Reducing DRAM Latency via *In-situ* Temperature- and Process-Variation-Aware Timing Detection and Adaption

Yuxuan Qin[1†], Chuxiong Lin[1†], Mingche Lai[2], Zhang Luo[2], Shi Xu[3*], and Weifeng He[1*]
[1]Department of Micro-Nano Electronic, Shanghai Jiao Tong University, Shanghai, China
[2]College of Computer, National University of Defense Technology, Changsha, China
[3]Artificial Intelligence Research Center, National Innovation Institute of Defense Technology, Beijing, China
*Corresponding authors: hewf@sjtu.edu.cn, and xushi9018@aliyun.com

ABSTRACT

Long DRAM access latency has a significant impact on modern system performance. However, the improvement of DRAM access latency is limited, as the DRAM vendors reserve considerable timing margins against seldom worst-case conditions. To mitigate such pessimistic timing margins, we propose a temperature- and process-variation-aware timing detection and adaption DRAM (TPDA-DRAM) architecture. It equips *in-situ* cross-coupled detectors to monitor the voltage difference between bitline pairs, enabling estimation of timing margins caused by process and temperature variations. Moreover, TPDA-DRAM incorporates two collaborative timing adaption schemes: 1) a process-variation-aware timing adaption scheme (PVA) that selectively accelerates the access to weak cells, and 2) a temperature-variation-aware timing adaption scheme (TVA) that precisely adjusts timing parameters by adopting temperature information. Compared to prior art, the proposed detector reduces detection deviation by 54.8% and area overhead by 88.1%. The system-level evaluation in an eight-core system shows that TPDA-DRAM improves the average performance and energy efficiency by 20.5% and 15.0%, respectively.

KEYWORDS

DRAM, access latency, *in-situ* timing detection, adaptive timing, overdrive

1 Introduction

In modern computing systems, long DRAM access latency has become a critical performance bottleneck as a processor has to consume hundreds of clock cycles to access data in DRAM. However, over the past decades, the DRAM latency has only improved by 16.7% [1, 2], while its capacity and bandwidth have grown by more than two orders of magnitude.

†Equal contribution. This research was supported by in part by National Key R&D Program of China (No.2023YFB4403502), and NSFC (Grant No. 62274106)

DAC '24, June 23–27, 2024, San Francisco, CA, USA

ACM ISBN 979-8-4007-0601-1/24/06…
https://doi.org/10.1145/3649329.3656228

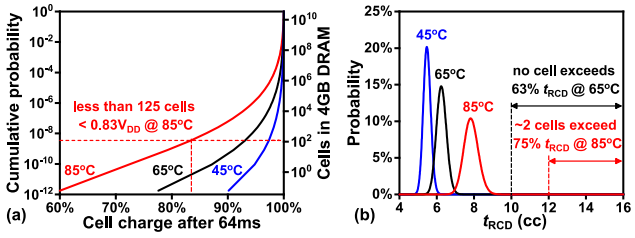

Figure 1: The distribution of (a) cell charge after 64ms and (b) corresponding t_{RCD} based on SPICE simulation.

A major barrier to reducing DRAM latency is the substantial timing guardbands reserved in timing parameters, which ensure correct DRAM operation under seldom worst-case conditions. However, under typical conditions, these timing guardbands are too pessimistic for most DRAM cells. For example, as shown in Figure 1a, the process variation leads to few cells with poor charge retention, yet almost all cells sustain over 83% of charge after 64ms [3]. Besides, although the timing guardbands are reserved for the worst temperature of 85°C [1, 2], DRAM latency is much lower at a typical working temperature below 65°C [4–6]. Figure 1b shows that such process and temperature variations lead to over 37% timing margins compared to the worst case. Therefore, mitigating these timing margins is essential to reduce DRAM latency.

Many prior works have expedited DRAM access by mitigating timing margins. These works typically profiled the DRAM to reduce process-induced timing margins [3, 7–9] or temperature-induced timing margins [4]. However, profiling is time-consuming and must run periodically due to the phenomena like variable retention time (VRT) and long-term aging, leading to non-negligible performance loss [10–12]. Moreover, profiling operates under static conditions, overlooking dynamic variations like fluctuating temperature. Recent works [5, 6] eliminated the timing margins for dynamic variations by using an *in-situ* charge detector that monitors the restoration time of DRAM cells at runtime. However, their skew-inverter-based detector is sensitive to process variations, requiring large transistor sizes and higher supply voltage to achieve sufficient timing detection accuracy.

In this work, we aim to aggressively reduce the DRAM access latency by eliminating timing margins caused by process and temperature variations simultaneously. To achieve this goal, we introduce a Temperature- and Process-variation-aware timing Detection and Adaption DRAM (TPDA-DRAM) architecture, which features novel *in-situ* cross-coupled detectors and two collaborative timing adaption schemes. The detector monitors the

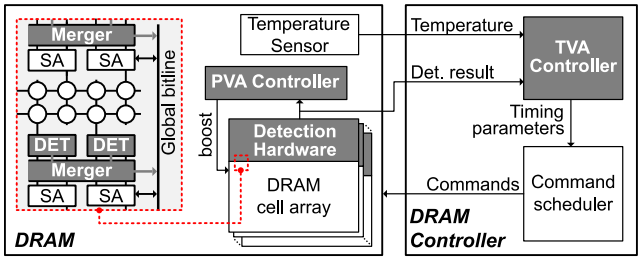

Figure 2: The proposed TPDA-DRAM architecture.

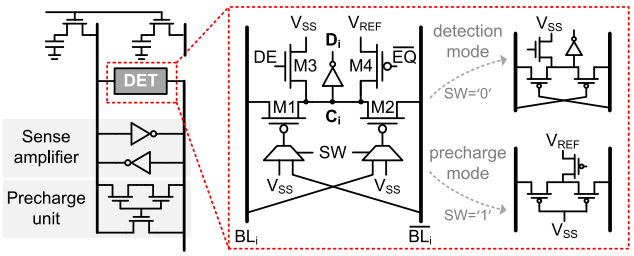

Figure 3: The hardware for *in-situ* cross-coupled detector.

voltage difference between a pair of bitlines, providing an estimation of the cell access latency. Moreover, since the precharge operation also largely contributes to access latency, we reuse the detector to expedite the precharge operation. Compared to prior works [5, 6], the proposed detector estimates cell access latency with 54.8% less deviation, occupies 88.1% less area, and reduces precharge latency by 62.5%.

With the estimation of access latency, TPDA-DRAM integrates two collaborative timing adaption schemes to accelerate DRAM access accordingly. Firstly, a process-variation-aware timing adaption (PVA) scheme uses the overdriven sensing technique [13] to boost the sense amplifiers, thereby speeding up access to seldom weak cells. By enabling weak cells to achieve the same speed as typical cells, PVA mitigates process-induced timing margins and allows the DRAM to adopt a unified set of reduced timing parameters, simplifying the control of timing parameters. Secondly, a temperature-variation-aware timing adaption (TVA) scheme adjusts timing parameters at runtime based on the estimated latency and temperature information, thus reducing temperature-induced timing margins. The evaluation in an eight-core system shows that TPDA-DRAM improves system performance by 20.5% and energy efficiency by 15.0% on average.

The main contributions of the paper are summarized as follows:

- We present an *in-situ* cross-coupled detector that exploits the voltage difference across the bitline pair to infer DRAM access latency with 54.8% less deviation and 88.1% less area compared to the prior art. The detector can also be reused to accelerate the precharge operation by 62.5%.
- We propose a PVA scheme that accelerates access to rare weak cells, mitigating process-induced timing margins. The scheme allows DRAM to adopt a uniform set of timing parameters, simplifying the control of timing parameters.
- We propose a TVA scheme that exploits the latency estimation and temperature information at runtime to accurately mitigate temperature-induced timing margins.
- We comprehensively evaluate the TPDA-DRAM. Our evaluation shows that TPDA-DRAM improves both the system performance and energy efficiency with the synergistic collaboration of its TPA and TVA schemes. Moreover, we discuss the compatibility of TPDA-DRAM with other relevant works.

2 TPDA-DRAM Architecture

In this section, we present the TPDA-DRAM architecture, which integrates two timing adaption schemes, PVA and TVA, based on a timing detection mechanism, as shown in Figure 2. Within each DRAM cell array, we integrate a row of *in-situ* cross-coupled detectors. Each detector is placed across a bitline pair and produces an output transition when the bitline voltage changes, thereby indicating the latency of the accessed cell. These detectors' outputs are then merged via a merger block to generate a timing representative of the accessing row. Based on the detection results, a PVA controller triggers a higher supply voltage to boost sense amplifiers when accessing weak cells, thereby mitigating process-induced timing margins. Concurrently, TVA fetches temperature information and detection results during refresh operations and leverages these to accurately adjust timing parameters, thereby mitigating temperature-induced timing margins.

2.1 *In-situ* Cross-Coupled Detector

To provide timing estimation, as shown in Figure 3, TPDA-DRAM integrates a cross-coupled detector next to each sense amplifier and precharge unit. TPDA-DRAM also reuses the detector to expedite the precharge operation. To support both the timing estimation and precharge acceleration, the detector switches flexibly between a timing detection mode and a precharge mode.

The cross-coupled detector is made up of ten transistors, including two multiplexers for mode switching, an inverter for output generation, and four transistors (M1-M4).The transistors M1 and M2 are stacked between BL and $\overline{\text{BL}}$. By configuring the multiplexers, the gates of M1 and M2 can be either connected to bitlines for timing estimation or V_{SS} for precharge acceleration. In the detection mode, M3 precharges the internal node C_i to V_{SS} and the inverter generates the detection result. In the precharge mode, M4 activates to expedite the restoration of the voltage on BL and $\overline{\text{BL}}$ back to $V_{DD}/2$.

The timing detection starts along with a refresh operation to hide its performance penalty. As shown in Figure 4a, assuming the cell stores "1", the refresh operation begins by sharing the cell's charge with BL. Simultaneously, the detector precharges its internal node C_i to V_{SS}. Then, the voltage difference between BL and $\overline{\text{BL}}$ enlarges as the SA starts to amplify. When the voltage difference surpasses a predefined voltage threshold (i.e., ΔV_{DET}), the C_i voltage arises as M1 is turned on. As a result, the detector's output D_i transits low at the moment of t_{DET}.

As a result, the t_{DET} serves as an indicator of the cell's t_{RCD}, which is defined as the time when the voltage difference between BL and $\overline{\text{BL}}$ reaches a ready-to-read level (typically $0.75V_{DD}$) [5–8].

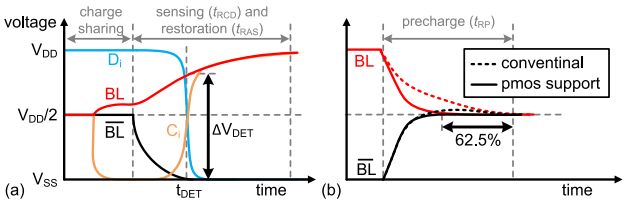

Figure 4: (a) Waveform of timing detection and (b) bitline precharge with the hardware of the precharge unit.

Furthermore, t_{DET} also serves as an indicator of the restoration time (i.e., t_{RAS}), which is proportional to the sensing time t_{RCD}. According to our circuit-level SPICE simulation, by setting a proper ΔV_{DET}, t_{DET} can approximate t_{RCD} and t_{RAS} within a deviation of less than 2% and 3%.

We can also reuse the detector to expedite the precharge operation. In a normal precharge operation, a precharge unit, which consists of three NMOS transistors, recovers the voltage of BL and $\overline{BL}$ to $V_{DD}/2$. However, as shown in Figure 4b, the time to precharge BL from V_{DD} to $V_{DD}/2$ is significantly longer than that for pulling up $\overline{BL}$ from V_{SS} to $V_{DD}/2$, dominating the precharge time (i.e., t_{RP}). That is due to the unbalanced strength of the two NMOS transistors in the precharge unit.

To accelerate the precharge operation, we reuse the PMOS transistors M1 and M2 of the detector and add M4 to provide complementary driving strength for precharging BL and $\overline{BL}$. Based on circuit simulation results, the precharge latency is reduced by 62.5% with the detector in precharge mode.

We compare the proposed cross-coupled bitline detector (denoted as CC-BD) with the prior skew-inverter-based bitline detector (denoted as Skew-BD) [5, 6], as shown in Table 1. Each Skew-BD uses a pair of skew-inverter (6 transistors in total) for detecting one bitline, whereas a single CC-BD detects two complementary bitlines, halving the detector requirement. In addition, Skew-BD needs a higher supply voltage and larger transistor sizes because it is sensitive to process variations. In contrast, CC-BD estimates timing with 54.8% higher accuracy with normal supply voltage and smaller transistor sizes. Furthermore, CC-BD can accelerate precharge operations for DRAM performance improvement.

Table 1: Comparison between the proposed CC-BD and Skew-BD [5, 6].

	Skew-BD	CC-BD	Reduction
Transistors	12	10	16.7%
Area (um^2)	0.360	0.043	88.1%
Dynamic power (nW)	111.93	63.02	43.7%
Static power (pW)	662.64	14.24	97.9%
Variation (ns)	0.305	0.138	54.8%

2.2 PVA for Process-Induced Margins

Based on the detection mechanism, we propose PVA to leverage the untapped process-induced timing margins for most DRAM

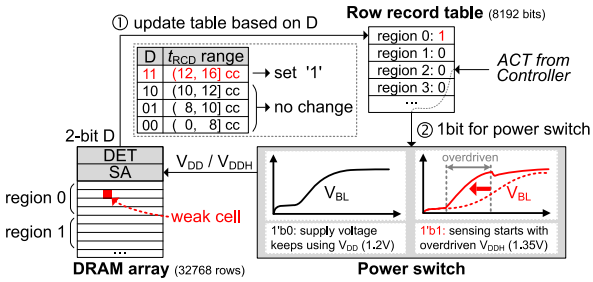

Figure 5: The workflow of PVA.

cells. The key idea of PVA is to speed up the access of rare weak cells, aligning their latency with the majority of other cells. This uniformity of latency then allows TPDA-DRAM to use unified timing parameters across all cells, mitigating the process-induced timing margins. To do so, we exploit the overdriven sensing scheme [13], a hardware acceleration scheme that temporarily applies a higher supply voltage (i.e., V_{DDH}) for the SA, which has been adopted in commodity DRAM to speed up the access operation [14]. However, unlike the conventional overdriven scheme that accelerates all cells uniformly, we selectively boost the voltage only for rare weak cells to prevent them from being a performance bottleneck with negligible power consumption.

TPDA-DRAM traces the weak cells that need to be boosted in a row record table. As the DRAM requires 8192 refresh operations to refresh all rows, the record table contains 8192 items to trace the location of weak cells. Each item contains only 1 bit to indicate whether the corresponding row region contains any weak cells.

The record table is updated according to the *in-situ* timing detection during each refresh operation. As shown in Figure 5, the detection hardware generates a result D, which indicates a possible t_{RCD} range for a DRAM row. For example, a D of 2'b11 indicates that the corresponding t_{RCD} exceeds 12cc. We record that by setting the corresponding bit in the table to 1'b1 (①) to mark that the row region contains at least one weak cell.

As a result, when accessing those rows containing weak cells, TPDA-DRAM uses V_{DDH} at the early sensing stage (②). Otherwise, TPDA-DRAM keeps the nominal voltage. This scheme reduces weak cells' t_{RCD} to less than 12cc, preventing weak cells from becoming the performance bottleneck. Moreover, the power required for applying V_{DDH} is negligible since only 0.02% of the rows require overdriven sensing.

With the overdriven sensing for weak cells, PVA effectively mitigates timing margins caused by process variations. This approach eliminates the need for specific timing parameters for weak cells and thereby simplifies the timing control.

2.3 TVA for Temperature-Induced Margins

To mitigate temperature-induced timing margins, we further propose a TVA scheme to adjust timing parameters in runtime. As shown in Figure 6, TVA exploits the timing detection mechanism to trace the DRAM's up-to-date worst-case latency (①). The up-to-date worst-case latency is stored by a 2-bit register

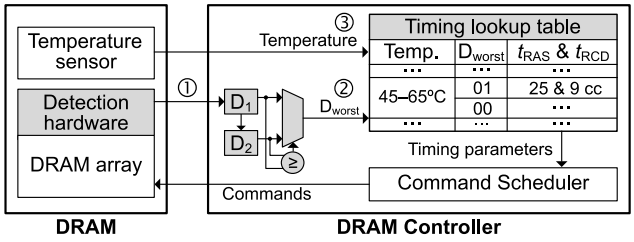

Figure 6: The workflow of TVA.

D_1. However, it takes 64ms to traverse the up-to-date latency of all rows. Before D_1 has been completely traversed for all rows, we avoid a false positive D_1 value by recording the worst-case latency in the last 64ms. It can be simply implemented by using another 2-bit register D_2, which samples the completed D_1's value every 64ms. As a result, the up-to-date worst-case latency should be the larger value of D_1 and D_2 (②), defined here as D_{worst}.

Given that the 2-bit D_{worst} provides only the potential timing range for the DRAM cells, we adopt the temperature information from the integrated temperature sensor to infer the timing range more precisely (③). The temperature information is readily achievable in today's commodity DRAM like DDR4 [15]. With D_{worst} and temperature at runtime, the DRAM controller can refer to a predefined lookup table for proper timing parameters.

As shown in Table 2, the predefined table categorizes timing parameters into three temperature levels, each of which is subdivided by the 2-bit detection results D_{worst}. For example, when the temperature is 85°C and D_{worst} is 2'b01, the t_{RCD} is reduced from 16cc to 10cc. At a most likely temperature of 65°C, D_{worst} may be still 2'b01 but indicates a further 10% reduction in t_{RCD} as compared to that of 85°C, bringing more performance benefits.

In the proposed TPDA-DRAM architecture, the detection mechanism, PVA, and TVA collaboratively mitigate the process- and temperature-induced timing margins. As a result, TPDA-DRAM reduces t_{RAS} and t_{RCD} by up to 47% and 50%, respectively. Reusing the detection hardware for precharge acceleration also reduces t_{RP} by 62.5%.

Table 2: The timing parameters of TPDA-DRAM.

Temperature	D_{worst}	t_{RAS} (cc)	t_{RCD} (cc)
65 – 85°C	**11**	need overdriven sensing (PVA)	
	10	32	12
	01	29	10
45 – 65°C	**01**	25	9
	00	23	8
< 45°C	**00**	19	8

3 Evaluation Methodology

3.1 Circuit-Level Evaluation

We perform circuit-level SPICE simulations for the cross-coupled bitline detector to verify its detection accuracy and determine the timing parameters listed in Table 2. We use the 22nm Predictive Technology Model (PTM) [16] to model the DRAM MAT and the proposed detectors. The transistors are sized based on the Rambus DRAM model [17] and scaled to 22nm according to the ITRS roadmap [18]. To model the process variations, we adopt the device process variation based on [19] and add a 5% variation to the other parameters like the capacitance of the cell and bitline.

3.2 System-Level Evaluation

We evaluate the TPDA-DRAM architecture with system-level simulation tools to show its performance and energy efficiency on single- and eight-core systems. We run the evaluation with a cycle-accurate DRAM simulator, Ramulator [20], which uses application traces generated by Pin [21] as input.

The applications come from SPEC CPU2006, MemBen [22], and MSC [23]. We classify them into memory intensive (>5 misses-per-kilo-instruction, or MPKI) and memory non-intensive (≤5 MPKI) applications. In the eight-core system, we generate four categories of workloads, each involving 25%, 50%, 75%, and 100% memory intensive applications, respectively. The simulation lasts for at least one billion CPU cycles. We use the instruction-per-cycle (IPC) as the performance metric for the single-core system and the weighted speedup metric [24] for the eight-core system. Moreover, DRAMPower [25] is used to report the energy results.

Table 3 lists the system configurations for our evaluation.

Table 3: The system configuration.

Components	Parameters
Processor	1–8 cores, 4.3GHz, IPC=4, 128-entry ROB
Last-level cache	2 MB/core, 64B cache-line, 16-way
Memory Controller	FR-FCFS-Cap scheduling policy, timeout-based row policy, 64-entry read/write request queue
DRAM	DDR4, 2133MHz bus frequency, 4Gb chip density, $t_{RCD}/t_{RAS}/t_{RP}$ = 16/36/16 cc, 2 channels, 1 rank, 4 bank groups per rank, 4 banks per bank group, 32768 rows per bank
Ideal CDAR-DRAM [6]	Reduced refresh rate: 256ms, Truncated t_{RAS}/t_{WR} for 4 sub-windows: At 85°C: 36/16, 27/14, 23/12, 22/11cc, At 65°C: 23/16, 18/14, 17/12, 17/11cc, Reduce t_{RCD} for rows accessed recently: 12cc at 85°C and, and 8cc at 45/65°C

4 Evaluation Results and Discussion

There are many works [4–9, 28–35] that reduced DRAM latency for system performance improvement. We compare TPDA-DRAM to the CDAR-DRAM architecture [6], which also adapts timing parameters via *in-situ* detectors. CDAR-DRAM stores a timing tag for each row. For simplicity, we evaluated the ideal CDAR-DRAM that all cells have the best timing tag (see Table 3). We will analyze and discuss other related works in Section 4.4.

TPDA-DRAM and CDAR-DRAM are evaluated under two conditions: the worst operating temperature and the typical one. We set the worst temperature to 85°C according to DDR4 specifications [15]. We assume the typical temperature varies

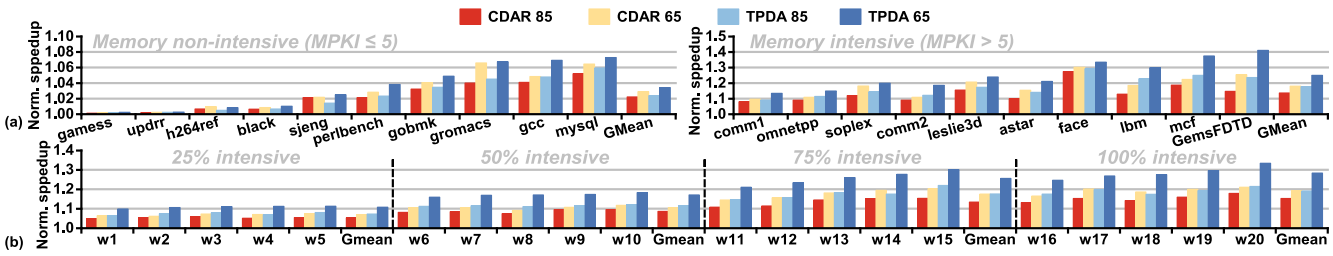

Figure 7: Speedup over the DDR4 baseline for (a) single-core and (b) eight-core workloads.

between 45°C and 65°C [4–6]. In the evaluation of the typical condition, the temperature is 45°C for half of the duration and 65°C for the remainder. The following schemes are evaluated:

- CDAR 85: prior CDAR-DRAM in the worst condition.
- CDAR 65: prior CDAR-DRAM in the typical condition.
- TPDA 85: TPDA-DRAM in the worst condition.
- TPDA 65: TPDA-DRAM in the typical condition.

4.1 Impact on Performance

TPDA-DRAM reduces the DRAM access latency by mitigating timing margins caused by process and temperature variations, thus improving system performance. Under single-core workloads, TPDA-DRAM exhibits more benefits on memory intensive workloads compared to memory non-intensive ones. As shown in Figure 7a, TPDA 65 (85) has an average speedup of 2.2% (1.4%) for memory non-intensive workloads but 25.1% (17.8%) for memory intensive ones. This is because intensive workloads have more frequent DRAM accesses, which makes them more sensitive to reduced timing parameters. Compared to CDAR 65 (85), TPDA 65 (85) achieves an average of 3.4% (2.0%) higher speedup among all single-core workloads. That is because i) the proposed detector in TPDA-DRAM not only identifies *in-situ* timing parameters like CDAR-DRAM but also aids in reducing the precharge latency; ii) TPDA-DRAM accelerates all accesses with a unified reduced timing parameter while CDAR-DRAM only adopts the reduced timing parameters for recently accessed rows.

Under eight-core workloads, the performance benefits of TPDA-DRAM increase as workload memory intensity increases. As shown in Figure 7b, under 25%, 50%, 75%, and 100% intensive workloads, TPDA 65 has an average speedup of 8.7%, 17.2%, 25.6%, and 28.4%, respectively. Moreover, TPDA 65 outperforms CDAR 65 and its speedup over CDAR 65 also increases as the workload becomes more intensive. For example, TPDA 65 exhibits a 3.8% speedup over CDAR 65 for 25% intensive workloads and 7.6% for 100% intensive workloads. That is because CDAR-DRAM only focuses on reducing activation latency while TPDA-DRAM reduces the latency for both precharge and activation operations, which are increasingly critical with higher intensity of access.

4.2 Impact on Energy Efficiency

TPDA-DRAM also improves the DRAM energy efficiency as it completes more requests within the same period. In Figure 8, the energy breakdown, normalized to the DDR4 baseline, is depicted

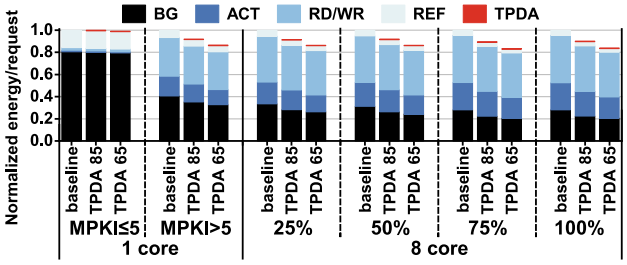

Figure 8: Energy breakdown for TPDA-DRAM.

across various workload categories. For memory non-intensive single-core workloads, different architectures show similar energy efficiency since the DRAM is rarely accessed and the background energy dominates. In contrast, for memory intensive workloads, TPDA-DRAM improves the energy efficiency by 8.0% at 85°C and 13.5% at 65°C because it greatly reduces the activation energy with less activation time. Similarly, for eight-core workloads, TPDA-DRAM improves the energy efficiency by 9.2% at 85°C and 15.0% at 65°C on average. Note that the overdriven sensing scheme and the detection hardware occupy only 0.6% of the total energy, which is negligible as compared to the energy reduction from reduced activation time. In conclusion, TPDA-DRAM can improve energy efficiency for various workloads.

4.3 Area Overhead

TPDA-DRAM requires additional hardware in a DRAM, including cross-coupled detectors, the merger block, and a 1KB storage for tracing the weak cells. To estimate its area overhead, we utilize DRAMSpec [26] with DDR4 parameters to evaluate the area of DDR4 DRAM. Then we estimate the area overhead of detectors by calculating the area ratio between the detectors and the SAs as shown in Equation 1.

$$Area\ Overhead = \frac{\sum W_{detector} \times L_{detector}}{\sum W_{SA} \times L_{SA}} \times \frac{S_{SA}}{S_{chip}} \quad (1)$$

Moreover, the area overhead of the merger block is conservatively estimated after a place and route (P&R) flow with 28nm CMOS technology. Based on this conservative estimation, the area overhead of the detectors and the merger circuits are 1.0% and 1.4%, respectively. Furthermore, our evaluation indicates that the 1KB row record table incurs a 0.02% overhead with the cache modeling tool CACTI [27]. Therefore, the total area overhead of the proposed TPDA-DRAM architecture is approximately 2.42%.

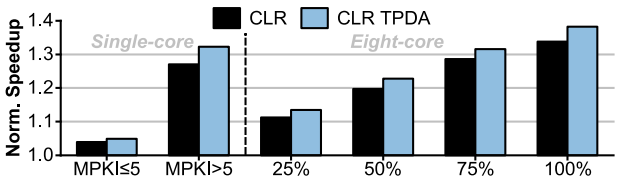

Figure 9: Performance of CLR-DRAM and the combined structure of CLR-DRAM and TPDA-DRAM.

4.4 Related Works

Many previous works reduced the DRAM latency through different techniques, such as 1) mitigating timing margins due to PVT variations, 2) exploiting charge level in the cell, and 3) modifying the DRAM structure.

Some works mitigated timing margins by profiling DRAM in a static condition [4, 7–9]. However, the profiling process incurs non-negligible performance overhead [10–12]. Previous works attempted to reduce this overhead by using ECC [11] or profiling under aggressive conditions [12]. Nevertheless, DRAM profiling still demands a stable condition that is hard to achieve in a running system and thereby overlooks timing margins resulting from dynamic variations. In contrast, TPDA-DRAM detects the timing without stalling DRAM operations and mitigates timing margins caused by both static and dynamic variations.

Other works exploited the cell charge level to improve DRAM latency [28–31]. They achieved this by either reducing activation latency for recently refreshed cells [28], decreasing restoration latency for soon-to-be-refreshed cells [29, 30], or eliminating refresh latency for recently accessed cells [31]. These works have explored the access pattern to reduce DRAM access latency, while the TPDA-DRAM focuses on leveraging the timing margins. Consequently, these works are complementary to TPDA-DRAM and can potentially be combined for better system performance.

Numerous works modified the DRAM structure to reduce the DRAM latency [32–35]. They typically achieved this by adopting smaller DRAM cell arrays [32, 33], coupling two cells [34] or two sense amplifiers [35]. These structural modifications reduced the DRAM access latency without considering the temperature and process induced timing margins, which can be mitigated by combining with TPDA-DRAM. For example, the state-of-the-art CLR-DRAM [35] couples adjacent DRAM cells in a row along with their sense amplifiers to reduce access latency. In our evaluation, as shown in Figure 9, CLR-DRAM combined with TPDA-DRAM (i.e., CLR TPDA) achieves an additional average performance improvement of 2.4% for a single-core system and 2.6% for an eight-core system over CLR-DRAM.

5 Conclusion

In this paper, we introduce a TPDA-DRAM architecture to mitigate pessimistic timing margins due to temperature and process variations. TPDA-DRAM adopts an *in-situ* timing detection mechanism with robust cross-coupled detectors, which detect t_{RCD} with less than 2% deviation and accelerate precharge operation by 62.5%, with 88.1% less area than that of the prior art [6, 7]. We also reduce the t_{RAS} and t_{RCD} by up to 50% by mitigating timing margins with two timing adaption schemes, PVA and TVA. The evaluation shows that under typical operating conditions, TPDA-DRAM improves the average performance by 20.5% and energy efficiency by 15.0% in an eight-core system.

REFERENCES

[1] Samsung. K4D263238M 128M DDR SDRAM. 2001.
[2] JEDEC. DDR5 SDRAM standard, JESD79-5. 2020.
[3] J. Liu *et al.*, "RAIDR: Retention-aware intelligent DRAM refresh," In *ISCA*, 2012.
[4] D. Lee *et al.*, "Adaptive-Latency DRAM: Optimizing DRAM Timing for the Common-Case," In *HPCA*, 2015.
[5] C. Lin *et al.*, "CDAR-DRAM: An In-situ Charge Detection and Adaptive Data Restoration DRAM Architecture for Performance and Energy Efficiency Improvement," In *DAC*, 2021.
[6] Y. Qin *et al.*, "CDAR-DRAM: Enabling Runtime DRAM Performance and Energy Optimization via In-situ Charge Detection and Adaptive Data Restoration," *TCAD*, 2023.
[7] A. Agrawal *et al.*, "Mosaic: Exploiting the Spatial Locality of Process Variation to Reduce Refresh Energy in On-Chip eDRAM Modules," In *HPCA*, 2014.
[8] K. Chandrasekar *et al.*, "Exploiting Expendable Process-Margins in DRAMs for Run-Time Performance Optimization," In *DATE*, 2014.
[9] J. S. Kim *et al.*, "Solar-DRAM: Reducing DRAM Access Latency by Exploiting the Variation in Local Bitlines," In *ICCD*, 2018.
[10] J. Liu *et al.*, "An Experimental Study of Data Retention Behavior in Modern DRAM Devices: Implications for Retention Time Profiling Mechanisms," In *ISCA*, 2013.
[11] M. K. Qureshi *et al.*, "AVATAR: A Variable-Retention-Time (VRT) Aware Refresh for DRAM Systems," In *DSN*, 2015.
[12] M. Patel *et al.*, "The Reach Profiler (REAPER) Enabling the Mitigation of DRAM Retention Failures via Profiling at Aggressive Conditions," In *ISCA*, 2017.
[13] T. Takahashi *et al.*, "A Multigigabit DRAM Technology with 6F/sup 2/ Open-Bitline Cell, Distributed Overdriven Sensing, and Stacked-Flash Fuse," *JSSC*, 2001.
[14] H. -C. Shih *et al.*, "DArT: A Component-Based DRAM Area, Power, and Timing Modeling Tool," *TCAD*, 2014.
[15] JEDEC. DDR4 SDRAM standard, JESD79-4B. 2012.
[16] PTM, Predictive Technology Model, http://ptm.asu.edu.
[17] Rambus. DRAM Power Model (2010). http://www.rambus.com/energy, 2010.
[18] ITRS, "ITRS Reports," http://www.itrs2.net/itrs-reports.html.
[19] K. Chandrasekar *et al.*, "Towards Variation-Aware System-Level Power Estimation of DRAMs: An Empirical Approach," In *DAC*, 2013.
[20] Y. Kim *et al.*, "Ramulator: A Fast and Extensible DRAM Simulator," In *IEEE Computer Architecture Letters*, 2016.
[21] C. Luk *et al.*, "Pin: Building Customized Program Analysis Tools with Dynamic Instrumentation," In *PLDI*, 2005.
[22] S. Ghose *et al.*, "Demystifying Complex Workload-DRAM Interactions: An Experimental Study," In *SIGMETRICS*, 2019.
[23] N. Chatterjee *et al.*, "Memory Scheduling Championship (MSC)," 2012.
[24] A. Snavely and D. M. Tullsen, "Symbiotic Job Scheduling for a Simultaneous Multithreading Processor," in *ASPLOS-IX*, 2000.
[25] K. Chandrasekar *et al.*, "DRAMPower: Open-source DRAM Power & Energy Estimation Tool," *http://www.es.ele.tue.nl/drampower/*, 2012.
[26] C. Weis et al., "DRAMSpec: A High-Level DRAM Timing, Power and Area Exploration Tool," *Int J Parallel Prog*, 2017.
[27] N. Muralimanohar et al., "CACTI 6.0: A Tool to Model Large Caches," *HP Laboratories, Tech. Rep*, 2009.
[28] H. Hassan *et al.*, "ChargeCache: Reducing DRAM Latency by Exploiting Row Access locality," In *HPCA*, 2016.
[29] X. Zhang *et al.*, "Restore Truncation for Performance Improvement in Future DRAM Systems," In *HPCA*, 2016.
[30] Y. Wang *et al.*, "Reducing DRAM Latency via Charge-Level-Aware Look-Ahead Partial Restoration," In *MICRO*, 2018.
[31] M. Ghosh and H. -H. S. Lee, "Smart Refresh: An Enhanced Memory Controller Design for Reducing Energy in Conventional and 3D Die-Stacked DRAMs," In *MICRO*, 2007.
[32] K. K. Chang *et al.*, "Low-Cost Inter-Linked Subarrays (LISA): Enabling Fast Inter-Subarray Data Movement in DRAM," In *HPCA*, 2016.
[33] Y. Wang *et al.*, "FIGARO: Improving System Performance via Fine-Grained In-DRAM Data Relocation and Caching," in *MICRO*, 2020.
[34] H. Hassan *et al.*, "CROW: A Low-Cost Substrate for Improving Dram Performance, Energy Efficiency, and Reliability," in *ISCA*, 2019.
[35] H. Luo *et al.*, "CLR-DRAM: A Low-Cost DRAM Architecture Enabling Dynamic Capacity-Latency Trade-Off," In *ISCA*, 2020.