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

Karthik Chandrasekar[1], Christian Weis[2], Benny Akesson[3], Norbert Wehn[2], Kees Goossens[4]
[1]Computer Engineering, TU Delft, The Netherlands
[2]Microelectronic Systems Design, TU Kaiserslautern, Germany
[3]CISTER-ISEP Research Centre, Polytechnic Institute of Porto, Portugal
[4]Electronic 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.

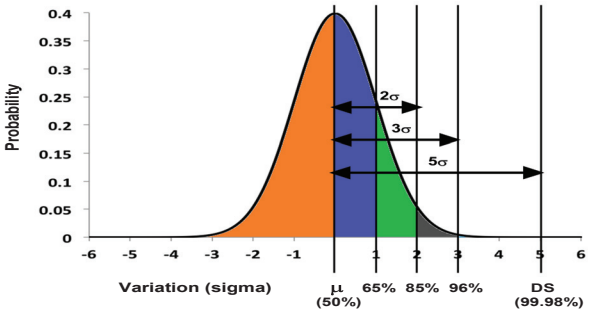
1. INTRODUCTION

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 expected 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.

DAC '13, May 29 - June 07 2013, Austin, TX, USA.

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.



IDD	Typical μ (mA)	+1σ (mA)	+1σ %	+2σ (mA)	+2σ %	+5σ (mA)	+5σ %
IDD1	79.1	80.2	1.4	81.3	2.8	84.7	7.1
IDD2P0	9.2	10.1	9.2	10.9	18.4	13.5	46
IDD6	7.1	7.9	12	8.8	24	11.4	60

Figure 1: Distribution of Current Consumption

This data shows very large difference between the datasheet current measures (DS) and the typical (μ) current values (by a factor of 5σ) of up to 46% and 60% for the low-power modes (power-down: I_{DD2P0} and self-refresh: I_{DD6}) and up to 7% for the activate-read-precharge (I_{DD1}) current [9]. With DRAM memories now being manufactured at technologies below 50nm, these current variations are only expected to worsen, and so is the accuracy of the power models employing the datasheet measures. *Unfortunately, such current distributions are not provided for all DRAMs, and only worst-case measures are given in vendor datasheets* [10].

Intel in [11,12] and others in [13,14] observed similar power variation in DRAM memories and suggested different techniques to work around this problem [14–17]. However, there are no known realistic models or studies that estimate the actual impact of variation on power consumption of a DRAM memory, impairing the usage of these proposed solutions. Existing power models [18–22] choose to ignore the impact of variations on power consumption due to the lack variation data, which reflects poorly on their accuracy.

In this paper, we intend to provide an insight into the possible effects of process variations on DRAM power consumption to help improve the accuracy of DRAM power models and to enable the optimal use of DRAMs at run-time. The three important contributions of this work are: (1) We demonstrate the impact of process variations by performing Monte-Carlo simulations on a detailed DRAM cross-section modeled in NGSPICE [24]. (2) We propose a methodology to empirically determine this impact for any given DRAM memory, by assessing its actual performance characteristics during the DRAM calibration phase [9] at system boot-time. (3) We extend the Monte-Carlo analysis to examine the impact of DRAM architecture parameters, such as capacity, width and frequency, on variations and current estimates.

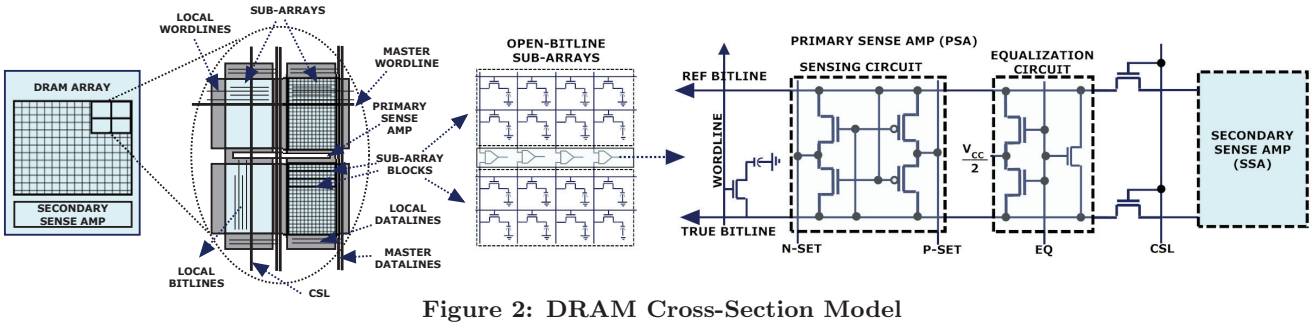


Figure 2: DRAM Cross-Section Model

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 currents to enable use of such solutions.

When it comes to DRAM power estimation, many power models have been proposed. Among the circuit-level models, CACTI 5 [22] was proposed for embedded DRAMs, Rambus presented a circuit-level open-source DRAM power model in [18] and Weis et al. employed a SPICE based model in [19] for 3D-stacked DRAMs. At the system-level, Micron presented a datasheet-based power model in [21] and Chandrasekar et al., proposed a transaction-based power model in [20] that also uses datasheet measures. Unfortunately, none of these power models consider the impact of process-variations on power consumption in DRAMs, due to lack of variation analysis and data. In this work, we provide possible distributions of the current measures for different DRAM operations, which can be employed with the datasheet-based power models, such as [20, 21, 23], to obtain more realistic DRAM power and energy estimates.

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 (lwl) at its gate, which connects the capacitor to the local bitline (lbl) 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 capacitors as long as the transistors remain on. The primary sense amplifiers hold the data till all column accesses to the same row are completed or till a precharge is issued. In our architecture, we used the open bitline array structure and hence differential sense amplifiers (in PSA), which use a reference bitline from a neighboring inactive array segment to detect the minute difference in active bitline voltage. When the Read (RD) command is issued, the data/charge is read out using column select lines (CSLs) from the row buffer (PSA). 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. 16 master wordlines and 16 column select lines (CSLs) connect the 256 memory sub-arrays to form 64Mb memory array macros. The row and column decoders and the master wordline drivers are placed per memory array. The N-Set and P-Set control signal drivers used for activating the primary sense amplifiers are shared between a set of sub-arrays in the memory array. The voltage regulators and charge pumps are shared between the different banks.

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 $6F^2$ 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.

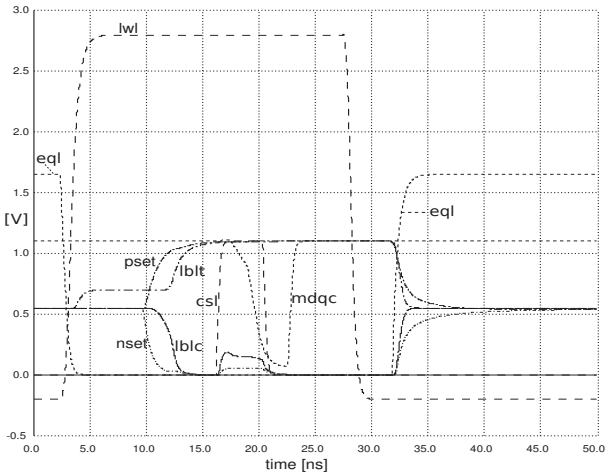


Figure 3: Behavior of DRAM Cross-Section

As depicted in the figure, first the equalization circuit (eql) forces both the true (active) bitline (lblt) and the complementary (reference) bitline (lblc) to the same reference voltage (0.55V). This is followed by the local wordline (lwl) 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 (lblc) 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, which drives the active bitline (lblt) to logic level 1 (the charge stored in the cell corresponds to 1 here) and the reference bitline (lblc) to 0 at around 15ns. Both the lblt and pset signals are driven to the core voltage of 1.1V, while lblc and nset signals are driven to 0V. This is followed by the read operation depicted by the rising column select line (csl) voltage at 18ns. Following this, the charge detected at all the PSAs in a memory sub-array are transferred via their respective local datalines to a master dataline, which is reflected by the current drawn from the NMOS components of the PSA (lblc) 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 (V_{th}) (primarily the variations in line edge roughness (LER) [31]), besides the interconnect parameters including wire width and wire thickness, within pre-defined 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 V_{th} 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 V_{th} 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 V_{th} 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 1: Transistor Process Parameter Variations

Tech *nm*	*Mobility* σ *(%)*	V_{th} *(LER)* σ *(V)*	*Length* σ *(m)*	T_{ox} σ *(%)*
45	8.2	3.0e-9/$\sqrt[2]{(w \times l)}$	45e-9×0.03	1.67

Amongst the different characteristic DRAM currents, we identify the dynamic (active and background) currents as: I_{DD0}, I_{DD1}, I_{DD2N}, I_{DD3N}, I_{DD4R}, I_{DD4W}, and I_{DD5}, and the static (leakage) currents as: I_{DD2P0} and I_{DD6} (when the clock is disabled). 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_{DD} currents. We also provide σ% value to get relative variation for the different current measures.

Table 2: Variation Impact on Current Measures

Current	μ *mA*	σ%	$+1\sigma$ *mA*	$+2\sigma$ *mA*	$+5\sigma$ *mA*
I_{DD0}	98.4	2.37	100.7	103.1	110.6
I_{DD1}	104.3	2.32	106.7	109.1	116.9
I_{DD2N}	37.7	4.77	39.5	41.4	47.6
I_{DD3N}	41.5	5.71	43.8	46.3	54.7
I_{DD4R}	118.1	2.96	121.6	125.2	136.6
I_{DD4W}	123.4	2.75	126.7	130.2	141.2
I_{DD5}	146.1	2.15	149.6	153.1	164.2
I_{DD2P0}	8.41	13.69	9.56	10.9	15.9
I_{DD6}	8.04	14.02	9.17	10.5	15.5

As can be noticed from the table, the $+5\sigma$ 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 $\pm1\sigma$ variations in device and interconnect parameters, as observed from the 1000 Monte-Carlo runs.

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 optimal functional measures for the performance parameters and conservative estimates for the current measures of a particular DRAM memory during the calibration phase at system boot-time, in Section 4.2.

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

Speed Bin	*Freq (MHz)*	*Fast Core (cc)* n_{CL}-n_{RCD}-n_{RP}	*Slow Core (cc)* n_{CL}-n_{RCD}-n_{RP}
800	400	5-5-5(12.5ns)	6-6-6(15ns)
1066	533	7-7-7(13.125ns)	8-8-8(15ns)
1333	666	9-9-9(13.5ns)	10-10-10(15ns)
1600	800	10-10-10(12.5ns)	11-11-11(13.75ns)

In Table 3 in the DDR3-800 speed-bin, the memories are guaranteed to work at the lower bound (F_{LB}) of 400MHz. All memories that fall short of the lower bound of the next speed-bin (and upper bound F_{UB} of the current speed-bin: $\leq$532MHz), are down-binned into the 400MHz speed-bin, ignoring the fact that they can operate at higher frequencies. Besides this frequency-sorted speed-binning, 3 core-timings (in cc) are used to define a DRAM memory within a speed-bin [9]: (1) n_{CL} - RD to Data Latency, (2) n_{RCD} - ACT to RD/WR Latency and (3) n_{RP} - PRE Latency (see Section A3 for details). However, only two sets of core-timings (fast and slow) are used to sub-categorize the memories within a speed-bin. In the case of DDR3-800, the fast memories have core-timings of 5-5-5 cc and slow memories have core-timings of 6-6-6 cc. Hence, memories that may achieve core-timings of 5-5-6 cc are conservatively categorized among the slow 6-6-6 memories, further ignoring their individual core-timings.

When it comes to providing current measures for these memories, only one set of worst-case measures per speed-bin are provided in datasheets [10], thus ignoring the actual performance characteristics of the DRAM memories completely. Determining the actual functional F_{MAX} and core-timings is important to derive the actual performance and current measures of a particular DRAM memory and we present a methodology to obtain the same in Section 4.2.

4.2 Conservative Memory Characterization

As discussed in Section 4.1, when reporting the worst-case current measures in datasheets, DRAM vendors consider only the memories operating near the upper frequency bound with the fastest core timings of their speed-bins. However, since slower memories draw less current than the faster ones, their individual F_{MAX} and core-timings should be determined and used to identify the actual current measures.

In this section, we 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 (Max_FS 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 (Min_FS 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})
{Comment: $\sum$CT = [n_{CL} + n_{RCD} + n_{RP}]}
Define: CT_Min[] = {5,5,5} and CT_Max[] = {8,8,8}
Define: F_σ = (F_{UB} - F_{LB})/12 {Here: F_σ=11MHz}
Define: Banks = 8
CT[i] = CT_Max[i] {Initialized}
for $i = 0 \rightarrow 2$ **do**
  {Comment: Representing CL, RCD and RP}
  for $j = 0 \rightarrow 2$ **do**
    {Comment: Representing CT range 8cc to 5cc}
    CT[i] = CT[i] - j - 1
    for $k = 0 \rightarrow Banks$ **do**
      {Comment: Iterating over all 8 banks}
      for f =$F_{LB} \rightarrow F_{UB}$ **do**
        {Comment: Checking all frequency levels}
        if CT_check (k, f, CT*) == True **then**
          FS_Bank[i][j][k] = $\sum$CT $\times$ $1/f$
          {Comment: Store corresponding f and CT*}
          $f = f$ + F_σ
        else
          Break;
        end if
      end for{f}
      {Comment: Stores least FS for bank k for set CT*}
    end for{k}
  end for{j}
end for{i}
Min_FS = min (FS_Bank[*][*][*])
{Comment: Return corresponding f and CT*}
Max_FS = max (FS_Bank[*][*][*])
{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

memories [8-8-8] (15ns at 532MHz) (in Step 5) and reduce one core-timing parameter (say n_{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 F_σ (here 11MHz between 400MHz and 532MHz, given by Step 13). With these new core-timing settings, we propose to execute a 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_σ (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 n_{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 n_{CL} value is reached. The same procedure is then employed with the other core-timing parameters (n_{RCD} and n_{RP}), assuming the fastest n_{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=15}) at 532 MHz 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 -6σ position at 45ns (slowest). An example of the new performance parameters is highlighted at Max_FS - 1σ position (fastest overall DRAM FS) at 41ns and an example of realistic current measures is highlighted at Min_FS + 1σ position (fastest individual bank FS) at 34ns.

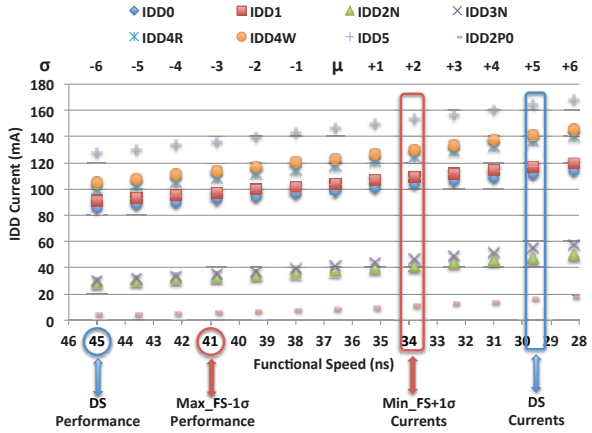


Figure 4: FS Vs. Current Consumption

From this analysis, we derive new conservative performance parameters and realistic current measures for a DRAM, for its optimal run-time usage and power management.

5. RESULTS AND ANALYSIS

In this section, we apply our variation study on different DRAM system configurations. Towards this, we first repeat the Monte-Carlo Analysis with modified system parameters, such as capacity, frequency and data-bus width in Section 5.1. We then apply the results on a 2Gb DDR3-1600 (800MHz) x16 memory from Micron in Section 5.2 and present the μ and σ% estimates for current measures for this memory. Finally, in Section 5.3, we employ these results on a set of MediaBench applications [33], and show up to 28% difference in energy estimates by using more realistic $\mu+2\sigma$ current estimates instead of the datasheet (DS) measures.

5.1 System Parameters Impact on Variation

DRAM vendors sort the memories by three system parameters: frequency, capacity and width. In this experiment, we repeat the Monte-Carlo simulations to analyze the impact on μ and σ% for the different currents when these system parameters change. Our baseline configuration targeted 1Gb DDR3-533MHz x8 memories. In this experiment, we alter the system parameters individually to simulate different configurations. Accordingly, we change: (1) the frequency to 800MHz and simulate a 1Gb-800MHz-x8 memory, (2) the capacity to 2Gb and simulate a 2Gb-533MHz-x8 memory, and (3) the data-width to x16 to simulate a 1Gb-533MHz-x16 memory and observe the impact on μ and σ% in Table 4.

Table 4: System Parameters Vs. Current Measures

	Baseline		Freq		Capacity		Width	
Config	1Gb-533-x8		1Gb-800-x8		2Gb-533-x8		1Gb-533-x16	
I_{DD} Type	μ mA	σ %	μ mA	σ %	μ mA	σ %	μ mA	σ %
I_{DD0}	98.4	2.4	112	2.6	99.3	2.5	98.4	2.37
I_{DD1}	104	2.3	118	2.2	105	2.5	113	2.22
I_{DD2N}	37.7	4.8	42.7	4.5	46.5	6.1	37.7	4.77
I_{DD3N}	41.5	5.7	56.7	4.5	49.9	5.3	41.5	5.71
I_{DD4R}	118	2.9	153	3.5	127	3.3	208	3.14
I_{DD4W}	123	2.7	159	4.1	132	3.7	213	2.6
I_{DD5}	146	2.1	161	2.4	184	2.2	146	2.15
I_{DD2P0}	8.4	13.7	8.4	13.7	16.6	16.1	8.4	13.7
I_{DD6}	8	14	8	14	13.7	19.9	8	14

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_{DD4R} and I_{DD4W} 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_{DD3N}) than the datasheet (DS) measures, while those for the leakage currents are up to 86% lower (for I_{DD6}). These large differences in the current measures highlight the pessimism in the datasheets.

Table 5: Datasheet Values Vs. Nominal-Case

Current	DS mA	μ (mA)	μ vs DS %	2σ mA	2σ vs DS %
IDD0	110	98	-12.2	102.8	-6.96
IDD1	125	112.2	-11.4	117.3	-6.53
IDD2N	42	33.5	-25.4	36.9	-13.8
IDD3N	45	34.6	-30.1	38.7	-16.1
IDD4R	270	232.2	-16.2	247.3	-9.15
IDD4W	280	246.7	-13.5	260	-7.67
IDD5	215	193.6	-11.1	202.1	-6.34
IDD2P0	12	6.62	-81.1	8.77	-36.7
IDD6	12	6.45	-85.9	8.67	-38.4

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 I_{DD} 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.

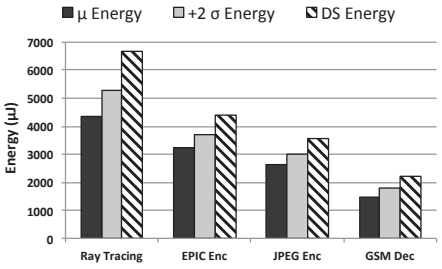


Figure 5: Application Energy using μ and 2σ vs. DS

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 F_{max}) 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 HiPEAC collaboration grant and projects EU FP7 288008 T-CREST and 288248 Flextiles, Catrene CA104 COBRA, DFG SPP 1500, PT FCT, NL STW 10346 NEST and Artemis 100202 RECOMP.

7. REFERENCES

[1] C.H.Berkel, *Multi-core for mobile phones*, In Proc. DATE 2009.
[2] L.Minas et al., *Energy Efficiency for Information Technology: How to Reduce Power Consumption in Servers and Data Centers*, Intel Press, 2009.
[3] Y.Cao et al., *Mapping Statistical Process Variations Toward Circuit Performance Variability: An Analytical Modeling Approach*, In Proc. DAC 2005.
[4] S.Bhardwaj et al., *Modeling of intra-die process variations for accurate analysis and optimization of nano-scale circuits*, In Proc. DAC 2006.
[5] S.Borkar et al., *Parameter variations and impact on circuits and microarchitecture*, In Proc. DAC 2003.
[6] K.Bowman et al., *Impact of Die-to-Die and Within-Die Parameter Variations on the Throughput Distribution of Multi-Core Processors*, In Proc. ISLPED 2007.
[7] X.Liang et al., *Mitigating the Impact of Process Variations on Processor Register Files and Execution Units*, In MICRO 2006.
[8] J.Tschanz et al., *Adaptive Frequency and Biasing Techniques for Tolerance to Dynamic Temperature-Voltage Variations and Aging*, In Proc. ISSCC 2007.
[9] JEDEC SST Assn., *DDR3 Specification*, JESD79-3E, 2010.
[10] Micron Tech. Inc., *2Gb: X4, X8, X16 DDR3 Datasheet*, 2010.
[11] Intel, *Memory 3-sigma Power Analysis Methodology.*
[12] S.Y.Ji et al., *An Empirical Study of Performance and Power Scaling of Low Voltage DDR3*, In Proc. EPEPS 2010.
[13] M.Gottscho et al., *Power Variability in Contemporary DRAMs*, IEEE Embd. Sys. Letters, Vol. 4, No. 2, 2012.
[14] S.Desai et al., *Process Variation Aware DRAM Design Using Block Based Adaptive Body Biasing Algorithm*, In ISQED 2012.
[15] H.David et al,. *RAPL: memory power estimation and capping*, In Proc. ISLPED 2010.
[16] L.Bathen et al., *Vamv: Variability-aware memory virtualization*, In Proc. DATE 2012.
[17] L.Bathen et al., *ViPZonE: OS-Level Memory Variability-Driven Physical Address Zoning for Energy Savings*, In Proc. CODES+ISSS 2012.
[18] T.Vogelsang, Rambus Inc., *Understanding the Energy Consumption of Dynamic Random Access Memories*, In Proc. MICRO 2010.
[19] C.Weis et al., *Design Space Exploration of 3D-stacked DRAMs*, In Proc. DATE 2011.
[20] K.Chandrasekar et al., *Improved Power Modeling of DDR SDRAMs*, In Proc. DSD 2011.
[21] Micron Technology Inc., *TN-41-01: Calculating Memory System Power for DDR3*, Tech Report, 2007.
[22] S.Thoziyoor et al., *CACTI 5.1*, HP Labs, 2008.
[23] K.Chandrasekar et al., *DRAMPower: Open-source DRAM power & energy estimation tool*, *www.es.ele.tue.nl/drampower*
[24] NGSPICE, *http://ngspice.sourceforge.net/*
[25] T. Schloesser et al., *A 6F^2 Buried Wordline DRAM Cell for 40nm and Beyond*, In Proc. IEDM 2008.
[26] B.Jacob et al., *Memory Systems: Cache, DRAM, Disk*, Morgan Kaufmann Publishers, 2008.
[27] BSIM4.6, *http://www-device.eecs.berkeley.edu/bsim/*
[28] Predictive Technology Model, *http://ptm.asu.edu/*
[29] ITRS, *Design*, Report and Tables, 2011.
[30] ITRS, *Modeling & Simulation*, Report and Tables, 2011.
[31] Y. Ye et al., *Statistical Modeling and Simulation of Threshold Variation Under Random Dopant Fluctuations and Line-Edge Roughness*, IEEE Trans. VLSI Sys. vol.19, no.6, 2011.
[32] C.Lin et al., *Statistical Compact Modeling of Variations in Nano MOSFETs*, In Proc. VLSI-TSA 2008.
[33] C.Lee et al., *MediaBench: a tool for evaluating and synthesizing multimedia and communications systems*, In Proc. MICRO 1997.
[34] A.Keshavarzi et al., *Leakage and Process Variation Effects in Current Testing on Future CMOS Circuits*, IEEE Design & Test Comp., vol.19, no.5, 2002. ASP-DAC 2002.
[35] S.Herbert et al., *Variation-aware dynamic voltage/frequency scaling*, In Proc. HPCA 2009.
[36] T.Austin et al., *SimpleScalar: An infrastructure for computer system modeling*, IEEE Computer, vol.35, no.2, 2002.
[37] B.Akesson et al., *Architectures and Modeling of Predictable Memory Controllers for Improved System Integration*, In Proc. DATE 2011.
[38] K.Chandrasekar et al., *Run-Time Power-Down Strategies for Real-Time SDRAM Memory Controllers*, In Proc. DAC 2012.

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_{DD2N} (Precharge Standby Current)**: Measured when all banks are closed (precharged state).

(4) **I_{DD2P0} (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_{DD3N} (Active Standby Current)**: Measured when all banks are open (active state).

(6) **I_{DD4R} (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_{DD4W} (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_{DD5} (Refresh Current)**: Measured during Refresh (REF) operation, with REF commands issued every nRFC.

(9) **I_{DD6} (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 using Monte-Carlo analysis on the same, considering $\pm 1\sigma$ variations in the device and interconnect parameters. In Figure 6, we present the effects on the local wordline activation (lwl), and the sensing of the true (lblt) and complementary (lblc) bitlines by the PSA.

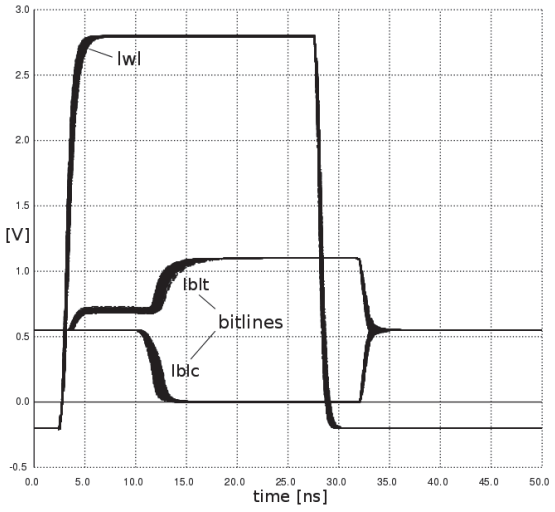


Figure 6: Variation Impact on Bitline and Wordline

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-timing 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-timing 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 = 0xAAAAAAAA; Addr_0_Offset = 0x0000
5: Data_1 = 0x55555555; Addr_1_Offset = 0x000F
6: {Comment: I_{DD1} Test Phase}
7: **for** $i = 0 \rightarrow 1$ **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**{i}
25: **return** check

The inputs to Algorithm 2 include selected core-timings, operating frequency and target bank provided by Step 16 in Algorithm 1. We begin by initializing these settings (Steps 2 and 3) and selecting two unique data values with all bits toggling and two address offsets within the target bank, as required by the JEDEC I_{DD1} test loop (Steps 4 and 5). Before the test phase commences (in Step 7), the two data values are written at the corresponding addresses.

In the I_{DD1} test phase, two sets of ACT-RD-PRE operations are performed, with each targeting a different row in the same memory bank and all the data-bits toggling across the two accesses. Note that since the n_{RAS} parameter covers the period between ACT and PRE, it encompasses both the n_{RCD} and n_{CL} core-timings within itself. As a result, the total latency for one set of ACT-RD-PRE operations performed in this test is given by n_{RAS}+n_{RP}. The test begins by issuing an ACT to the address of the first specified transaction and waiting n_{RCD} clock cycles for it to complete (Steps 9 and 10). This is followed by issuing the READ command to the same address and waiting n_{RAS} - n_{RCD} clock cycles (representing the n_{CL} core-timing parameter and the complete data transfer period) to read out the data from the corresponding address (Steps 11 to 13). Once the data is received, we adapted the I_{DD1} test to merely check this data against the expected value in Step 14, to verify the correct functioning of the memory for the ACT and RD operations. If this test passes, a precharge is issued (Step 16) and after waiting for n_{RP} cycles for the completion of the precharge operation (Step 17), the second transaction starts. If the test fails, the test issues a precharge and waits for its completion (Steps 20 and 21) returns a FALSE to the CT_check call in Algorithm 1. If both set of ACT-RD-PRE tests pass, a TRUE is returned instead.

To enable this performance assessment, the core-timings and frequency are set using the Mode register settings [9] and frequency scaling is performed using existing support in standard DRAM memory controllers.

A4:Combination of System Parameters

Using the analysis presented in Table 4 in Section 5.1, it is now possible to estimate the impact of the three system parameters viz., capacity (C), frequency (F) and width (W), on DRAM current consumption. However, when a combination of system parameters differ between two DRAM memories, the influence all the concerned system parameters must be taken into account. This can be derived directly from the results in Table 4 by adding the corresponding impact (μ and σ%) for one parameter at a time, considering the most influential parameter first (determined by % change in μ). In this section, we present the corresponding impact of the combinations in Table 6, by extrapolating the results presented in Section 5.1, using the same system parameter values. Similar extrapolation was performed to derive the current distributions in Section 6 for the Micron memory. The baseline values for μ and σ% are presented in Table 2.

Table 6: Multi-Parameter Impact on Currents

	F&C		F&W		C&W		F&C&W	
Config	2Gb-800-x8		1Gb-800-x16		2Gb-533-x16		2Gb-800-x16	
I_{DD} Type	μ mA	σ %	μ mA	σ %	μ mA	σ %	μ mA	σ %
I_{DD0}	113	2.5	112	2.4	99.3	2.4	113	2.4
I_{DD1}	119	2.4	127	2.3	115	2.3	128	2.3
I_{DD2N}	48.6	6.3	42.7	5.0	43.7	5.1	48.6	5.1
I_{DD3N}	52.4	5.5	56.7	5.9	47.2	6.0	52.4	6.0
I_{DD4R}	159	3.4	244	3.2	214	3.2	250	3.2
I_{DD4W}	163	3.8	250	2.7	217	2.7	253	2.7
I_{DD5}	194	2.3	161	2.2	179	2.2	194	2.2
I_{DD2P0}	14.1	18.3	8.4	15.5	14.1	15.9	14.1	16.2
I_{DD6}	11.2	22.6	8.0	15.9	11.2	16.8	11.2	17.2

A5: Impact on Power and Delay

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

The impact of variation on 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.

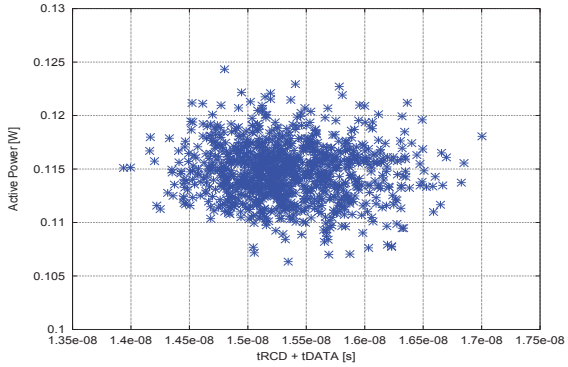


Figure 7: Impact on ACT-RD-PRE Operations

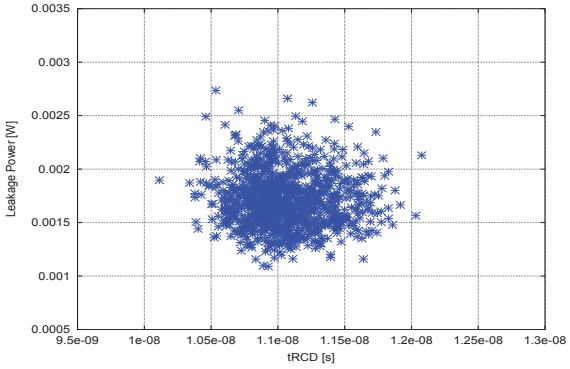


Figure 8: Impact on Leakage current

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 $\pm 1\sigma$ variations. The linearity in the four measures shows a Gaussian distribution in the variation, as expected.

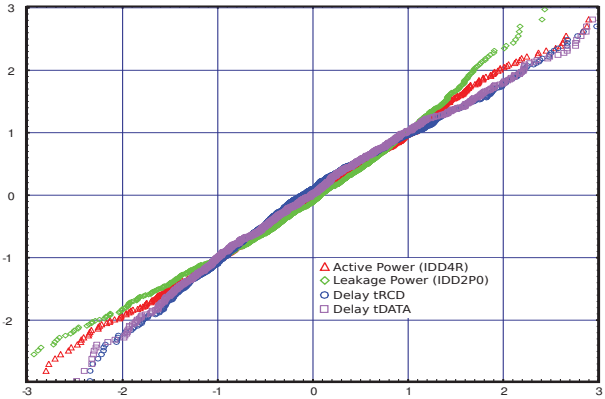


Figure 9: Impact on Currents and Timing

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.