMs are often matched in pairs when sold, and likely come from the same production batch. While there is a maximum of 12.29% difference between the five DIMMs of V1S1M1 (A, B, C, R, S), there is a visible gap between



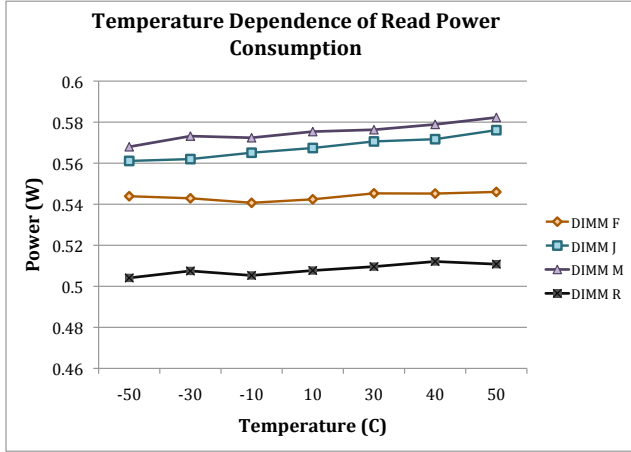


Figure 13. Temperature Dependence of Power Consumption - Read

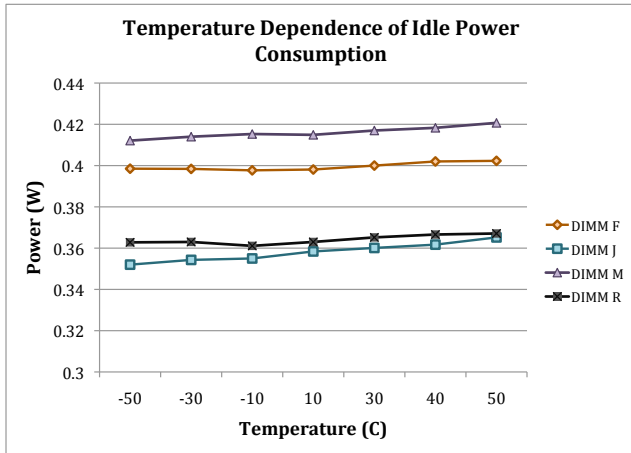


Figure 14. Temperature Dependence of Power Consumption - Idle

the DIMMs A, B, and C (batch 1) and the DIMMs R and S (batch 2). The maximum variation within batch 1 is only 1.34% for idle, and 1.47% between batch 2. This suggests that the majority of the variation in V1S1M1 is between the two batches.

2) *Variability Between Models of the Same Vendor/Supplier (1GB)*: Now consider all DIMMs from Vendor 1, which has the most DIMMs tested (naturally, the more samples there are in a group, the greater the likely maximum range of values). We would expect that there would be more variation in Vendor 1 than in V1S1M1 only, and this is confirmed in the data. The maximum variation observed in Vendor 1 (1GB) was 16.40% for the idle case.

3) *Variability Across Vendors (1GB)*: In order to isolate variability between vendors and all specimens of all vendors and to mitigate any effects of different sample sizes for each vendor, consider differences between the vendor

Table III  
RELATIVE VARIATIONS OF POWER CONSUMPTION BY OPERATION DUE TO TEMPERATURE

DIMM	Operation	Max % Power Variation from -50C to 50C
F	Write	1.05%
	Read	0.97%
	Idle	1.14%
J	Write	1.34%
	Read	2.60%
	Idle	3.61%
M	Write	2.47%
	Read	2.46%
	Idle	2.04%
R	Write	2.33%
	Read	1.56%
	Idle	1.63%

means for write, read, and idle (Figure 16). It is clear that Vendor 3 consumed the most write power at 1.157W, while read power was distributed more tightly between the four vendors. Indeed, this is confirmed with the variation for write being 17.73%, while read power variation was only 6.04%. Idle power came second with 14.65% spread.

4) *Effects of Capacity on Power Consumption*: It is clear from Figure 15 that the three 2GB DIMMs of V1S1M3 consume significantly more power than their 1GB V1S1M1 cousins, which are otherwise marked with identical model numbers. This is expected, as there is bound to be higher leakage power with twice as many DRAMs (in two ranks instead of one), as well as increased active power due to additional decoding overhead in the inactive rank. Indeed, the maximum variation between the 2GB and 1GB versions was 37.91%, which occurred for idle power, while write power only differed by 22.93%.

5) *Overall Variability Amongst 1GB DIMMs*: As one might have expected, the variations across all DIMMs were significantly higher than within models and vendors/suppliers, with the maximum variation occurring for write power at approximately 21.84%. Interestingly, while idle power tended to vary most amongst the other categories, only across vendor means and overall did write power variability dominate.

6) *Variability Summary*: Figure 17 highlights some major results of the investigation of power variability in memory that were discussed above.

## VI. DISCUSSION

Temperature had a small positive effect on memory power consumption (approximately 1-3% over -50C to

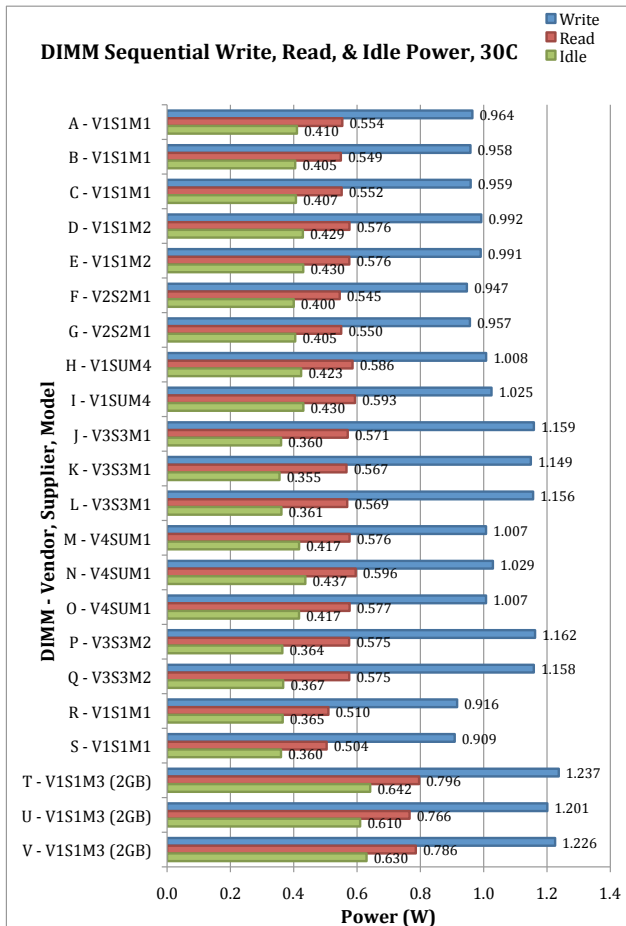


Figure 15. Write, Read, and Idle Power by DIMM, 30C

50C range). This may come as a surprise, as traditional integrated circuits tend to display strong temperature dependency of power consumption. This result might be explained by considering the DRAM architecture. In order to maximize array densities, it is desirable to keep cell access transistors small. This also has the benefit of reducing leakage current from the cells, in turn reducing the need for refresh cycles [1]. Compared to a conventional digital IC, which typically optimizes gate sizes for delay, DRAM does the opposite by optimizing for area and power. Because of the small transistors in much of the device, we suspect that leakage current does not constitute a large portion of power. Therefore, large temperature changes do not cause significant changes in DRAM power consumption.

For the specimen test within V1S1M1, we found that idle power showed the most variation (12.29%), followed by read, while write showed the greatest consistency. Note that the idle power variations were lesser in magnitude than those for write. The majority of these differences were between two batches of DIMMs, while the variability within

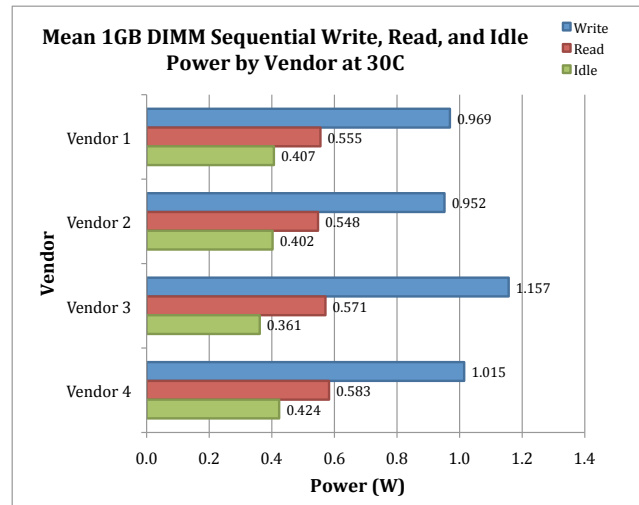


Figure 16. Mean Write, Read, and Idle Power by Vendor

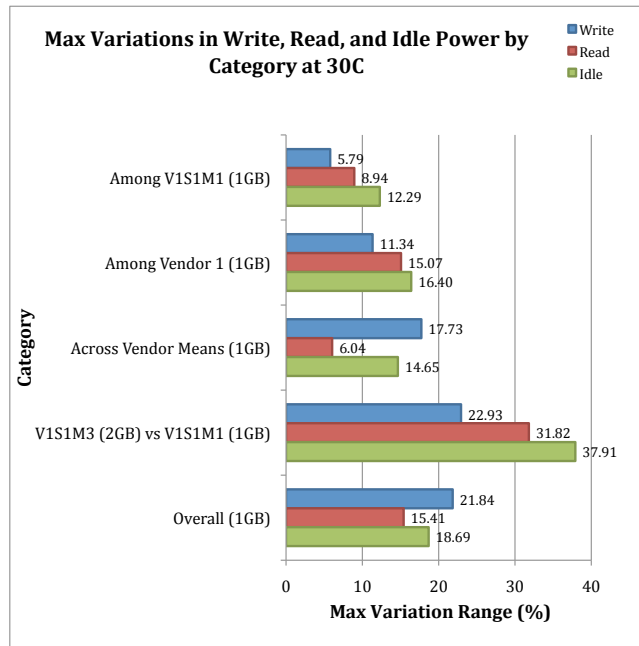


Figure 17. Highlights of Relative Variations by Category

each batch of 2 or 3 DIMMs was much smaller, on the order of 1-2%. Again, this should not come as a surprise, because memory modules are often sold as matched sets. It may be that the differences between batches are a result of temporal process variability or changes in fabrication methods. This might explain the larger deviations in idle power, because leakage current has a strong dependence on process variations [2]. However, because active power in DRAM is dominated by performance specifications and architecture, variations in leakage power would have less of an overall impact.

We can reasonably expect that increasing the sample size in any group would also increase the range of values. This expectation is supported by the results from the Vendor 1 (1GB) comparison – the relative variability increased in all three categories compared with the smaller V1S1M1 test. The extra sources of variation come from the other models included in the test.

In order to separate the variations as a function of vendor from the overall variability among 1GB models, we computed the ranges for the vendor *means*, and found up to 17.73% deviations in write power. Interestingly, read power was fairly consistent among all four vendors, at only 6.04% across the mean values. Idle power deviations were up to 14.65%.

We saw higher variability when looking at all 1GB specimens from all vendors individually, and found up to 21.84% deviations in write power, 18.69% in idle, and 15.41% in read. These values are higher than any amongst the V1S1M1 test, the Vendor 1 test, and across vendor means. Contrasting the overall results with the vendor means analysis, it appears that the differences between vendors account for most of the write and idle power variability overall, while it appears that overall read variability is more strongly a function of variability within vendors.

Lastly, we observed much higher deltas between the 2GB (V1S1M3) and 1GB (V1S1M1) models, which are from the same vendor, supplier, and possess the same model numbers. Up to 37.91% differences in idle power were observed between all of these specimens, 31.82% for read, and 22.93% for write. Contrasting these numbers with those of the V1S1M1 test, the majority of these deviations are due to the differences in capacity. This is not surprising; while the 1GB model had a single rank, the 2GB version had two ranks of otherwise identical DRAM devices. The most plausible explanation for this trend is that more DRAMs must be on with the 2GB device, serving as a positive offset in power consumption. However, when active power dominates, the differences between the 1GB and 2GB models decreases, because the base power offset consumes a smaller proportion of total power in the 2GB model.

The key result of this work is the overall variability between all vendors, suppliers, and 1GB models. In a server-type scenario, which must have large working sets and many processes, it is possible that many different DIMM models will be utilized. The large variability leaves room for an OS-level optimization to minimize power consumption in the memory system. Furthermore, a mixture

of 1GB and 2GB devices, which may also be common for desktop-class systems, could benefit from similar optimizations.

## VII. FUTURE WORK

It would be prudent to obtain a greater number of DIMMs to allow a more traditional statistical analysis of power variability, instead of looking only at maximum observed variations. More importantly, however, is the potential of applications to harness any inherent variability in a memory system. We found earlier that in our Atom-based test platform with two 1GB DIMMs, roughly 20% of the total CPU and memory power was consumed by the DIMMs when running the PARSEC Canneal benchmark. For other memory-bottlenecked or single-threaded applications, this proportion may increase. Regardless, in any system with significant memory power consumption, a variability-aware optimization can help reduce overall power usage. Such an optimization might be done in several ways. The best method might be to implement it in the kernel, which can make intelligent, flexible decisions about memory usage and does not require any hardware development.

The first task for the kernel would be to determine the memory power profiles to analyze any present variability. One approach might instrument the hardware, and run a suite of benchmarks once. However, due to the dynamic needs of a particular user and system, power profiles generated by this method may not be indicative of the actual usage patterns. An alternative approach could sample the power of each DIMM over a recent “window” of time, using the most recent information to make informed decisions. This method would offer increased flexibility, but would require constant hardware monitoring.

The second task for the kernel is to apply the known DIMM power information to the memory subsystem and determine on-the-fly where to grant allocations; the goal is to minimize power consumption without causing a performance penalty. Research has been done in the past that made power-aware allocation and page relocation decisions, but they did not consider variability. These existing methods might be adapted to make power and variability-aware allocations. One approach may “predict” the memory patterns of running processes based upon trends. Another might allocate memory for temporally adjacent scheduled processes on the same DIMM, such that the others can be put into the idle state. Further research is required in this area.

## VIII. CONCLUSION

We have analyzed the read, write, and idle power consumption of several mainstream DDR3 DIMMs from different vendors, DRAM suppliers, and models, and found several important trends. Firstly, we did not find significant variation in write, read, or idle power consumption as a function of temperature. We ran the remainder of tests at a constant ambient temperature of 30C. There was some power variation between specimens of the same model, and between DIMMs of the same vendor. More important was the trend of variations, with idle power generally displaying the most, followed by read and write power. However, a different trend was evident across the vendor means, with write power varying the most, followed by idle and read. This trend was dominant overall amongst all tested 1GB DIMMs, where we observed up to 21.84% variation in write power. Lastly, we found that specific 2GB models consumed significantly more power in all areas than their corresponding 1GB versions. While we avoided statistical inferences about the DRAM population due to small sample sizes, these findings serve as ample motivation for power and variability-aware kernel optimizations to reduce system power consumption. Further research should be done on methods of profiling memory power characteristics for the OS, and how the kernel should implement a variability-aware solution without negatively impacting system performance and reliability.

## ACKNOWLEDGMENT

Thank you to Professor Puneet Gupta and Abde Ali Kagalwalla in the UCLA NanoCAD group for their guidance in this work. Further thanks to Lucas Wanner of the UCLA Networked and Embedded Systems Laboratory (NESL) for kindly aiding me in instrumentation. Lastly, thank you to the National Science Foundation (NSF) for funding my research.

## REFERENCES

- [1] J. M. Rabaey, A. Chandrakasan, and N. Borivoje, "Designing memory and array structures," in *Digital Integrated Circuits - A Design Perspective*, 2nd ed. Pearson Education, Inc., 2003, pp. 623–717.
- [2] S. Borkar, "Designing reliable systems from unreliable components: the challenges of transistor variability and degradation," *Micro, IEEE*, vol. 25, no. 6, p. 1016, 2005.
- [3] K. Rajamani, C. Lefurgy, S. Ghiasi, J. C. Rubio, H. Hanson, and T. Keller, "Power management for computer systems and datacenters," in *International Symposium on Low Power Electronics and Design tutorial*, <http://www.islped.org/X>, 2008.
- [4] C. Bienia, "Benchmarking modern multiprocessors," Ph.D. dissertation, Princeton University, 2011.
- [5] M. Bhadauria, V. M. Weaver, and S. A. McKee, "Understanding PARSEC performance on contemporary CMPs," 2009.
- [6] "The PARSEC benchmark suite." [Online]. Available: <http://parsec.cs.princeton.edu/>
- [7] V. Delaluz, A. Sivasubramaniam, M. Kandemir, N. Vijaykrishnan, and M. J. Irwin, "Scheduler-based DRAM energy management," in *Proceedings of the 39th annual Design Automation Conference*, 2002, p. 697702.
- [8] E. Cooper-Balis and B. Jacob, "Fine-Grained activation for power reduction in DRAM," *Micro, IEEE*, vol. 30, no. 3, pp. 34–47, 2010.
- [9] D. T. Wang, "Modern dram memory systems: performance analysis and scheduling algorithm," 2005.
- [10] K. Itoh, *VLSI Memory Chip Design*, 1st ed. Springer, Apr. 2001.
- [11] A. A. Kagalwalla, "Software approaches to coping with variability in memory banks."
- [12] "Memtest86 test algorithms," <http://memtest86.com/>. [Online]. Available: <http://memtest86.com/>
- [13] "Calculating memory system power for DDR3." [Online]. Available: [download.micron.com](http://download.micron.com)