Exploiting Expendable Process-Margins in DRAMs for Run-Time Performance Optimization

Karthik Chandrasekar[1], Sven Goossens[2], Christian Weis[3], Martijn Koedam[2],
Benny Akesson[4], Norbert Wehn[3], Kees Goossens[2]

[1]Computer Engineering, TU Delft, The Netherlands
[2]Electronic Systems Group, TU Eindhoven, The Netherlands
[3]Microelectronic Systems Design, TU Kaiserslautern, Germany
[4]Control Engineering, Czech Technical University, Czech Republic

***Abstract*—Manufacturing-time process (P) variations and run-time voltage (V) and temperature (T) variations can affect a DRAM's performance severely. To counter these effects, DRAM vendors provide substantial design-time PVT timing margins to guarantee correct DRAM functionality under worst-case operating conditions. Unfortunately, with technology scaling these timing margins have become large and very pessimistic for a majority of the manufactured DRAMs. While run-time variations are specific to operating conditions and as a result, their margins difficult to optimize, process variations are manufacturing-time effects and excessive process-margins can be reduced at run-time, on a per-device basis, if properly identified.**

In this paper, we propose a generic post-manufacturing performance characterization methodology for DRAMs that identifies this excess in process-margins for any given DRAM device at run-time, while retaining the requisite margins for voltage (noise) and temperature variations. By doing so, the methodology ascertains the actual impact of process-variations on the particular DRAM device and optimizes its access latencies (timings), thereby improving its overall performance. We evaluate this methodology on 48 DDR3 devices (from 12 DIMMs) and verify the derived timings under worst-case operating conditions, showing up to 33.3% and 25.9% reduction in DRAM read and write latencies, respectively.

I. Introduction

From high-performance to mobile computing systems, DRAM access times contribute significantly to the so-called "Memory Wall" problem [1]. Recently, DRAM vendors and JEDEC have come up with different solutions to address this issue, including high-frequency [2] and multi-channel [3] DRAMs. Although these solutions address the memory bandwidth problem, they do not resolve the latency issue, which remains a critical performance bottleneck. These memory access latencies are high due to two factors: (1) the internal analog delays in DRAMs [4] and (2) the conservative timing margins [5] added by DRAM vendors to improve their production yield. While the internal delays are an inherent design aspect, the timing margins are a design choice, for instance, for accurate DRAM functionality at +85°C. Unfortunately, these timing margins are often too pessimistic for a majority of the manufactured DRAMs and the access latencies do not reflect a DRAM's achievable performance.

When estimating these timing margins, vendors consider three primary variation sources (besides aging [6]) that can affect a DRAM's performance: (1) Process (P), (2) Voltage (V) and (3) Temperature (T), also referred to as PVT variations [7].

Process variations are observed due to manufacturing-time disparities in device parameters, such as channel mobility, length and oxide thickness [8], [9]. Their impact on DRAM timings can vary randomly across all devices produced with the same configuration [5]. Hence, vendors add significant process (timing) margins to cover the worst-case impact on the entire lot.

When it comes to run-time variations in supply voltage and operating temperature, these have a defined and deterministic effect on all manufactured DRAMs, as opposed to the random and distributed effects of process variations.

Voltage variations are represented by noise in the power-supply, which increases the transistor propagation delays in the device. To address this, DRAM vendors define an acceptable operating voltage range (between 1.425V and 1.575V for DDR3 [10], [11]) and add appropriate noise-margins to assure correct functionality in the presence of maximum noise.

Temperature variations are observed due to two factors: (1) self-heating during operation and (2) ambient temperature. High operating temperatures also increase the propagation delays. Hence, DRAM vendors define an operating temperature limit of up to +85°C for DRAMs and add temperature-margins to assure correct DRAM functionality at that temperature.

Besides these variations, aging also impacts DRAM performance [6]. However, all devices susceptible to aging and infant failures are discarded by vendors and hence, all shipped DRAMs are guaranteed to work reliably for a long time. This filtering is done by the 'burn-in' test [12], which pre-ages the devices by stress testing them at +125°C and 1.9V and identifies and removes the devices likely to fail early (in $\leq$ 10 years).

In a nutshell, while requisite voltage (noise) and temperature margins are required to guarantee correct functionality under extreme operating conditions, process-margins are generously over-dimensioned [13] and reflect the worst-case. In short, most DRAMs can perform better than their specifications.

Figure 1 depicts the probability density functions (pdfs) of the impact of process-variations on critical DRAM timings [4] for 1Gb Micron DDR3-800 (400MHz) devices [11], in clock cycles (cc). These pdfs are derived from Monte-Carlo simulations on an NGSPICE model of a DRAM cross-section [9]. The datasheet measures are given in brackets. As can be noticed, the fastest timings can be up to 66% lower than datasheet measures.

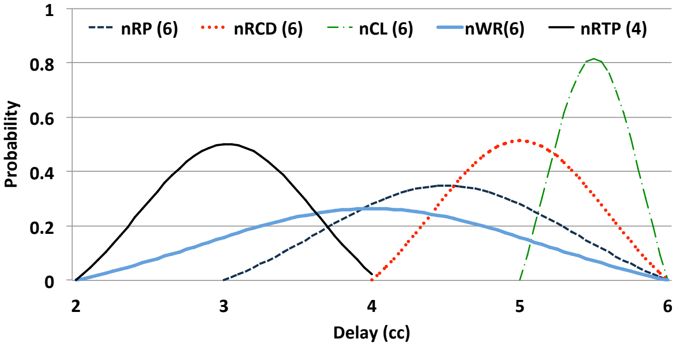

Fig. 1: Distribution of actual vs. worst-case datasheet timings

978-3-9815370-2-4/DATE14/©2014 EDAA

To address this pessimism in DRAM timings, we propose a generic post-manufacturing performance characterization methodology for DRAM devices that identifies their actual achievable performance at run-time under worst-case operating conditions (maximum noise and highest temperature), thereby, improving their overall performance and energy-efficiency.

We demonstrate this methodology on 48 DDR3 devices (from 12 identical DIMMs from one vendor [14]), derive their actual delays and verify their correct functionality under worst-case operating conditions (maximum allowed power-supply noise and temperature). In doing so, we achieve up to 33.3% and 25.9% reduction in DRAM read and write latencies, respectively.

II. Background

DRAMs consist of memory cells arranged in rows and columns, organized as a set of banks. Each bank is equipped with a set of row-buffers that act as intermediates for reading from or writing into the memory cells. To access a DRAM, a memory controller issues a set of commands in a specific order to perform a given operation [4]. For instance, when reading from the DRAM, an activate command is issued to transfer the data from the cells through the bitlines to a row-buffer and then the row-buffer is partly read from based on the memory's interface width and burst length. Similarly, when writing, data is first written to the row-buffer and then a precharge command is sent to store the charge into the DRAM cells. For efficient design, the row-buffers are shared between a set of successive rows in a bank, only one of which can be accessed at a time.

Additionally, when issuing these commands certain timings must be satisfied. The timings associated with reading and writing and their datasheet measures (incl. process-margins), for Micron DDR3-800 devices [11] are specified in Table I:

TABLE I: Micron DDR3-800 (400MHz) Timing Constraints

Constraint	*Description (Minimum Time between)*	*Time (cc)*
nRCD	Activating and Reading/Writing from/to a row	6
nRP	Precharging and Activating a row	6
nRTP	Reading from and Precharging a row	4
nWR	Write Recovery and Precharging a row	6
nCL	Reading/Writing to first data	6

The goal of this work is to identify the excessive process-margins in these datasheet measures, specific to a given DRAM device. Some of these timings along with the operating frequency are also used to classify the DRAM devices into different speed-bins. Table II presents four DDR3 speed-bins [11].

TABLE II: DDR3 Speed-Bins

Speed Bin	*Freq (MHz)*	n_{CL} - n_{RCD} - n_{RP} *(ns)*
800 [Slow-Core]	400	15.0 - 15.0 - 15.0
800 [Fast-Core]	400	12.5 - 12.5 - 12.5
1066 [Slow-Core]	533	15.0 - 15.0 - 15.0
1066 [Fast-Core]	533	13.125 - 13.125 - 13.125

Note that the memories capable of operating between 400MHz and 532MHz and with any of the timings slightly larger than 12.5ns, are down-binned as '800 [Slow-Core]', ignoring the fact that they can perform better than their classified speed-bin.

Besides these timings, nRFC gives the minimum time required for a DRAM refresh, nREFI gives the average time between refreshes, nFAW gives the time between activating more than four banks in parallel and nRRD gives the time between activating two banks. If any of these timings are violated, the DRAM's reliability can be affected. Also, the nCL parameter cannot be optimized due limitations imposed by JEDEC in DRAM designs. Hence, we do not consider optimizing these timings.

III. Performance Characterization Methodology

In this section, we propose a post-manufacturing performance characterization methodology for DRAMs to identify their actual achievable performance at run-time under worst-case operating conditions (lowest voltage - 1.425V and highest temperature - +85°C). *To verify their functional correctness, we propose to read from a memory location that has already been written into with a known dataset and verify if the read data is the same.* We begin by identifying a set of requirements and solutions for deriving a comprehensive DRAM characterization methodology. We then propose three algorithms that can be used at run-time to determine the actual timing measures for a given DRAM device in nominal conditions (1.5V and room temperature). Next, we derive conservative timing margins to address noise and temperature variations using an NGSPICE DRAM model [9] to assure correct DRAM functionality under worst-case conditions.

A. Requirements and Solutions

This section defines a set of requirements to derive a comprehensive DRAM characterization methodology, with the aim of testing and identifying the robustness of the circuitry associated with activation, precharging, reading and writing operations.

(1) Requirement: Assure completion of DRAM operations - Data previously written into the row may be partially retained in the row buffer, if the precharging and activation operations are only partially completed. When reading from a DRAM row, it is therefore important to assure that these operations are completed and that the data is being read from the cells.

Solution: Internally, DRAMs employ an open-bitline array structure for their row-buffers [4], which implies that the row buffers are shared between adjacent memory rows (as described in Section II). This pair-wise sharing enables use of bitlines of either of the rows as voltage reference by the differential sense amplifiers in the row buffers, when the other row is being accessed. Hence, if both of these rows are written into in succession with the exact negated dataset, it would assure that the row buffer is completely over-written with flipped bits of data before reading from the first row of cells. If the test is successful, this guarantees the completion of all relevant operations.

(2) Requirement: Varied datasets - It is important to detect the impact of the charge in neighboring bitlines on the bitline in the middle [15]–[17], due to the possibility of crosstalk in high speed signals. For instance, if the neighboring bitlines transfer the opposite charge as the bitline in the middle, it is important to observe the effect on that bitline (if any). It is also important to test each bitline for stuck-at faults [15], [16].

Solution: To assure this, the datasets should include data with: (a) neighboring bits flipped and (b) all bits set to '1' or '0'. According to requirement (1), each dataset should have a negated dataset. Thus, we derive these four datasets (ds):
ds [0] - {0xA5A5A5A5} ds [1] - {0x5A5A5A5A}
ds [2] - {0xFFFFFFFF} ds [3] - {0x00000000}

To properly test each DRAM cell and the associated circuitry, we should employ all combinations of these datasets, while writing into and reading from alternating rows with flipped datasets. Hence, we derive the following testsets (ts) in Table III:

TABLE III: Testsets

	ts [i] [0]	ts [i] [1]	ts [i] [2]	ts [i] [3]
ts [0] [j]	W [x] [0]	W [x+1] [1]	R [x] [0]	R [x+1] [1]
ts [1] [j]	W [x] [1]	W [x+1] [0]	R [x] [1]	R [x+1] [0]
ts [2] [j]	W [x] [2]	W [x+1] [3]	R [x] [2]	R [x+1] [3]
ts [3] [j]	W [x] [3]	W [x+1] [2]	R [x] [3]	R [x+1] [2]

Here in a W/R [x][y] test, W/R refers to writing or reading operation, x refers to the memory row in the bank and y refers to the dataset element being written or verified against.

(3) Requirement: Test the entire DRAM - Some circuit components may perform better than the others within or across different DRAM devices [15]–[17]. Hence, all DRAM circuitry relevant to reading from and writing into all DRAM cells must be tested and the weakest components identified.

Solution: In the testsets defined before, each Write and Read test corresponds to accessing 64 bytes of data. These testsets write into and read from 512 cells per row, 4 times each, two rows at a time, with different datasets. Once the testsets finish, we move to the next 512 cells in the same two rows till we reach the end of the rows, before switching to the next two rows in the same bank. These are then repeated for all banks. When finished, all DRAM cells and all associated circuitry would be tested and the best achievable performance identified.

(4) Requirement: Worst-case test - It is important to assure functional correctness of the DRAM under worst-case operating conditions supported by default (maximum power-supply noise (1.425V) and maximum case temperature (+85°C)).

Solution: While pruning out the excessive expendable process-margins, requisite noise and temperature margins must be retained. These noise and temperature margin compensations must be identified, conservatively derived and verified.

B. DRAM Characterization Algorithms

In this section, we define a set of algorithms that determine the fastest timing measures for a given DRAM device at run-time under nominal test conditions (+27°C and 1.5V supply).

Algorithm 1 (Memory Check), writes, reads and verifies each of the datasets based on the combinations explored by the different testsets. These operations are performed over the entire memory range, writing and reading four times per memory location, in the order specified before. This algorithm satisfies the first 3 requirements in Section A.

Algorithm 1 Memory Check (Mem_Check)

```
Require: RD[ ], WR[ ]
for all banks, rows, columns do
    for i = 0 $\rightarrow$ 3 do
        for j = 0 $\rightarrow$ 3 do
            if j $\leq$ 1 then
                Write[j%2] = ts[i][j]
            else
                Read[j%2] = ts[i][j]
                if Read[j%2] $\neq$ Write[j%2] then
                    Return FAIL
                end if
            end if
        end for{j}
    end for{i}
end for{banks, rows, columns}
Return PASS
```

The best timings check (btc) function in Algorithm 2 reduces a target timing parameter (indicated by Test_ID, incremented to identify the next parameter) one cycle at a time and derives new read (RD[ ]) and write (WR[ ]) test patterns (set of DRAM commands) using the reduced timing measures. It then calls the memory check function (Algorithm 1) to verify correct DRAM functionality, with the new test patterns. In Algorithm 2, RD[1] corresponds to the DRAM command issued on clock cycle #1 in the read test pattern. Once regular DRAM commands (ACT, READ, WRITE, PRE) are issued, an explicit NOP command is issued to indicate last clock cycle in the test patterns.

For each DRAM device, the test trigger in Algorithm 3, calls Algorithm 2 for each timing parameter to be optimized. It first targets nRCD and nRP timings, since they concern independent memory operations (activating and precharging). It first optimizes one of them and uses the derived minimum value to optimize the other (ordering is irrelevant).

Next, it targets the read/write to precharge timings viz., nRTP and nWR. They are related to each other since they both include nCL (column access latency). In this case, it first individually minimizes both of them to identify their best measures. Then it employs the minimal independent measure of nRTP and tries optimizing nWR and then employs the minimal independent measure of nWR and tries optimizing nRTP. Finally, it identifies one of the combinations of nRTP and nWR that gives the minimum sum of the two parameters. If the application is read or write dominant, the appropriate timing (nRTP or nWR) may be targeted specially for minimization. Each parameter is reduced from its datasheet value to 1 or till the test returns a FAIL.

Algorithm 2 Best Timings Check (btc)

```
Require: Test_ID, RCD, RP, RTP, WR
Init [ ] = {RCD,RP,RTP,WR}
Min [ ] = {1,1,1,1};
# Define: WL = {Write Latency}; BL = 8 {Burst Length}
for $i = Init[Test\_ID] - 1 \rightarrow Min[Test\_ID]$ do
    Init[Test_ID] = i
    RCD=Init[0], RP=Init[1], RTP=Init[2], WR=Init[3]
    RD[1] = ACT
    RD[RCD] = READ
    RD[RCD+RTP] = PRE
    RD[RCD+RTP+RP-1] = NOP
    WR[1] = ACT
    WR[RCD] = WRITE
    WR[RCD+WL+WR+BL/2] = PRE
    WR[RCD+WL+WR+BL/2+RP-1] = NOP
    if Mem_Check(RD[ ],WR[ ]) == FAIL then
        Return Init[Test_ID] + 1
        Break;
    end if
end for{$i$}
Return Init[Test_ID]
```

Algorithm 3 Test Trigger

```
Require: Trigger(RCD,RP,RTP,WR)
for j = 0 $\rightarrow$ 3 do
    bRCD[j] = btc(0,RCD,RP,RTP,WR)
    bRP[j] = btc(1,bRCD[j],RP,RTP,WR)
    bRTP_ind[j] = btc(2,bRCD[j],bRP[j],RTP,WR)
    bWR_ind[j] = btc(3,bRCD[j],bRP[j],RTP,WR)
    bRTP_wr[j] = btc(2,bRCD[j],bRP[j],RTP,bWR_ind[j])
    bWR_rtp[j] = btc(3,bRCD[j],bRP[j],bRTP_ind[j],WR)
    if (bRTP_wr[j] + bWR_ind[j]) $\leq$ (bWR_rtp[j] + bRTP_ind[j])
    then
        bRTP[j] = bRTP_wr[j], bWR[j] = bWR_ind[j]
    else
        bRTP[j] = bRTP_ind[j], bWR[j] = bWR_rtp[j]
    end if
end for{j}
```

In all, Algorithm 1 (Mem_Check) is called 26 times, with each call lasting 4 seconds. The entire memory characterization is performed once at system boot-time and finishes in less than 2 minutes. For ECC DRAMs, we propose to disable ECC during characterization and re-enable during regular operation. Together these algorithms derive the fastest timings under nominal operating conditions at which the DRAMs continue to work. However, these timings must be compensated for noise and temperature variations (as required), which is addressed next.

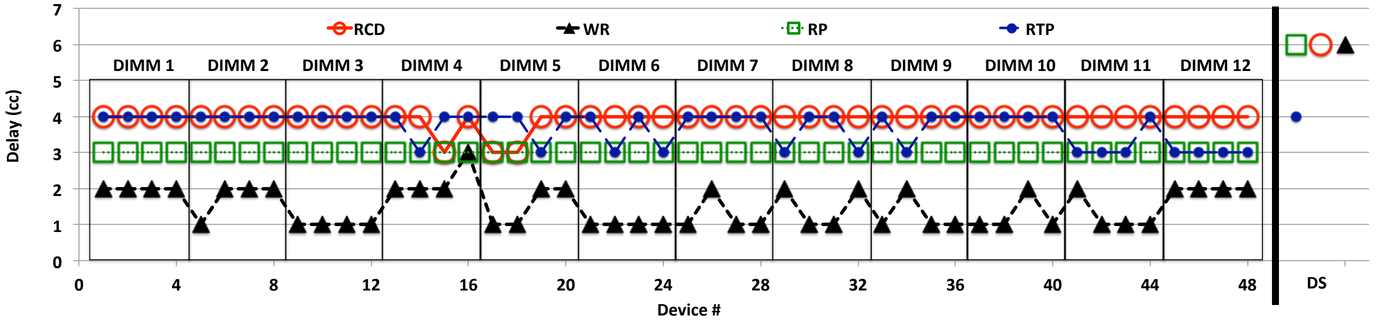

Fig. 2: Actual vs. Datasheet (DS) Timing Measures - 1Gb DDR3-800 devices - Nominal Conditions

C. Conservative Voltage and Temperature Compensations

The goal of this work is to eliminate the excessive process-margins from a DRAM post-manufacturing, while retaining the requisite temperature and noise margins and assuring functional correctness under worst-case conditions. These can be identified if either the worst-case test conditions are employed during characterization or equivalent worst-case circuit-level simulations are performed. Since access to such an experimental setup cannot be expected of the users of this methodology, we propose to derive the impact of noise and temperature on these critical timing measures using the NGSPICE DRAM model from [9].

From our NGSPICE experiments, we derive the following results: Table IV presents the impact of power-supply noise and temperature variations on the critical DRAM timings. The power-supply noise impact is derived at +85°C for voltage swings between 1.425V and 1.575V. As observed, the delays increase with increase in noise (reduced power-supply). The temperature-variation impact is derived at 1.425V. Operating temperatures between +27°C (nominal) and +85°C (maximum) are simulated and as expected, the higher the temperature, the longer the delays. Here, tRCD refers to nRCD in ns.

TABLE IV: Impact of Noise and Temperature on Timings

	Noise @ +85°C			Temperature @ 1.425V		
Timings	*1.575V*	*1.5V*	*1.425V*	*+27°C*	*+70°C*	*+85°C*
tWR (ns)	5.19	5.28	5.38	4.79	5.22	5.38
tRP (ns)	5.60	6.01	6.68	6.48	6.64	6.68
tRCD (ns)	9.64	9.83	10.02	9.00	9.72	10.02
tRTP (ns)	9.64	10.13	10.66	9.12	10.09	10.66

Using these observations, we derive the voltage and temperature margin compensations for 1Gb DDR3 devices in Table V, to be added to the timings obtained at nominal conditions using the algorithms in Section B. We also extend our analysis to study the influence of increasing DRAM capacities on these margin compensations, since that also increases the number of row-buffers in the DRAM. We present the results for higher capacity (2Gb) DDR3 memories in Table V as well. *Since we are minimizing analog delays, the derived margin compensations will be the same for all DRAM frequencies.*

TABLE V: Conservative Margin Compensations

	1Gb			2Gb		
	Nominal	*WC*	*Diff*	*Nominal*	*WC*	*Diff*
Timings	*1.5V* *+27°C*	*1.425V* *+85°C*	*Margin* *(%)*	*1.5V* *+27°C*	*1.425V* *+85°C*	*Margin* *(%)*
tWR (ns)	4.69	5.38	**14.71**	4.71	5.38	**14.21**
tRP (ns)	5.88	6.68	**13.5**	5.88	6.68	**13.6**
tRCD (ns)	8.84	10.02	**13.3**	8.87	10.02	**13.03**
tRTP (ns)	8.84	10.66	**20.55**	9.74	12.79	**31.36**

As can be noticed, the margins for tRTP increased with capacity, since it may take slightly longer to initiate a precharge after a read operation due to the higher number of row-buffers.

When employing these derived margin compensations, we propose rounding-up of the resultant timing measures to integer clock cycles, to be conservative. Note that the timings are the same for all vendors, since they are defined by JEDEC and are not vendor-specific. These derived conservative margins address the worst-case test requirement (4) in Section III.A. Next, we derive these timings (including compensations) for 12 identical DDR3 DIMMs and verify them under worst-case conditions.

IV. Experiments

We perform three experiments to evaluate our performance characterization methodology. In the first experiment, we perform this characterization on a set of twelve 512MB MICRON DDR3-1066 DIMMs [14] (operated at 400MHz) each with four 1Gb x16 DDR3 devices. We derive the fastest and compensated set of timings for each of the 4 devices. In the second experiment, we verify if the most scaled DIMM operates correctly with the compensated timings under worst-case conditions. In the final experiment, we observe the benefits of this characterization on the energy, bandwidth and latency of DRAM operations.

A. Deriving fastest and compensated timings

To perform these tests, we use Xilinx ML605 boards mounted with 512MB DDR3 DIMM module and program the FPGA to: (1) trigger the test algorithms through a MicroBlaze, (2) program the test patterns to a memory controller [19] that uses the Xilinx PHY [20] to communicate to the DRAM, and (3) verify the dataset read back in the memory controller. This test returns the actual fastest timing measures at which the DIMM and the individual devices on the DIMM continue to work under nominal test conditions (1.5V/+27°C). The DIMM is operated at 400MHz instead of 533MHz, due to limitations of the Xilinx PHY [20]. However this does not have any impact, since the optimized delays are frequency-independent analog timings. The results for the 48 DDR3 devices (from 12 DIMMs) are presented in Figure 2. As can be noticed, the datasheet (DS) measures are very pessimistic compared to the actual measures. We add the derived margin compensations (from Table V) to obtain the compensated measures (rounded-up to integer clock cycles) for the critical timings. In the next section, we verify if these compensated measures hold for the fastest (most scaled) DIMM (#6) under worst-case operating conditions.

B. Verifying Compensated Timings

To emulate worst-case operating conditions, we introduce: (1) maximum noise in the power-supply (by reducing supply voltage to 1.42V), (2) maximum temperature variations (by forcing extreme operating temperature of +85°C) and (3) heavy DRAM usage (by continuously reading and writing from the DRAM for over several hours without idling).

1) Experimental Setup: To introduce worst-case operating temperature and maximum noise, we devise our experimental setup as described below: JEDEC specifies that commercial DRAM modules should work reliably up to +85°C [10]. To test this specification, we locally heat the DRAM module to +85°C without overheating the rest of the ML605 platform. The experimental setup is depicted in Figure 3. A small 4W Peltier element (PE) [21] is used to heat up a single DRAM module, while sensors placed on top of the DRAM module provide feedback to keep it at the required temperature. The cold side of the PE is connected to a sufficiently large heat-sink to keep it at room-temperature. The hot side is then connected to the DRAM module using thermal paste. Two temperature sensors are placed between the PE and the DRAM module: a thermocouple for accurate tracking of the temperature and a thermistor used by a control loop to regulate the temperature, based on JEDEC [22] and Micron [23] standards.

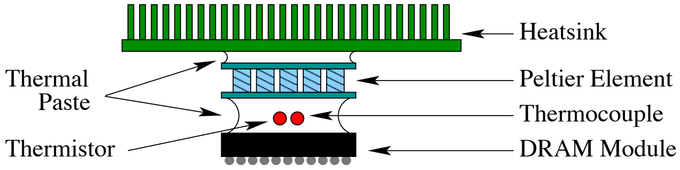

Fig. 3: Experimental Setup

Both the PE and the thermistor are connected to an Arduino board [24] that implements a simple control loop regulating the temperature. It reads out the current temperature using the thermistor and based on the temperature turns the PE on and off. In doing so, it limits the output of the PE using Pulse Width Modulation (PWM). Both the Peltier element and the thermistor are calibrated off-line (before the test is initiated) against the reference thermocouple, which has a known temperature response and is used in parallel to track the temperature of the DRAM module. Before starting the experiment, the DRAM module is heated up until it reaches the desired temperature (+85°C). The control loop implemented on the Arduino keeps the device temperature between +85°C and +86°C (+1°C range). If it goes above the temperature threshold the PE is switched off, when it drops below +86°C, it is immediately turned on again. This temperature check is performed five times (5X) per second.

To enable testing in the presence of power-supply noise, we introduce JET-5466 SODDR3 extender board [25], which connects the DDR3 DIMM to the ML605 board. The extender is equipped with a 100mΩ resistor that helps bring down the power-supply to 1.42V at peak current, which is the minimum supported operating voltage (maximum noise).

2) Testing the Fastest DIMM: With the worst-case operating conditions in effect, we employed the fastest DIMM (#6) from the observations in the first experiment, which indicated the most process-margin exploitation by our method, for verification. We first programmed the test patterns with the actual fastest (nominal-condition) timing measures of this DIMM on to the MicroBlaze and DRAM controller and triggered the memory test Algorithm 1. We observed that the test failed immediately, since these measures do not compensate for power-supply noise or temperature. We then programmed the test patterns with the rounded-up compensated timing measures and observed that the device worked correctly without any issues. We continued the test un-interrupted for over 4 hours under the worst-case operating conditions, transferring over 16TB of data to and from the memory in the process. At the end the test finished successfully, thereby verifying our derived temperature and noise compensations. The fastest and compensated timings (margins from Table V) for this DIMM are mentioned in Table VI.

TABLE VI: Test Measures and Results

Test Type	*nRCD (cc)*	*nRP (cc)*	*nRTP (cc)*	*nWR (cc)*	*Result*
Fastest	4	3	3	1	FAIL
Margins	13.3%	13.5%	20.55%	14.71%	-
Compensated	5	4	4	2	PASS

3) Addressing Side-Effects: As a result of reducing these timing measures, we risk increasing the DRAM power consumption. However, since we are only exploiting process-margins, the current measures, should *NOT* go higher than the datasheet current values, which also include process-margins [9]. We can verify this by measuring the voltage drop across the 100mΩ current-sensing resistor (CSR) on the JET-5466 extender board at +85°C. A high-end Lecroy Wavesurfer 454 Oscilloscope (2 GS/s) reporting at 500MHz was used for the measurements.

The currents affected by our optimization of the four critical analog timings include: (1) activation-precharge current (I_{DD0}) [10], (2) activation-read-precharge current (I_{DD1}) [10], and (3) activation-write-precharge current (I_{DD1W}). I_{DD1W} is not a standard JEDEC current measure, however, its reference can be calculated by substituting write current (I_{DD4W}) instead of read current I_{DD4R} in I_{DD1} current using [27].

Also, the datasheet current measures from Micron do not include I/O power consumption, however the real measurements on the JET-5466 board do. We resolve this by employing the Micron power calculator [26] estimates for I/O power consumption and add them to their datasheet I_{DD} measures for a fair comparison. The impact on current measures for the fastest (most exploited) DIMM are presented in Table VII.

TABLE VII: Impact on Current Measures

Current Type	*Measured (incl. I/O) (mA)*	*Datasheet (incl. I/O) (mA)*
I_{DD0}	300	360
I_{DD1}	524	584
I_{DD1W}	356	460

The results show that the current measures, despite shrinking the timings, remain lower than the datasheet estimates. This is as expected, since we are not violating any noise/temperature margins or the actual analog delays for these operations. We are merely conservatively identifying the excessive process-margins associated with these delays and eliminating them partially depending on the actual impact on a given DRAM device.

C. Impact on Energy, Bandwidth and Latency

In our final experiment, we analyze the benefit of using our timing optimizations on the energy consumption, latency and net bandwidth of read and write operations. Towards this, we employ the compensated timings of the fastest DIMM, identified in the first experiment and verified in the second. The latency and bandwidth measures are derived based on the reduced read and write latencies for 64-bytes accesses, while the energy measures are computed using the datasheet current measures [14] with the DRAMPower tool [28]. The graph in Figure 4 shows the gains by using the optimized (Opt) measures instead of the original (Orig) datasheet measures. As can be noticed from the analysis, our timing optimizations improve all three metrics significantly. The DRAM access latency reduces by 33.3% for reads and 25.9% for writes. The worst-case memory bandwidth (BW) also increases by 50% for reads and 35% for writes.

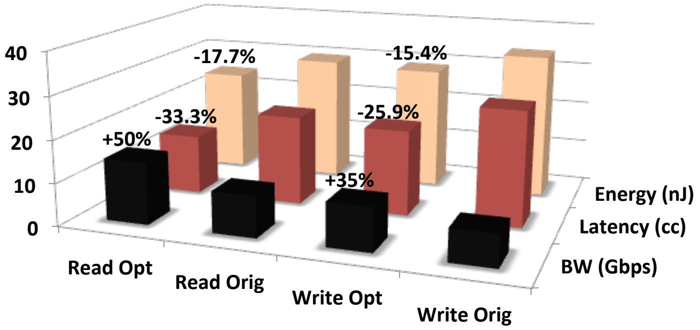

Fig. 4: Optimized Energy, Bandwidth and Latency

The DRAM energy consumption also reduces by 17.7% when reading and by 15.4% when writing, resulting in improved energy-efficiency. This highlights the significance of our proposed DRAM performance characterization methodology.

V. Related Work

When it comes to studying the impact of process variation in DRAMs, Intel observed performance degradation and power variation in DRAM memories in [29]. However, their test mechanisms are not publicly available. Gottscho *et al.* in [30] also observed variations of around 15% in power consumption across several 1GB DIMMs from the same vendor. However, they did not test the DIMMs for variations in timings and performance. Bathen *et al.* in [31], [32] employed these observations and suggested memory mapping and partitioning solutions to exploit this variability, but also did not verify performance variations. Desai *et al.* in [33] performed Monte-Carlo analysis on a single DRAM cell and basic circuit components to estimate the variation impact for an entire DRAM memory. However such an extrapolation is at best, a coarse approximation.

In the context of DRAM timing and functionality testing, authors of [15] and [16] proposed industrial DRAM tests and DRAM fault models. However, they only employed default DRAM timings in their tests. JEDEC proposed I_{DD} tests [10] for functional testing of the DRAM under worst-case conditions, but they also didn't change the timing measures and only tested a few pre-selected rows in the memory. Memtest86 [17] also did not alter the DRAM timings and only verified if the DRAM accepts and correctly retains arbitrary set of data written to it. Additionally, its testing is not guaranteed to stress the memory, since it depends on the underlying processor/cache architecture. Authors in [18] tested the retention capacity of DRAM cells to optimize refreshes, but did not test the timings of other DRAM operations. Moreover, none of these tests considered the timing margins introduced due to process, temperature or voltage variations. In [9], we derived impact of process variations on DRAM currents, and suggested using JEDEC's I_{DD1} test under nominal test conditions, without changing all the critical timing measures or testing the entire memory. In this paper, we proposed a comprehensive test methodology that varies all critical DRAM timings and tests the entire memory with different datasets and testsets under worst-case operating conditions.

VI. Conclusions

In this paper, we proposed a generic post-manufacturing DRAM characterization methodology that prunes the expendable process-margins in the datasheet timings of a DRAM device, while retaining the requisite noise and temperature margins. We demonstrated and evaluated this methodology for 48 DDR3 devices (from 12 identical DIMMs), derived their actual timings and verified their correct functionality under worst-case power-supply noise and temperature conditions. As a result, we achieved up to 33.3% and 25.9% reduction in read and writes latencies, respectively. We also improved the energy-efficiency of reads by up to 17.7% and writes by up to 15.4% and the worst-case bandwidth of reads by up to 50% and writes by up to 35%. The test bitstream used in our experiments for the ML605 board will be made available online at [28].

VII. Acknowledgements

This research was supported by EU FP7-288008 T-CREST, FP7-288248 FlexTiles, CA104 Cobra, CA505 BENEFIC, NL STW10346 NEST, DFG SPP 1500 & Czech Ministry of Education CZ.1.07/2.3.00/30.0034. We also appreciate the help of fellow researchers at TU Delft, TU Eindhoven and University of York for testing our methodology on their ML605 boards.

References

[1] W.A.Wulf *et al.*,*Hitting the memory wall: implications of the obvious*, In ACM SIGARCH Comp. Arch. News, Vol. 23, No. 1, 1995.
[2] JEDEC SST Assn., *DDR4 Standard*, JESD79-4, 2012.
[3] JEDEC SST Assn., *Wide I/O SDR Standard*, JESD229, 2011.
[4] B.Jacob *et al.*, *Memory Systems: Cache, DRAM, Disk*, Morgan Kaufmann Publishers, 2008.
[5] J.Vollrath, *Signal margin analysis for DRAM sense amplifiers*, In Proc. DELTA 2002.
[6] Micron, *TN-00-18: Temperature Uprating on Semiconductors*, 2010.
[7] S.Nassif *et al.*, *Delay variability: sources, impacts and trends*, In Proc. ISSCC 2000.
[8] 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.
[9] K.Chandrasekar *et al.*, *Towards Variation-Aware System-Level Power Estimation of DRAMs: An Empirical Approach*, In Proc. DAC 2013.
[10] JEDEC SST Assn., *DDR3 Standard*, JESD79-3E, 2010.
[11] Micron, *1Gb: X4, X8, X16 DDR3 Datasheet*, 2010.
[12] Il-Gweon Kim *et al.*, *DRAM reliability characterization by using dynamic operation stress in wafer burn-in mode*, In Proc. IRPS 2003
[13] T.Sato *et al.*, *Dynamically reducing overestimated design margin of MultiCores*, In Proc. HPCS 2012.
[14] Micron, *512MB (x64, Single Rank) 204-Pin DDR3 SDRAM SODIMM*, MT4JSF6464H, 2009.
[15] Z.Al-Ars *et al.*, *Space of DRAM Fault Models and Corresponding Testing*, In Proc. DATE 2006.
[16] A.J.Van De Goer *et al.*, *Industrial evaluation of DRAM tests*, In Proc. DATE 1999.
[17] MemTest86, *http://www.memtest86.com/*.
[18] J.Liu *et al.*, *An Experimental Study of Data Retention Behavior in Modern DRAM Devices: Implications for Retention Time Profiling Mechanisms*, In Proc. ISCA 2013.
[19] S.Goossens *et al.*, *A Reconfigurable Real-Time SDRAM Controller for Mixed Time-Criticality Systems*, In Proc. CODES+ISSS 2013.
[20] Xilinx Memory Interface Generator and Physical Layer (PHY), UG086, User Guide, 2010.
[21] Multicomp, *Peltier Cooler - 4W*, MCPF-031-10-25.
[22] JEDEC SST Assn., *Integrated Circuits Thermal Test Method Environment Conditions*, JESD51, 1995.
[23] Micron, *TN-00-08: Thermal Applications*, Tech. Notes, 2008.
[24] Arduino Platform, *http://www.arduino.cc/*.
[25] JU-JET electronics, JET-5466 Extender, *http://www.eztest.com.tw/*
[26] Micron Tech. Inc., *System Power Calculator for DDR3*, 2007.
[27] K.Chandrasekar *et al.*, *Improved Power Modeling of DDR SDRAMs*, In Proc. DSD 2011.
[28] K.Chandrasekar *et al.*, *DRAMPower: Open-source DRAM power and energy estimation tool*, *www.drampower.info*
[29] Intel, *Memory 3-sigma Power Analysis Methodology.*
[30] M.Gottscho *et al.*, *Power Variability in Contemporary DRAMs*, IEEE Embd. Sys. Letters, Vol. 4, No. 2, 2012.
[31] L.Bathen *et al.*, *Vamv: Variability-aware memory virtualization*, In Proc. DATE 2012.
[32] L.Bathen *et al.*, *ViPZonE: OS-Level Memory Variability-Driven Physical Address Zoning for Energy Savings*, In Proc. CODES+ISSS 2012.
[33] S.Desai *et al.*, *Process Variation Aware DRAM Design Using Block Based Adaptive Body Biasing Algorithm*, In Proc. ISQED 2012.