Towards Variation-Aware System-Level Power Estimation of DRAMs: An Empirical Approach

Karthik Chandrasekar
Christian Weis
Benny Akesson
Norbert Wehn
Kees Goossens

CISTER-TR-130402
Version:
Date: 04-08-2013
Towards Variation-Aware System-Level Power Estimation of DRAMs: An Empirical Approach

Karthik Chandrasekar, Christian Weis, Benny Akesson, Norbert Wehn, Kees Goossens

CISTER Research Unit
Polytechnic Institute of Porto (ISEP-IPP)
Rua Dr. António Bernardino de Almeida, 431
4200-072 Porto
Portugal
Tel.: +351.22.8340509, Fax: +351.22.8340509
E-mail: http://www.cister.isep.ipp.pt

Abstract

DRAM vendors provide pessimistic current measures in memory datasheets to account for worst-case impact of process variations and to improve their production yield, leading to unrealistic power consumption estimates. In this paper, we first demonstrate the possible effects of process variations on DRAM performance and power consumption by performing Monte-Carlo simulations on a detailed DRAM cross-section. We then propose a methodology to empirically determine the actual impact for any given DRAM memory by assessing its performance characteristics during the DRAM calibration phase at system boot-time, thereby enabling its optimal use at runtime. We further employ our analysis on Micron's 2Gb DDR3-1600-x16 memory and show considerable overestimation in the datasheet measures and the energy estimates (up to 28%), by using realistic current measures for a set of MediaBench applications.
Towards Variation-Aware System-Level Power Estimation of DRAMs: An Empirical Approach

Karthik Chandrasekar1, Christian Weis2, Benny Akesson3, Norbert Wehn2, Kees Goossens4
1Computer Engineering, TU Delft, The Netherlands
2Microelectronic Systems Design, TU Kaiserslautern, Germany
3CISTER-ISEP Research Centre, Polytechnic Institute of Porto, Portugal
4Electronic Systems Group, TU Eindhoven, The Netherlands

ABSTRACT

DRAM vendors provide pessimistic current measures in memory datasheets to account for worst-case impact of process variations and to improve their production yield, leading to unrealistic power consumption estimates. In this paper, we first demonstrate the possible effects of process variations on DRAM performance and power consumption by performing Monte-Carlo simulations on a detailed DRAM cross-section. We then propose a methodology to empirically determine the actual impact for any given DRAM memory by assessing its performance characteristics during the DRAM calibration phase at system boot-time, thereby enabling its optimal use at run-time. We further employ our analysis on Micron’s 2Gb DDR3-1600-x16 memory and show considerable over-estimation in the datasheet measures and the energy estimates (up to 28%), by using realistic current measures for a set of MediaBench applications.

1. INTRODUCTION

DRAM memories account for a significant share of any system’s power and energy consumption, be it battery-driven mobile devices [1] or high-performance computing servers [2]. With system power and energy budgets getting tighter, it becomes absolutely essential to employ highly accurate power models and energy estimates for every component in the system, including processors and DRAMs. Unfortunately, with the impact of process variations [3, 4] on power consumption scaling significantly at technologies below 90nm, existing power models are becoming less and less accurate, while worst-case power estimates are just too pessimistic to use. Hence, it has become imperative to estimate the impact of process variations on power consumption, for all system components, for an accurate system power analysis.

In the case of processors, many solutions have been proposed, both by vendors and academia that estimate [5, 6] and even help mitigate [7, 8], the expected performance and power impact. However, when it comes to DRAMs, vendors merely sort the memories into discrete speed-bins and furnish one set of worst-case current measures per speed-bin in datasheets, leading to over-estimation of DRAM power consumption. With DRAM memories becoming increasingly prominent in a system’s power/energy profile, employing such worst-case datasheet measures leads to unrealistic over-dimensioning of the system. This calls for variation-aware DRAM power-estimation methodologies that address the pessimism in the datasheets and improve the accuracy of the power models and energy estimates.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. DAC ’13, May 29 - June 07 2013, Austin, TX, USA.
Copyright 2013 ACM 978-1-4503-2071-9/13/05 ...$15.00.

Figure 1 shows the impact of process-variation observed by a memory vendor in the production analysis of a lot of 11,000 DDR3 1Gb memories with 533MHz frequency and x8 width, manufactured at 70nm, in batch U6PN8XBS-13G3.
Using these mechanisms, we derive possible current distributions for DRAM memories of any configuration, and also determine the actual performance measures and realistic current measures for a given DRAM memory using the characterization step at system boot-time. The derived performance measures can be used to improve the performance of the given DRAM memory and the realistic current measures can be employed in place of the worst-case datasheet values to obtain variation-aware DRAM power and energy estimates. We evaluate our proposed solution by deriving the current distributions for Micron’s 2Gb DDR3-1600-x16 memories [10] based on the the Monte-Carlo analysis, and employing them with a system-level power model [20,23] to show significant differences between typical and worst-case datasheet current measures and the corresponding energy estimates (up to 28%) for four MediaBench applications [33].

2. RELATED WORK

When it comes to studying the impact of process variation on power consumption in DRAMs, Intel observed performance degradation and power variation in DRAM memories in [11,12] and suggested performance throttling to maintain an average power budget assuming datasheet estimates in [15], as a work-around to this problem. M. Gottscho et al. in [13] also observed variations of around 15% in power consumption across several 1GB DIMMs from the same vendor and around 20% across different vendors, although they did not establish the causes for the observed extent of variations. L. Bathen et al. in [16,17] employed these observations by [13], and suggested memory mapping and partitioning solutions to exploit this variability. S. Desai et al. in [14] on the other hand, performed Monte-Carlo analysis on a single DRAM cell and basic circuit components and together with interconnect delay models estimated the variation impact for an entire DRAM memory. They further proposed using adaptive body biasing to improve the yield of DRAMs. Although the variation estimates may be acceptable for the basic circuit components, such an extrapolation to an entire DRAM is at best a coarse approximation. Unfortunately, there are no known realistic models or studies that provide acceptable estimates of the expected impact of process-variations on DRAM power consumption and no variation data is made available by DRAM vendors, undermining the applicability of the solutions suggested in [14–17].

In this work, we derive realistic estimates of the impact of variations on DRAM power currents and we observe the timings for different DRAM operations and verify the functional correctness of our design. We then perform Monte-Carlo analysis [27] on this cross-section to derive the variation-impact in DRAMs.

3. DRAM MODELING AND ANALYSIS

In this section, we first describe the baseline DRAM cross-section model to be used for our NGSPICE simulations. In these simulations, we observe the timings for different DRAM operations and verify the functional correctness of our design. We then perform Monte-Carlo analysis [27] on this cross-section to derive the variation-impact in DRAMs.

3.1 Baseline DRAM Cross-Section Model

The basic DRAM cell is modeled as a transistor-capacitor (1T1C) pair and stores a single bit of data in the capacitor as a charge. As shown in Figure 2, the transistor is controlled by a local wordline (lw) at its gate, which connects the capacitor to the local bitline (lb) when turned on (activated). Before reading the data from the memory cell, the bitlines in the memory array are precharged (set to halfway voltage level) using an equalization circuit. When connected, the cell capacitors change the precharged (PRE) voltage levels on the bitlines very slightly. Hence, a set of primary sense amplifiers (PSA) (or row buffer) distributed across memory sub-arrays are used to detect the minute changes and pull the active bitline voltage all the way to logic level 0 or 1. Once the bitline voltage is amplified, it also recharges the active bitline voltage all the way to logic level 0 or 1. The data is then switched via master datalines from the PSA to the secondary sense amplifiers (SSA), which connects to the I/O buffers. Once finished, the wordlines can be switched off, safely restoring the charge in the memory cells, before starting to precharge (PRE) the bitlines again.

The memory arrays are organized in a hierarchical structure of memory sub-arrays for efficient wiring. A memory sub-array consists of 256k cells connecting up to 512 cells per local bitline and per local wordline. 16 memory sub-arrays connect to one master wordline forming 4Mb blocks. The data is then switched via master datalines from the PSA to the secondary sense amplifiers (SSA), which connects to the I/O buffers. Once finished, the wordlines can be switched off, safely restoring the charge in the memory cells, before starting to precharge (PRE) the bitlines again.

The memory arrays are organized in a hierarchical structure of memory sub-arrays for efficient wiring. A memory sub-array consists of 256k cells connecting up to 512 cells per local bitline and per local wordline. 16 memory sub-arrays connect to one master wordline forming 4Mb blocks. The data is then switched via master datalines from the PSA to the secondary sense amplifiers (SSA), which connects to the I/O buffers. Once finished, the wordlines can be switched off, safely restoring the charge in the memory cells, before starting to precharge (PRE) the bitlines again.

The memory arrays are organized in a hierarchical structure of memory sub-arrays for efficient wiring. A memory sub-array consists of 256k cells connecting up to 512 cells per local bitline and per local wordline. 16 memory sub-arrays connect to one master wordline forming 4Mb blocks. The data is then switched via master datalines from the PSA to the secondary sense amplifiers (SSA), which connects to the I/O buffers. Once finished, the wordlines can be switched off, safely restoring the charge in the memory cells, before starting to precharge (PRE) the bitlines again.

The memory arrays are organized in a hierarchical structure of memory sub-arrays for efficient wiring. A memory sub-array consists of 256k cells connecting up to 512 cells per local bitline and per local wordline. 16 memory sub-arrays connect to one master wordline forming 4Mb blocks. The data is then switched via master datalines from the PSA to the secondary sense amplifiers (SSA), which connects to the I/O buffers. Once finished, the wordlines can be switched off, safely restoring the charge in the memory cells, before starting to precharge (PRE) the bitlines again.
3.2 DRAM Cross-Section SPICE Simulations

In our NGSPICE [24] modeling of the DRAM cross-section, we employed BSIM [27] model cards built on Low Power Predictive Technology models (LP-PTM) [28], since there are no openly available technology libraries specific to DRAMs. As a result, the LP-PTM devices had to be adapted appropriately, to ensure functional and timing correctness of the simulated DRAM cross-section.

We modeled the memory cell architecture (of 6T² area), the equalization circuit, the wordline driver, and the sense amplifier using the designs suggested in [18], [25] and [26]. The baseline DRAM configuration targets a 1Gb DDR3-1066 (533MHz) x8 memory with core timings of 7-7-7 cc (refer Section 4.1) at 45nm. We chose 45nm, since it is the common technology node employed by vendors for DDR3 memories including Samsung, Micron and Hynix. To verify our DRAM cross-section, we present the timings and voltages of the different signals [26] corresponding to basic DRAM operations (ACT-RD-PRE) in Figure 3.

![Figure 3: Behavior of DRAM Cross-Section](image)

As depicted in the figure, first the equalization circuit (eql) forces both the true (active) bitline (lblt) and the complementary (reference) bitline (lblr) to the same reference voltage (0.55V). This is followed by the local wordline (lw) going high to begin the activation process that switches the relevant transistors on and connects the cell capacitors to the corresponding local bitlines. Simultaneously, the equalization circuit (eql) de-activates to enable sensing of the change in bitline voltage due to the charge transfer. As the wordline high reaches the required voltage of 2.8V at around 5ns, the pre-sensing phase begins to create a minimum voltage difference (around 200mV) between the reference (lblr) and active (lblt) bitlines at the PSA. This is followed by the activation of the sensing circuit by N-Set and P-Set control signals (pset, nset) and the gradual drop in voltage level is reflected by the current drawn from the NMOS components of the PSA (lblr) and the gradual drop in voltage level of master dataline complement signal (depicted by mdqc). Once the mdqc drops by around 200mV (in relation to the core voltage), the data is sensed at the SSA at around 20ns. Once the read operation finishes, (data received by SSA) the mdqc (master dataline complement) is precharged back to its reference voltage (1.1V) at around 24ns and the local wordline is switched off at around 28ns. After a short delay to close the transistor and avoid destroying the charge in the cell, the sensing circuit in PSA is deactivated and the bitline equalization re-starts at around 33ns. This precharges both the local bitlines back to reference voltage levels, finishing at 50ns, as expected for a DDR3-1066 memory [9].

Besides the basic ACT-RD-PRE operations depicted in this figure, we also modeled the write and refresh operations in a similar manner and observed accurate functionality and timing [26], thus, verifying our modeling of the DRAM cross-section. We employ this DRAM cross-section to perform Monte-Carlo analysis to observe the impact of variation on delay and power consumption in the next section.

3.3 Baseline Monte-Carlo Analysis

In this section, we present the results from Monte-Carlo analysis on our verified 1Gb DDR3-1066 x8 DRAM cross-section, described in Section 3.2. Towards this, we vary global device parameters such as channel length, channel mobility, and oxide thickness and the local device threshold voltage (Vth) (primarily the variations in line edge roughness (LER) [31]), besides the interconnect parameters including wire width and wire thickness, within predefined variation ranges. We obtained the variability ranges (scaling metric (σ) in the corresponding Gaussian distributions) for these parameters from the ITRS technology requirements on Design for Manufacturability [29] and Modeling and Simulation [30] and the variation models of transistors [31,32]. We also introduce spatial-correlations in the variations among neighboring transistors, due to expected similarity in the parametric variations. Using these variability values, we performed Monte-Carlo runs on 1000 circuit instances reflecting the variations in all the device and interconnect parameters. From our observations, the variation in the device Vth parameter had the biggest impact on the circuit delay and current consumption [31], since it is also directly influenced by the variations in the global device parameters. As expected, the active (dynamic) DRAM currents and frequency increased linearly, while the leakage currents increased exponentially against the variations in the Vth parameter [34,35]. Hence, we analyzed the variations in leakage currents on the natural logarithmic scale [34] to obtain the σ values of their distributions corresponding to those of the Vth parameter.

The variations in the local and global device parameters at 45nm based on [29-32], as used in our simulations are presented in Table 1. These measures correspond to the variability introduced in the device parameters per σ change in their Gaussian distributions. In the table, the σ% value gives the relative variation to the nominal values (μ) obtained from the PTM models [28], while the σ values correspond to the absolute values of variation.

<table>
<thead>
<tr>
<th>Tech (nm)</th>
<th>Mobility (σ%)</th>
<th>Vth (LER (σ))</th>
<th>Length (σ)</th>
<th>Tce (σ)</th>
</tr>
</thead>
<tbody>
<tr>
<td>45</td>
<td>8.2</td>
<td>3.0e-9/√(w×l)</td>
<td>45e-9×0.03</td>
<td>1.67</td>
</tr>
</tbody>
</table>

Table 1: Transistor Process Parameter Variations

Amongst the different characteristic DRAM currents, we identify the dynamic (active and background) currents as \(I_{DDQ}, I_{DDQ}^{SP}, I_{DDQ}^{CQ}, I_{DDQ}^{DQ}, I_{DDQ}^{DA}, I_{DDQ}^{DAW}, I_{DDQ}^{D}, I_{DDQ}^{DA}, I_{DDQ}^{CQ}\), and the static (leakage) currents as \(I_{DDQ}^{SP}\). The different DRAM currents are described in detail in Section A1 and in [9,10].

In Table 2, we show the impact of process-variation on the different currents for a baseline 1Gb DDR3-1066 (533MHz) x8 DRAM memory. In this table, we present the nominal measures along with the 1σ, 2σ and 5σ estimates in the different \(I_{DDQ}\) currents. We also provide σ% value to get relative variation for the different current measures.
4. DRAM MEMORY CHARACTERIZATION

In this section, we relate a DRAM’s actual performance and current measures to the impact of process variations. Towards this, we first begin by reviewing the process of speed-binning in DRAMs in Section 4.1. We then propose a methodology to determine the frequency and core-timings of a particular DRAM memory and to relate them to the corresponding current measures from the distributions derived in Table 2. Towards this, we derive a performance metric, Functional Speed (FS), defined as the product of the sum of the memory’s core-timings (CL+RCD+RP) and the corresponding clock period ($1/F_{MAX}$), to represent both these performance parameters. (Lower the FS, faster the memory and higher the performance.) The goal of this proposed methodology is two-fold: (1) to derive the fastest overall DRAM functional speed ($MaxFS$ or lowest common FS) based on core timings and $F_{MAX}$, at which the entire DRAM can function, to improve the DRAM’s performance and (2) to derive the fastest individual bank functional speed ($MinFS$ or lowest individual bank FS) at which any individual DRAM bank may function, to conservatively identify the actual worst-case current measures, by relating this FS to the current distributions obtained in Table 2. The relation between the delays and currents is also shown in Section A5.

In Algorithm 1, we present this methodology to derive the overall DRAM and individual bank functional speeds. We propose to employ this algorithm once during the memory’s calibration phase [9] at system boot-time. Currently, this calibration phase in DRAMs is employed for timing synchronization and skew corrections in DRAM signals, to enable proper DRAM functionality. We propose to merely add a performance assessment step to this phase, to obtain realistic performance and current measures for use at run-time.

Algorithm 1 Frequency and Variation Estimation

Require: var_check($F_{LB}$, $F_{UB}$)
1: {Comment: $\sum CT = [n_{CL} + n_{RCD} + n_{RP}]$}
2: {# Define: $CT_{Min} = (5,5,5)$ and $CT_{Max} = (8,8,8)$}
3: {# Define: $F_{3,\sigma} = (F_{UB} - F_{LB})/12$ Here: $F_3\sigma=11MHz$}
4: {# Define: Banks = 8}
5: {# Define: $CT_{Max} = \{CT_{Max}[i]\}$ [Initialized]}
6: {for $i = 0 \rightarrow 2$ do}
7: {# Comment: Representing CL, RCD and RP}
8: {for $j = 0 \rightarrow 2$ do}
9: {# Comment: Representing CT range 8cc to 5cc}
10: {# Comment: Checking all frequency levels}
11: {# for $k = 0 \rightarrow Banks$ do}
12: {# Comment: Iterating over all 8 banks}
13: {for $f = F_{LB} \rightarrow F_{UB}$ do}
14: {# Comment: Checking all frequency levels}
15: {if CT_check ($k$, $f$, $CT^*$) == True then}
16: {# FS_Bank[i][f][k] = $\sum CT \times 1/f$}
17: {# Comment: Store corresponding f and CT*}
18: {else}
19: {# Break;}
20: {end if}
21: {end for($f$)}
22: {# Comment: Stores least FS for bank k for set CT*}
23: {end for($k$)}
24: {end for($i$)}
25: {end for($j$)}
26: {end for($i$)}
27: {# MinFS = min ($FS_Bank[i][f][k]$)}
28: {# Comment: Return corresponding f and CT*}
29: {MaxFS = max ($FS_Bank[i][f][k]$)}
30: {# Comment: Return corresponding f and CT*}

In this algorithm, we begin by identifying the fastest and slowest core-timings (in clock cycles) in a speed-bin (say DDR3-800) at the upper frequency bound of this speed-bin (532MHz). We then propose to start with the slowest set of

4.1 Variation and DRAM Speed-Binning

DRAM memories manufactured in a particular generation are down-binned into predefined speed-bins based on their minimum guaranteed frequency and memories within these speed-bins are classified according to their core-timings [9]. Table 3 presents the speed-bins and the core-timings in clock cycles (cc) used to classify Micron’s DDR3 memories [10].

Table 3: Micron Speed-Bins and Core-Timings

<table>
<thead>
<tr>
<th>Speed Bin (MHz)</th>
<th>Freq (MHz)</th>
<th>Fast Core (cc)</th>
<th>Slow Core (cc)</th>
</tr>
</thead>
<tbody>
<tr>
<td>800</td>
<td>400</td>
<td>5-5-5(12.5ns)</td>
<td>6-6-6(15ns)</td>
</tr>
<tr>
<td>1066</td>
<td>533</td>
<td>7-7-7(13.125ns)</td>
<td>8-8-8(15ns)</td>
</tr>
<tr>
<td>1333</td>
<td>666</td>
<td>9-9-9(13.5ns)</td>
<td>10-10-10(15ns)</td>
</tr>
<tr>
<td>1600</td>
<td>800</td>
<td>10-10-10(12.5ns)</td>
<td>11-11-11(13.75ns)</td>
</tr>
</tbody>
</table>

As can be noticed from the table, the +5σ estimate is significantly higher than the nominal (μ) values for the different $I_{DD}$ currents. In Sections A2 and A5, we present the impact of variation on the timing and power consumption corresponding to ±1σ variations in device and interconnect parameters, as observed from the 1000 Monte-Carlo runs.
memories [8-8-8] (15ns at 532MHz) (in Step 5) and reduce one core-timing parameter (say \( r_{t_{CL}} \)) by 1cc (Step 10), while maintaining the others at 8cc and increasing the memory frequency in steps along the 13 frequency values in steps of \( \Delta f \sigma \) (here 11MHz between 400MHz and 532MHz, given by Step 13). With these new core-timing settings, we propose to execute core-timings check (CT_check), which is based on JEDEC's \( I_{DD1} \) Measurement-Loop test [9] (described in Section A3) over all the 8 banks in the memory (Steps 11 and 15). This CT_check comprehensively checks the activation, reading and precharging operations on a given bank in different rows, which tests all the important DRAM timings [9]. If the test completes without any errors, the frequency is increased by another step \( F_{\sigma} \) (Step 18). If not, the last explored working frequency gives the lowest FS value for that bank for the selected set of core-timings (Step 16). We store this lowest FS value and the corresponding \( F_{MAX} \) and core-timings for reference. Steps 13 to 22 are repeated for all the banks and the lowest FS values are obtained for all the banks with the selected set of core-timings. Now we reduce the considered \( r_{t_{CL}} \) parameter further by 1cc and the tests are repeated with the new set of core-timings and the corresponding lowest FS values are noted, till the minimum functional \( r_{t_{CL}} \) value is reached. The same procedure is then employed with the other core-timing parameters (\( n_{t_{RCD}} \) and \( n_{pg} \)), assuming the fastest \( r_{t_{CL}} \) value at which the memory continued to work. All the corresponding lowest FS values for the different banks and set of core-timings are stored. Finally, the lowest FS value obtained for any of the DRAM banks (Min_FS) is used to conservatively identify the actual worst-case current measures of the memory and the maximum of the lowest FS values supported by all banks of the memory (Max_FS) and the corresponding core-timings and \( F_{MAX} \) are used to identify the new performance parameters.

Using the current distributions derived in Table 2, and the distribution of the functional speeds observed in Algorithm 1, in Figure 4, we overlap the two distributions to obtain the complete performance-power-variation relation in the \( \pm 6\sigma \) form. Here, the FS range for DDR3-800 is identified between 28.2ns (fastest memories (\( \sum CT=16 \)) at 532MHz and 45ns (slowest memories (\( \sum CT=18 \)) at 400 MHz. The datasheet current measures are identified at \( +5\sigma \) position at 29.6ns (fastest) and the datasheet performance is identified at \(-6\sigma\) position at 45ns (slowest). An example of the new current measures is highlighted at Max_FS - 1\( \sigma \) position (fastest overall DRAM FS) at 41ns and an example of realistic current measures is highlighted at Min_FS + 1\( \sigma \) position (fastest individual bank FS) at 34ns.

As shown in the results, when increasing the frequency from 533MHz to 800MHz, all currents except the leakage currents scale up linearly due to their dependency on the clock. When increasing the memory density, all currents scale up linearly due to the doubling of the number of memory cells and primary sense amplifiers. However, when the data-width is doubled, while retaining the same page-size (1KB), only the currents reflecting data transfer, viz., \( I_{DD1} \), \( I_{DD4} \) and \( I_{DD24} \) are affected, since only the number of data bits accessed during the column accesses increases.

5.2 Reverse Engineering Datasheet Values

In Section 5.1, we presented the impact of three system parameters on DRAM currents. However, since more than one parameter can be different between two DRAM memories, to estimate this impact, one should combine the influence of each of the concerned system parameters, from the observations in Table 4. We present the impact of all possible combinations in the appendix Section A4, since these are merely derived from the results in Table 4.

When applying this analysis on a 2Gb DDR3-800MHz x16 memory from Micron [10], all the three system parameters change at once. Accordingly, we estimate the possible current distributions in Table 5. As observed from the results in Table 5, the nominal estimates for the active currents are up to 30% lower (for \( I_{DDAN} \)) than the datasheet (DS) measures, while those for the leakage currents are up to 86% lower (for \( I_{DD} \)). These large differences in the current measures highlight the pessimism in the datasheets.

### Table 4: System Parameters Vs. Current Measures

<table>
<thead>
<tr>
<th>Config</th>
<th>Freq</th>
<th>Capacity</th>
<th>Width</th>
</tr>
</thead>
<tbody>
<tr>
<td>DDR3-800MHz</td>
<td>532MHz</td>
<td>2Gb-533MHz-x8</td>
<td>1Gb-533MHz-x8</td>
</tr>
<tr>
<td>I(_{DD1})</td>
<td>( \mu ) mA</td>
<td>( \sigma ) mA</td>
<td>%</td>
</tr>
<tr>
<td>I(_{DD})</td>
<td>95.4</td>
<td>2.4</td>
<td>112.2</td>
</tr>
<tr>
<td>I(_{DD1})</td>
<td>104.2</td>
<td>2.3</td>
<td>118.5</td>
</tr>
<tr>
<td>I(_{DD2})</td>
<td>37.7</td>
<td>4.8</td>
<td>42.7</td>
</tr>
<tr>
<td>I(_{DDAN})</td>
<td>41.5</td>
<td>5.7</td>
<td>56.7</td>
</tr>
<tr>
<td>I(_{DDAR})</td>
<td>118.9</td>
<td>2.9</td>
<td>153.9</td>
</tr>
<tr>
<td>I(_{DDAW})</td>
<td>125.9</td>
<td>2.7</td>
<td>159.4</td>
</tr>
<tr>
<td>I(_{DD})</td>
<td>146.2</td>
<td>2.1</td>
<td>161.4</td>
</tr>
<tr>
<td>I(_{DDPA})</td>
<td>8.4</td>
<td>13.7</td>
<td>8.4</td>
</tr>
<tr>
<td>I(_{DD6})</td>
<td>8</td>
<td>14</td>
<td>8</td>
</tr>
</tbody>
</table>

![Figure 4: FS Vs. Current Consumption](image-url)
Table 5: Datasheet Values Vs. Nominal-Case

<table>
<thead>
<tr>
<th>Current</th>
<th>DS mA</th>
<th>μ (mA)</th>
<th>μ vs DS</th>
<th>2σ</th>
<th>2σ vs DS</th>
</tr>
</thead>
<tbody>
<tr>
<td>I/DD0</td>
<td>110</td>
<td>98</td>
<td>-12.2</td>
<td>102.8</td>
<td>-6.96</td>
</tr>
<tr>
<td>I/DD1</td>
<td>125</td>
<td>112.2</td>
<td>-11.4</td>
<td>117.3</td>
<td>-6.53</td>
</tr>
<tr>
<td>I/DD2N</td>
<td>42</td>
<td>33.5</td>
<td>-25.4</td>
<td>36.9</td>
<td>-13.8</td>
</tr>
<tr>
<td>I/DD3N</td>
<td>45</td>
<td>34.6</td>
<td>-30.1</td>
<td>38.7</td>
<td>-16.1</td>
</tr>
<tr>
<td>I/DD4R</td>
<td>270</td>
<td>232.2</td>
<td>-16.2</td>
<td>247.3</td>
<td>-9.15</td>
</tr>
<tr>
<td>I/DD4W</td>
<td>280</td>
<td>246.7</td>
<td>-13.5</td>
<td>260</td>
<td>-7.67</td>
</tr>
<tr>
<td>I/DD5</td>
<td>215</td>
<td>193.6</td>
<td>-11.1</td>
<td>202.1</td>
<td>-6.34</td>
</tr>
<tr>
<td>I/DD2P0</td>
<td>12</td>
<td>6.62</td>
<td>-81.1</td>
<td>8.77</td>
<td>-39.1</td>
</tr>
<tr>
<td>I/DD6</td>
<td>12</td>
<td>6.45</td>
<td>-85.9</td>
<td>8.67</td>
<td>-39.4</td>
</tr>
</tbody>
</table>

5.3 Variation Impact on Application Energy

In these experiments, we employed four randomly selected MediaBench applications [33] including: (1) Ray Tracing, (2) EPIC Encoder, (3) JPEG Encoder, and (4) GSM Decoder. These applications were independently executed on the SimpleScalar simulator [36] with a 16KB L1 D-cache, 16KB L1 I-cache, 128KB L2 cache and 64-byte cache line configuration. We filtered out the L2 cache misses meant for the DRAM and forwarded them through a DRAM controller [37], which generated the memory commands. We also employed the power-down mode conservatively [38] during the idle periods. We compare the energy estimates, when employing the nominal (μ), datasheet (DS) and realistic μ+2σ IDD measures from Table 5, since this covers more than 85% of the memories in one generation. We used the I_DD measures with the DRAMPower tool [20, 23], to estimate DRAM energy consumption, depicted in Figure 5.

![Figure 5: Application Energy using μ and 2σ vs. DS](http://www.es.ele.tue.nl/drampower)

As can be noticed, the energy consumption when using +2σ current estimates is up to 28% lower for the Ray tracing application, compared to using the datasheet estimates. This difference increases to 58%, if the nominal (μ) measures are employed. Similarly, considerable differences are observed for other applications as well, highlighting the significance of variation-aware power and energy estimation.

6. CONCLUSION

In this paper, we demonstrated the effects of process variations on DRAM performance and power consumption. Towards this, we defined a detailed circuit-level DRAM cross-section in NGSPICE and performed Monte-Carlo analysis to derive the impact on DRAM performance and current measures. We also presented a methodology that assesses the performance characteristics of a given DRAM at system boot-time and conservatively identifies new performance parameters in terms of functional speeds, core-timings and I_round and realistic current measures, for use at run-time. We further extended the Monte-Carlo analysis to review the impact of system parameters on current consumption and applied the same on a Micron DDR3 memory, showing significant pessimism in the datasheet measures. In a nutshell, the contributions of this work can be employed to improve DRAM performance and obtain variation-aware realistic and accurate power consumption estimates.

Acknowledgments

This work was partially supported by HP/AC collaboration grant and projects EU FPT 288008 T-CREST and 288248 FlexTiles, Catrène CA104 COBRA, DFG SPP 1500, PT FCT, NL STW 10346 NEST and Artemis 100202 RECOMP.

7. REFERENCES

Appendix

A1: DRAM Currents and Power Consumption

In this section, we describe the different DRAM currents, when and how they are measured, and the state of the banks and changes to the DRAM settings, when they are measured. These currents are also described in detail in [9].

1. **I_{DD0} (One Bank Active-Precharge Current):** Measured across ACT and PRE commands to one bank. Other banks are retained in the precharged state.

2. **I_{DD1} (One Bank Active-Read-Precharge Current):** Measured across ACT, RD, and PRE commands to one bank, while other banks are retained in the precharged state. This measurement is performed twice, targeting two different memory locations and toggling of all data bits.

3. **I_{DD2} (Precharge Standby Current):** Measured when all banks are closed (precharged state).

4. **I_{DD3} (Precharge Power-Down Current - Slow Exit):** Measured during power-down mode, with CKE (Clock Enable) Low and the DLL locked (slow-exit), while the external clock is On and all banks are closed (precharged).

5. **I_{DD4} (Active Standby Current):** Measured when all banks are open (active state).

6. **I_{DD5} (Burst Read Current):** Measured during Read (RD) operation, assuming seamless read data burst with all data bits toggling between bursts and all banks open, with the RD commands cycling through all the banks.

7. **I_{DD6} (Burst Write Current):** Measured during Write (WR) operation, assuming seamless write data burst with all data bits toggling between bursts and all banks open, with the WR commands cycling through all the banks and the ODT (On Die Termination) stable at HIGH.

8. **I_{DD7} (Refresh Current):** Measured during Refresh (REF) operation, with REF commands issued every nRFC.

9. **I_{DD8} (Self Refresh Current):** Measured during self-refresh mode, with CKE Low and the DLL off, while the external clock is Off and all banks are closed (precharged).

A2: Monte Carlo on DRAM Cross-Section

In this section, we present the impact of process-variation on the timing behavior of the DRAM cross-section presented in Section 3.2 by performing Monte-Carlo analysis on the same, considering ±1σ variations in the device and intersect parameters. In Figure 6, we present the effects on the local wordline activation (lw), and the sensing of the true (lbit) and complementary (lbc) bitlines by the PSA.

As can be observed from the figure, there is a significant impact on the timings of the operations associated with the wordline and bitlines. For instance, the local word line reaches its required potential (upon activation) of 2.8V at around 6ns instead of 5ns, which was the case for the baseline configuration without any variation (shown in Figure 3). Similarly, the bitlines reach their potential (upon sensing by the PSA) at around 17ns, compared to around 15ns in the baseline configuration (Figure 3). The variations in the bitlines and wordline impact the activation latency given by the core-timing parameter n_{RCD}, thereby impacting both the DRAM frequency (delay) and power consumption.

A3: Core-Timings Check

In this section, we present the Core-Timings Check function in Algorithm 2, which is an adaptation of the I_{DD1} Measurement Loop test proposed by JEDEC for DRAMs in [9]. We begin by first providing background information on the three core-timings of a DRAM memory.

1. The n_{CL} parameter corresponds to the minimum CAS latency, which is the delay between the Read command and the availability of the first bit of output data.

2. The n_{RCD} parameter corresponds to ACT to RD/WR latency, which defines when an RD/WR can be issued after the ACT has been issued to assure completion of activation.

3. The n_{RP} parameter defined the precharge (PRE) latency, which defines the time required by the precharge operation to completely precharge the local bitlines.

Another important timing constraint to review is the n_{RAS} timing constraint, which gives the minimum delay between ACT and PRE commands to the same bank, thus encompassing both the n_{RCD} and n_{CL} core-timings [9].

The original I_{DD1} test on which the CT_check function is based, is employed by memory vendors to measure the worst-case estimates for I_{DD1} current. Interestingly, this test performs Activation, Read and Precharge operations that employ the three core-timings parameters viz., n_{CL}, n_{RCD} and n_{RP}, which form the core of DRAM performance assessment methodology proposed in this work. Hence, we selected this I_{DD1} test as a part of our methodology, by adapting it to check for functional accuracy of the memory, when the three core-timings parameters are modified by Algorithm 1. We do not use this test for current measurements, as this requires expensive current measurement hardware, which is generally available only with DRAM vendors.

**Algorithm 2: Core-Timings Check**

**Require:** CT_check (k, l, CT[])

1. {Initializing, Bank, Frequency, Data and Address Offsets}
2. Bank = k; Set_Freq = l;
3. Set_CL = CT[0]; Set_RCD = CT[1]; Set_RP = CT[2];
4. Data_0 = 0xAAAAAAA; Addr_0_Offset = 0x0000
5. Data_1 = 0x55555555; Addr_1_Offset = 0x0000
6. {Comment: I_{DD1} Test Phase}
7. for i = 0 to 1 do
8. {Comment: Representing two sets of data and addresses}
9. Issue: ACT, Addr[i]
10. wait(RCD);
11. Issue: RD, Addr[i]
12. wait(RAS-RCD);
13. Recv: Recv_Data[i]
14. if Recv_Data[i] == Data[i] then
15. check = TRUE
16. Issue: PRE, Addr[i]
17. wait(RP);
18. else
19. check = FALSE
20. Issue: PRE, Addr[i]
21. wait(RP);
22. Break;
23. end if
24. end for
25. return check

![Figure 6: Variation Impact on Bitline and Wordline](image-url)
A5: Impact on Power and Delay

In this section, we present the impact of ±1σ variations in device and interconnect parameters on basic memory operations including activation, read, precharge and power-down.

The impact of variation on 

$\frac{I_{DD1}}{I_{DD1}}$ 

active power and the corresponding operation latency ($t_{RCD}+t_{DATA}$) is depicted in Figure 7. This represents activation, read and precharge operations in a particular memory row, with $t_{RCD}$ being the activation period and $t_{DATA}$ being the latency to read the data out. Similarly, the impact on leakage power is plotted against the $t_{RCD}$ delay in Figure 8.

![Figure 7: Impact on ACT-RD-PRE Operations](image)

![Figure 8: Impact on Leakage current](image)

In Figure 9, we present a Q-Q (quantile) plot comparing the distributions observed in active and leakage currents (power) and the delay measures ($t_{RCD}$, $t_{DATA}$) corresponding to ±1σ variations. The linearity in the four measures shows a Gaussian distribution in the variation, as expected.

![Figure 9: Impact on Currents and Timing](image)

These results show the impact of device and interconnect variations on the delay and power consumption of DRAM memories, highlighting the significance of this work.