On the Optimal Refresh Power Allocation for Energy-Efficient Memories

Yongjune Kim*, Won Ho Choi*, Cyril Guyot*, and Yuval Cassuto*†
*Western Digital Research, Milpitas, CA, USA
Email: {yongjune.kim, won.ho.choi, cyril.guyot}@wdc.com
†Viterbi Department of Electrical Engineering, Technion – Israel Institute of Technology, Haifa, Israel
Email: ycassuto@ee.technion.ac.il

***Abstract*—Refresh is an important operation to prevent loss of data in dynamic random-access memory (DRAM). However, frequent refresh operations incur considerable power consumption and degrade system performance. Refresh power cost is especially significant in high-capacity memory devices and battery-powered edge/mobile applications. In this paper, we propose a principled approach to optimizing the refresh power allocation. Given a model for the bit error rate dependence on power, we formulate a convex optimization problem to minimize the word mean squared error for a refresh power constraint; hence we can guarantee the optimality of the obtained refresh power allocations. In addition, we provide an integer programming problem to optimize the discrete refresh interval assignments. For an 8-bit accessed word, numerical results show that the optimized nonuniform refresh intervals reduce the refresh power by $29\,\%$ at a peak signal-to-noise ratio of $50\,\mathrm{dB}$ compared to the uniform assignment.**

I. Introduction

Memory refresh is a periodically repeated procedure that reads and rewrites the data of a memory device to prevent loss of data. It is well known that dynamic random-access memory (DRAM) cells must be refreshed periodically due to charge leakage [1], [2]. A DRAM cell stores one bit of information by controlling the amount of charge on its capacitor. DRAM cells cannot retain their data permanently because of the gradual loss of charge over time. The time a cell can retain its data is called the *retention time* of the cell. The time interval between refresh operations is the *refresh interval*, which is the inverse of the *refresh rate*. A cell that cannot retain its data for the given refresh interval results in a failure, referred to as *retention failure (or retention error)* [3]–[5]. The typical refresh interval in current DRAM standards is $64\,\mathrm{ms}$, which is a conservative value [4], [5].

The conservative refresh operations lead to high refresh power consumption. This problem is expected to worsen as DRAM device capacity increases [1], [4]. As cell dimension shrinks, memory cells become susceptible to charge leakage and require more frequent refresh operations [5]. Further, the refresh power consumption is critical in battery-powered edge/mobile computing applications. Note that edge/mobile devices are idle most of the time and refresh operations are still required during idle periods unlike write and read operations [6].

Many refresh techniques were proposed to reduce refresh power [2]–[11]. Ohsawa et al. [3] and Ghosh et al. [7] proposed architectural techniques to avoid unnecessary refresh operations. Error control coding (ECC) schemes were proposed to decrease refresh rates and correct the resulting retention failures [2], [8]–[10]. These ECC schemes suffer from storage or bandwidth overheads. RAIDR [4] allocates different refresh intervals by identifying weak DRAM cells. Flikker [6] specifies critical and non-critical data and refreshes the memory cells storing non-critical data at a lower rate. Cho et al. [11] proposed tiered-reliability memory (TRM) to allocate different refresh intervals depending on the importance of bit positions. Since these previous techniques choose the refresh intervals empirically, the granularity of refresh interval assignments are inherently limited. Further, the optimality of refresh intervals has not been addressed.

We note that refresh is also considered in storage-class memories such as magnetic RAMs (MRAMs) and resistive RAMs (ReRAMs) [12]. For example, MRAMs suffer from high write latency and energy, which are the key drawbacks of MRAM technology. Several techniques [13], [14] attempt to address the write-inefficiency of MRAMs via relaxing retention time and introducing refresh operations. For the sake of concreteness, we focus on DRAM refresh, wherein refresh has been established as a central trade-off between power and fidelity.

This paper presents a *principled* approach to refresh interval assignments for machine learning (ML) and signal processing tasks. In these applications, the mean squared error (MSE) is a more meaningful fidelity metric than the bit error rate (BER). We formulate a convex optimization problem to minimize the MSE for a given refresh power constraint. Since the formulated problem is convex, the global optimal solutions can be obtained with standard convex programming algorithms. Even more favorably, we derive an analytic expression for the optimal solution using the Karush-Kuhn-Tucker (KKT) conditions. In addition, we formulate a discrete optimization problem by taking into account hardware implementation. Our evaluation shows that the penalty due to discrete intervals is marginal. A prior study in [15], [16] of voltage-swing optimization in static RAMs (SRAMs) is similar in spirit, but its results are not applicable to optimizing DRAM's refresh intervals. To the best of our knowledge, our work is the first rigorous treatment of the optimal refresh interval assignments, viz. refresh power allocations.

The rest of this paper is organized as follows. Section II explains the current DRAM architecture and refresh operations. Section III introduces the optimization metrics of DRAM's refresh power and fidelity. Section IV formulates optimization problems to determine the optimum refresh intervals and provides the theoretical analysis. Section V gives numerical results and Section VI concludes.

II. DRAM Architecture and Refresh Operations

A. DRAM Architecture

DRAM system is hierarchically organized channels, modules, ranks, and chips as shown in Fig. 1. Each memory channel drives commands, addresses, and data between a memory controller and one or more DRAM modules [5], [17]. Each module contains multiple DRAM chips that are organized into one or more ranks. A rank consists of multiple chips that operate synchronously to provide a wide data bus (e.g., 64-bit) to increase the bandwidth, as a single DRAM chip is designed to have a narrow data bus width (e.g., 8-bit) [17]. Each of the eight chips in the rank transfers 8 bits simultaneously in a unit interval of double-data rate (DDR) time frame to provide 64 bits of data as shown in Fig. 1(a).

A DRAM chip consists of multiple banks that can process DRAM commands independently to increase parallelism. A bank includes a memory array of DRAM cells that are organized into rows and columns, as shown in Fig. 1(b) [17]. A row consists of 1 KB or 2 KB cells in general and the number of rows depends on the chip capacity.

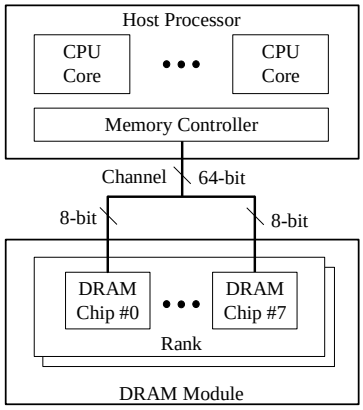
A cell has (i) a capacitor that stores binary data in the form of stored charge (e.g., charged and discharged states compared to a reference charge represent 1 and 0, respectively), and (ii) an access transistor that serves as a voltage-controlled switch to connect the capacitor to the bitline [5], [17]. DRAM cells in each column share a bitline, which connects them to a sense amplifier. The sense amplifier detects the charge stored in a cell and converts the charge to binary information. DRAM cells in each row share a wire called the wordline, which controls the corresponding cells' access transistors. When a wordline is enabled by the row decoder, the entire cells in the row get connected to the sense amplifiers through the bitlines, enabling the sense amplifiers to detect the data and latch them into the row buffer [17]. A chunk of the data in the row buffer is fetched out by the column decoder.

B. Refresh Operations

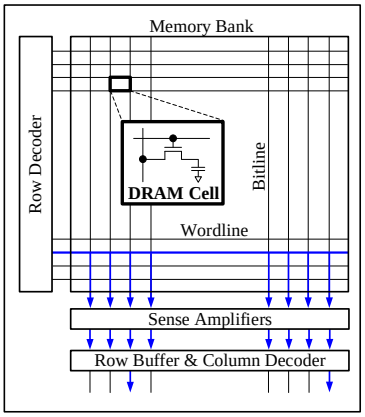
Since a DRAM cell capacitor leaks charge over time, the charge on each capacitor must be periodically refreshed. To prevent retention failure, the refresh interval should be less than the retention time. Since all memory cells do not have the same retention time because of process variations [1], [4], [18], the BER due to retention failure is given by

$$p = \Pr\left(T_{\text{retention}} < t\right), \tag{1}$$

where t denotes a given refresh interval value. The random variable $T_{\text{retention}}$ represents the retention time of DRAM cells. It is clear that shorter refresh intervals decrease the BER due to retention failure. To guarantee data integrity, current DRAM standards conservatively employ the refresh interval of 64 ms.



(a)



(b)

Fig. 1. Organization of DRAM system: (a) DRAM system and (b) DRAM bank architecture.

The refresh power P is inversely proportional to the refresh interval as follows [6], [19]:

$$P \propto \frac{C}{t}, \tag{2}$$

where C denotes the effective switching capacitance. This effective switching capacitance increases for higher-capacity DRAM devices. Hence, the refresh power consumption continues to increase as DRAM device capacity increases [1], [4], [19].

III. DRAM Optimization Metrics

The refresh interval t is a key parameter to control the trade-off between refresh power and fidelity. If we separate the data for each bit position in different subarrays by interleaving as in [11], [15], [16], then the corresponding refresh interval assignment is represented by a vector $\mathbf{t} = (t_0, \ldots, t_{B-1})$ as shown in Fig. 2. Note that t_0 and t_{B-1} represent the refresh intervals corresponding to least significant bit (LSB) and most significant bit (MSB), respectively. Subarrays can correspond to memory banks or memory chips depending on architecture configuration. Due to the current DRAM's multi-chip and

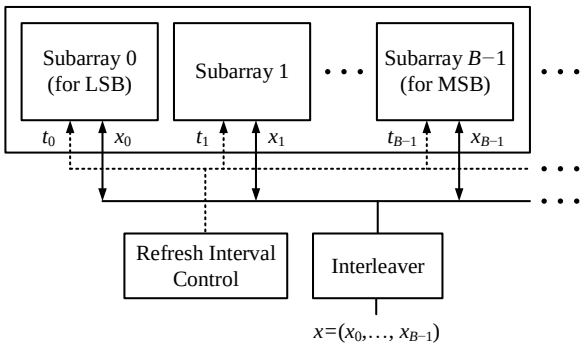


Fig. 2. Interleaved architecture [11] where $x = (x_0, \ldots, x_{B-1})$ denotes a stored B-bit word.

multi-bank architecture in Fig. 1, we can allocate different refresh intervals to each subarray with minimal hardware overhead [4], [6], [11].

In the following subsections, we describe the resource and fidelity metrics with the refresh interval assignment.

A. Resource Metric: Refresh Power

From (2), the normalized refresh power for a B-bit word is given by

$$\mathsf{P}(\mathbf{t}) = \sum_{b=0}^{B-1} \frac{1}{t_b}. \tag{3}$$

Remark 1: The refresh power $\mathsf{P}(\mathbf{t})$ is a *convex* function of $\mathbf{t}$ because $t_b > 0$ for $b \in [0, B-1]$.

B. Fidelity Metrics: BER and MSE

Suppose that p_b denotes the BER of the bth bit position. Since p_b is a function of refresh interval t_b, we set

$$p_b = g(t_b) \tag{4}$$

for $b \in [0, B-1]$.

In many signal processing and ML tasks, the impact of bit errors depends on the bit position. For example, errors in the MSB position of image pixels degrade overall image quality much more than errors in the LSB position. Likely, an MSB error can cause a catastrophic loss in the inference accuracy of ML applications [15]. Hence, we use the MSE as a fidelity metric instead of the BER.

The MSE of B-bit words is given by

$$\mathsf{MSE}(\mathbf{t}) = \sum_{b=0}^{B-1} 4^b g(t_b), \tag{5}$$

where the weight 4^b represents the differential importance of each bit position [15], [20].

Remark 2: $\mathsf{MSE}(\mathbf{t})$ is *convex* if $g(\cdot)$ is convex. It is because a nonnegative weighted sum of convex functions is convex.

It was reported that the BER increases exponentially with the refresh interval [5], [6], [11], [21]. Hence, we model the BER as

$$p_b = g(t_b) = \alpha \exp(\beta t_b), \tag{6}$$

TABLE I
RESOURCE AND FIDELITY METRICS FOR REFRESH OPERATION

	Single bit	B-bit word
Variable	t	$\mathbf{t} = (t_0, \ldots, t_{B-1})$
Refresh power	$\frac{1}{t}$	$\sum_{b=0}^{B-1} \frac{1}{t_b}$
Fidelity	$g(t)$	$\sum_{b=0}^{B-1} 4^b g(t_b)$

where positive values of α and β depend on the memory fabrication parameters.

Remark 3: $\mathsf{MSE}(\mathbf{t})$ is *convex* if $g(\cdot)$ is an exponential function as in (6).

Table I summarizes the resource and fidelity metrics for single-bit and B-bit word. We note that these metrics are convex.

IV. Formulation of Optimization Problems

A. Convex Optimization Problem

We formulate a convex optimization problem to determine the optimal refresh intervals. For a given refresh power constraint, we seek to minimize MSE as follows:

$$\begin{aligned} \underset{\mathbf{t}}{\text{minimize}} \quad & \mathsf{MSE}(\mathbf{t}) = \sum_{b=0}^{B-1} 4^b \alpha \exp(\beta t_b) \\ \text{subject to} \quad & \mathsf{P}(\mathbf{t}) = \sum_{b=0}^{B-1} \frac{1}{t_b} \le \mathcal{P} \\ & t_b \ge \delta, \quad b = 0, \ldots, B-1 \end{aligned} \tag{7}$$

where $\mathcal{P}$ is a constant corresponding to the given refresh power budget. Note that $\delta > 0$ denotes the conservative minimum refresh interval, which in particular prevents $t_b = 0$ (i.e., infinite refresh power). We set $\delta = 0.064$ based on current DRAM standards.

Because of Remark 1 and Remark 3, the optimization problem (7) is convex. Hence, we can obtain the global optimal solutions by standard convex programming algorithms. In addition, we can derive the optimal solution based on KKT conditions.

Theorem 4: The optimal refresh-interval vector $\mathbf{t}^*$ of (7) is given by

$$t_b^* = \begin{cases} \delta, & \text{if } \frac{\nu}{4^b} < \alpha\beta\delta^2 \exp(\beta\delta); \\ \frac{2}{\beta} W\left(\frac{\beta}{2}\sqrt{\frac{\nu}{4^b \alpha\beta}}\right), & \text{otherwise} \end{cases} \tag{8}$$

where ν is a dual variable of KKT conditions. Note that ν depends on the refresh power budget $\mathcal{P}$ for the given α and β. We can find ν efficiently by the bisection method as in [22]. Also, $W(\cdot)$ denotes the *Lambert W function*, which is the inverse function of $f(x) = xe^x$ [23].

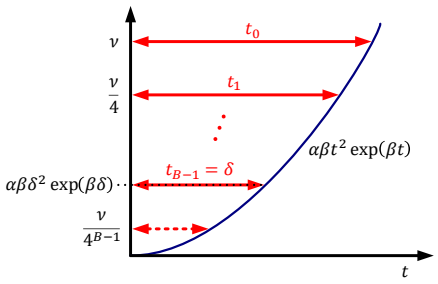


Fig. 3. A graphical interpretation of the optimal refresh intervals in Theorem 4.

Proof: We define the Lagrangian $L_1(\mathbf{t},\nu,\lambda)$ associated with problem (7) as

$$L_1(\mathbf{t},\nu,\lambda)=\sum_{b=0}^{B-1}4^b\alpha\exp(\beta t_b)+\nu\left(\sum_{b=0}^{B-1}\frac{1}{t_b}-\mathcal{P}\right)-\sum_{b=0}^{B-1}\lambda_b\left(t_b-\delta\right) \quad (9)$$

where ν and $\lambda=(\lambda_0,\ldots,\lambda_{B-1})$ are the dual variables. The optimal solution is derived from L_1 and the corresponding KKT conditions. The details of the proof are given in Appendix A. ∎

The optimal refresh interval (8) can be interpreted by Fig. 3. As shown in Appendix A, the condition of $\frac{\nu}{4^b}=\alpha\beta t_b^2\exp(\beta t_b)$ should be satisfied for any $t_b>\delta$ (i.e., $\frac{\nu}{4^b}>\alpha\beta\delta^2\exp(\beta\delta)$). If $\frac{\nu}{4^b}<\alpha\beta\delta^2\exp(\beta\delta)$, then the corresponding refresh interval is forced to $t_b=\delta$. As the refresh power budget $\mathcal{P}$ decreases, the dual variable ν is increased to allocate longer refresh intervals. If more refresh power is available, then ν is lower and the corresponding refresh intervals are reduced as shown in Fig. 3.

Note that $\mathbf{t}_0=(\delta,\ldots,\delta)$ corresponds to the maximum refresh power and the minimum MSE as follows.

Remark 5 (Maximum Refresh Power): The maximum refresh power is $\mathsf{P}_{\mathsf{max}}=\mathsf{P}(\mathbf{t}_0)=\frac{B}{\delta}$. If $B=8$ and $\delta=0.064$, then $\mathsf{P}_{\mathsf{max}}=125$.

Remark 6 (Minimum MSE): The minimum MSE is

$$\mathsf{MSE}_{\mathsf{min}}=\mathsf{MSE}(\mathbf{t}_0)=\frac{4^B-1}{3}\cdot\alpha\exp(\beta\delta) \quad (10)$$

which is obtained by the maximum refresh power. Note that the MSE increases exponentially with the refresh interval δ.

B. Discrete Refresh Intervals

In the previous subsection, we formulated the convex optimization problem by assuming that any real values can be assigned to refresh intervals. Here, we investigate the discrete-valued refresh interval optimization. If the optimized discrete refresh intervals are multiples of δ (e.g., 64 ms), then the proposed optimization technique is compatible with current DRAM products. It is because any multiple of δ can be set as a refresh interval by gating the refresh commands [3], [4].

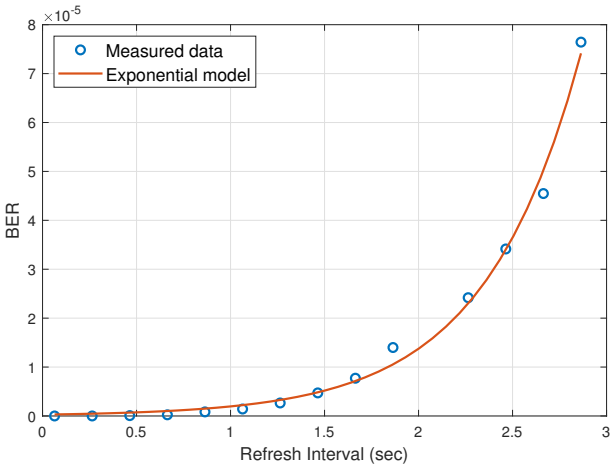


Fig. 4. The measured BERs at 80 °C [21, Table I] and the exponential model with the estimated $\alpha=2.7737\times10^{-7}$ and $\beta=1.9508$.

Suppose that $t_b=\Delta\cdot z_b$ where $\Delta=\gamma\delta$ and $z_b\in\mathbb{N}$ ($\mathbb{N}$ denotes the positive integers) for $b\in[0,B-1]$. Note that the step size of the refresh interval Δ is determined by $\gamma\in\mathbb{N}$, which controls the discrete optimization complexity and accuracy. Then, the convex optimization problem (7) can be modified into the following convex integer programming problem:

$$\begin{aligned}\underset{\mathbf{z}}{\text{minimize}}\quad & \mathsf{MSE}(\mathbf{z})=\sum_{b=0}^{B-1}4^b\alpha\exp(\beta\gamma\delta\cdot z_b)\\ \text{subject to}\quad & \mathsf{P}(\mathbf{z})=\frac{1}{\gamma\delta}\sum_{b=0}^{B-1}\frac{1}{z_b}\le\mathcal{P}\\ & z_b\in\mathbb{N},\quad b=0,\ldots,B-1\end{aligned} \quad (11)$$

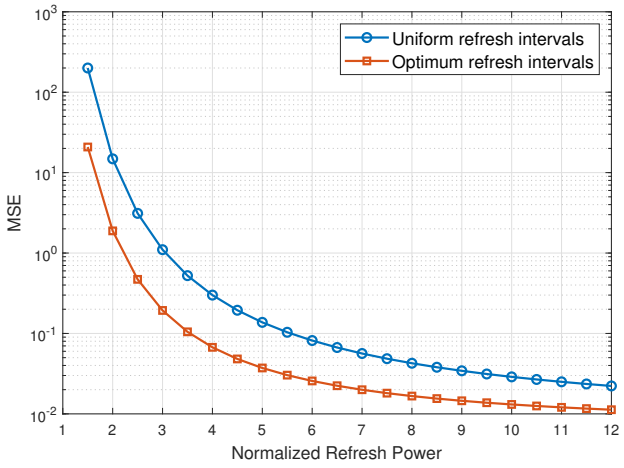
where the positive integer solution $\mathbf{z}^*$ results in the optimized discrete refresh interval by $\widetilde{\mathbf{t}}^*=\Delta\cdot\mathbf{z}^*$.

Although convex integer programming is NP-hard, it can be solved much more efficiently than general integer nonlinear programming problems [24], [25]. We obtained the optimized discrete solutions by standard mixed-integer nonlinear program (MINLP) solvers. The numerical results are provided in Section V.

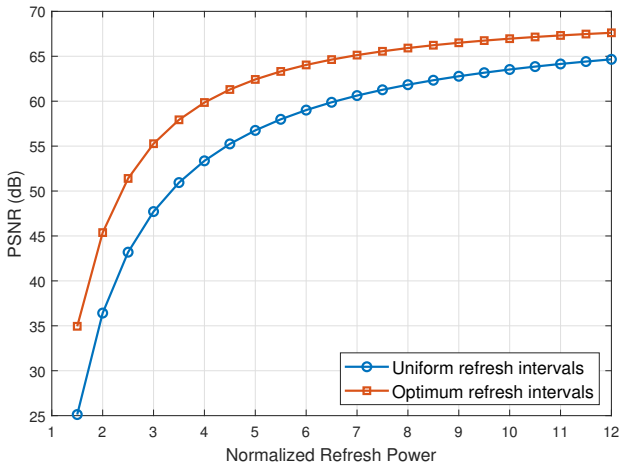
V. Numerical Results

We evaluate the solutions of convex optimization problem (7) and the discrete optimization problem (11). First, we estimate the parameters α and β of (6). From the measured data in [21], we obtained the estimates of $\alpha=2.7737\times10^{-7}$ and $\beta=1.9508$ (see Fig. 4). Note that these parameters depend on manufacturers, products, and temperature as shown in [5, Fig. 4]. We note that higher-capacity, later-generation DRAM devices suffer from more retention failures [5], [18].

Fig. 5 shows numerical results by solving (7). Fig. 5(a) compares the MSEs of uniform refresh intervals and the optimal refresh intervals. At $\mathsf{MSE}=1$, the optimal refresh intervals reduce the refresh power consumption by 27 %. For lower MSE, we can save more refresh power (e.g., 36 % refresh power reduction at $\mathsf{MSE}=10^{-1}$).



(a)



(b)

Fig. 5. Evaluation of proposed convex optimization in (7) (for $B = 8$): (a) MSE and (b) PNSR.

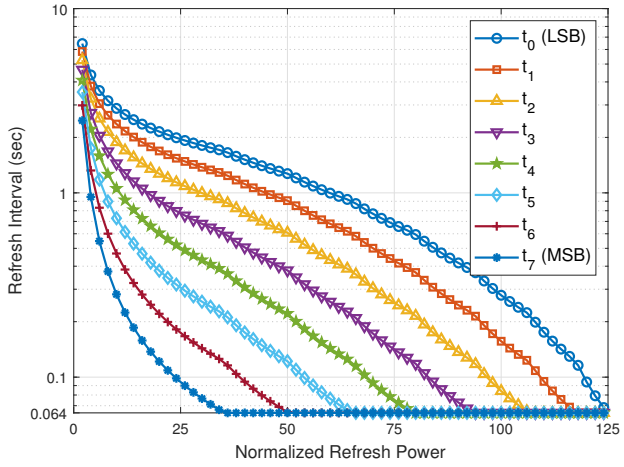


Fig. 6. The optimal refresh interval assignments by Theorem 4.

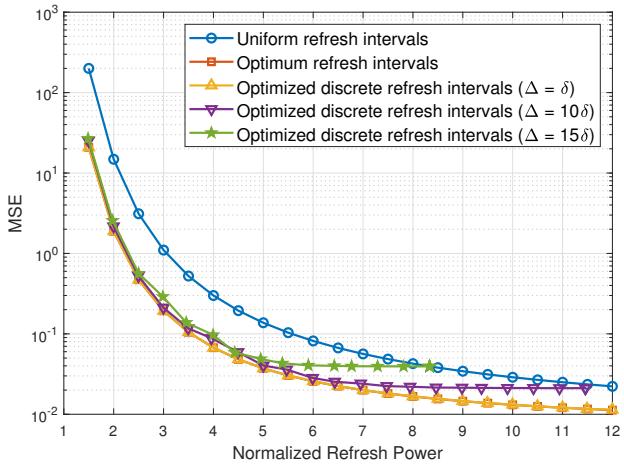


Fig. 7. Evaluation of proposed discrete optimization of (11).

Fig. 5(b) compares the peak signal-to-noise ratios (PSNRs) of refresh interval assignments, which is a widely used fidelity metric for image and video quality. The PSNR depends on the MSE as

$$\mathsf{PSNR} = 10\log_{10}\frac{(2^B-1)^2}{\mathsf{MSE}}. \tag{12}$$

At $\mathsf{PSNR} = 50\,\mathrm{dB}$, the optimized refresh intervals can reduce the refresh power by $29\,\%$. Further, the optimized refresh intervals achieve $38\,\%$ power reduction at $\mathsf{PSNR} = 60\,\mathrm{dB}$. The improvement by the optimized refresh intervals increases for a higher fidelity requirement. If we achieve a target fidelity (e.g., $\mathsf{PSNR} = 50\,\mathrm{dB}$ is a quite reliable value in real-world images [26]), we do not need to waste power by refreshing every $64\,\mathrm{ms}$, which requires $\mathsf{P}_{\mathsf{max}} = 125$ (see Remark 5). Note that the optimized refresh interval assignment achieves $\mathsf{PSNR} = 50\,\mathrm{dB}$ with $\mathsf{P}(\mathbf{t}^*) = 2.4$, which is less than $2\,\%$ of $\mathsf{P}_{\mathsf{max}}$.

Fig. 6 shows the optimal refresh interval assignments by Theorem 4. The shorter refresh intervals (i.e., more refresh power assignments) are allocated to the more significant bits to minimize the MSE. As the refresh power budget $\mathcal{P}$ in (7) increases, the refresh intervals for more significant bits converge to δ. Fig. 6 shows that $t_7 = \delta$ from $\mathcal{P} = 36$. More refresh intervals become δ for higher refresh power budget.

Fig. 7 shows the MSEs obtained by solving convex integer programming problem (11). This convex integer problem was solved by using *Bonmin* [25]. We observe that the MSE penalty due to discrete refresh intervals is negligible for a moderate step size $\Delta = \gamma\delta$. The MSE by discrete refresh intervals with $\Delta = \delta$ is almost the same as the optimal MSE. For $\Delta = 15\delta$, the MSEs are distinct from the optimal MSEs from $\mathsf{P} = 6$. Note that the maximum refresh power with $\Delta = 15\delta$ is $\mathsf{P} = \frac{B}{15\delta} \simeq 8.33$.

VI. Conclusion

We developed a principled approach to optimizing refresh intervals for energy-efficient memories. By formulating the convex optimization problem, we obtained the optimal refresh intervals to minimize the MSE under a refresh power budget. Also, we formulated a discrete optimization problem by taking into account the current DRAM standards and hardware implementation. The numerical results show that the optimum refresh intervals can achieve refresh power reductions of

$29\,\%$ (at $\mathsf{PSNR} = 50\,\mathrm{dB}$) and $38\,\%$ (at $\mathsf{PSNR} = 60\,\mathrm{dB}$), respectively.

Appendix A
Proof of Theorem 4

The KKT conditions of (7) are as follows:

$$\sum_{b=0}^{B-1} \frac{1}{t_b} \leq \mathcal{P}, \quad \nu \geq 0, \quad \nu \cdot \left(\sum_{b=0}^{B-1} \frac{1}{t_b} - \mathcal{P} \right) = 0, \tag{13}$$

$$t_b \geq \delta, \quad \lambda_b \geq 0, \quad \lambda_b \left(t_b - \delta \right) = 0 \tag{14}$$

$$\frac{\partial L_1}{\partial t_b} = 4^b \alpha \beta \exp(\beta t_b) - \frac{\nu}{t_b^2} - \lambda_b = 0 \tag{15}$$

for $b \in [0, B-1]$. From (15), λ_b is given by

$$\lambda_b = 4^b \alpha \beta \exp(\beta t_b) - \frac{\nu}{t_b^2}. \tag{16}$$

From (14) and (16),

$$\lambda_b \left(t_b - \delta \right) = \left(4^b \alpha \beta \exp(\beta t_b) - \frac{\nu}{t_b^2} \right) \left(t_b - \delta \right) = 0. \tag{17}$$

Suppose that $\nu = 0$. Then $\lambda_b = 4^b \alpha \beta \exp(\beta t_b) \neq 0$. Hence, $t_b = \delta$ for any $b \in [0, B-1]$. This is a trivial solution and the corresponding refresh power is $\mathsf{P}((\delta, \ldots, \delta)) = \frac{B}{\delta}$. If this trivial solution does not violate the power budget constraint (i.e., $\frac{B}{\delta} \leq \mathcal{P}$), then it will achieve the minimum MSE. However, we are more interested in the case of $\frac{B}{\delta} > \mathcal{P}$. Hence, we focus on $\nu \neq 0$, which results in $\sum_{b=0}^{B-1} \frac{1}{t_b} = \mathcal{P}$.

If $\lambda_b > 0$, then $t_b = \delta$. By (15), the condition of $\lambda_b > 0$ is equivalent to $\frac{\nu}{4^b} < \alpha \beta t_b^2 \exp\left(\beta t_b\right)$. By (17), we claim that $t_b^* = \delta$ for $\frac{\nu}{4^b} < \alpha \beta \delta^2 \exp\left(\beta \delta\right)$. If $\lambda_b = 0$, then

$$\alpha \beta t_b^2 \exp(\beta t_b) = \frac{\nu}{4^b} \tag{18}$$

which is equivalent to $\frac{\beta t_b}{2} \exp\left(\frac{\beta t_b}{2} \right) = \frac{\beta}{2} \sqrt{\frac{\nu}{4^b \alpha \beta}}$. By setting $x = \frac{\beta t_b}{2}$, we obtain $x \exp\left(x\right) = \frac{\beta}{2} \sqrt{\frac{\nu}{4^b \alpha \beta}}$. Hence, $W\left(\frac{\beta}{2} \sqrt{\frac{\nu}{4^b \alpha \beta}} \right) = x = \frac{\beta t_b}{2}$, i.e., $t_b = \frac{2}{\beta} W\left(\frac{\beta}{2} \sqrt{\frac{\nu}{4^b \alpha \beta}} \right)$.

Acknowledgment

The work of Yuval Cassuto was partly supported by the US-Israel Binational Science Foundation.

References

[1] I. Bhati, M. Chang, Z. Chishti, S. Lu, and B. Jacob, "DRAM refresh mechanisms, penalties, and trade-offs," *IEEE Trans. Comput.*, vol. 65, no. 1, pp. 108–121, Jan. 2016.

[2] P. G. Emma, W. R. Reohr, and M. Meterelliyoz, "Rethinking refresh: Increasing availability and reducing power in DRAM for cache applications," *IEEE Micro*, vol. 28, no. 6, pp. 47–56, Nov. 2008.

[3] T. Ohsawa, K. Kai, and K. Murakami, "Optimizing the DRAM refresh count for merged DRAM/logic LSIs," in *Proc. ACM/IEEE Int. Symp. Low Power Electron. Design (ISLPED)*, Aug. 1998, pp. 82–87.

[4] J. Liu, B. Jaiyen, R. Veras, and O. Mutlu, "RAIDR: Retention-aware intelligent DRAM refresh," in *Proc. ACM/IEEE Annu. Int. Symp. Comput. Archit. (ISCA)*, Jun. 2012, pp. 1–12.

[5] S. Khan, D. Lee, Y. Kim, A. R. Alameldeen, C. Wilkerson, and O. Mutlu, "The efficacy of error mitigation techniques for dram retention failures: A comparative experimental study," *SIGMETRICS Perform. Eval. Rev.*, vol. 42, no. 1, pp. 519–532, Jun. 2014.

[6] S. Liu, K. Pattabiraman, T. Moscibroda, and B. G. Zorn, "Flikker: Saving DRAM refresh-power through critical data partitioning," *SIGARCH Comput. Archit. News*, vol. 39, no. 1, pp. 213–224, Mar. 2011.

[7] 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 *Proc. IEEE/ACM Annu. Int. Symp. Microarchitecture (MICRO)*, Dec. 2007, pp. 134–145.

[8] Y. Katayama, E. J. Stuckey, S. Morioka, and Z. Wu, "Fault-tolerant refresh power reduction of DRAMs for quasi-nonvolatile data retention," in *Proc. IEEE Int. Symp. Defect and Fault Tolerance in VLSI Syst.*, Nov. 1999, pp. 311–318.

[9] C. Wilkerson, A. R. Alameldeen, Z. Chishti, W. Wu, D. Somasekhar, and S.-l. Lu, "Reducing cache power with low-cost, multi-bit error-correcting codes," in *Proc. ACM/IEEE Annu. Int. Symp. Comput. Archit. (ISCA)*, Jun. 2010, pp. 83–93.

[10] C. Chou, P. Nair, and M. K. Qureshi, "Reducing refresh power in mobile devices with morphable ECC," in *Proc. IEEE/IFIP Int. Conf. Dependable Syst. Netw.*, Jun. 2015, pp. 355–366.

[11] K. Cho, Y. Lee, Y. H. Oh, G.-c. Hwang, and J. W. Lee, "eDRAM-based tiered-reliability memory with applications to low-power frame buffers," in *Proc. ACM/IEEE Int. Symp. Low Power Electron. Design (ISLPED)*, Aug. 2014, pp. 333–338.

[12] W. H. Choi, M. Lueker-Boden, M. Grobis, N. Robertson, and Z. Bandic, "A comprehensive study on DDR4 MRAM and ReRAM power estimation using a parameterized NVM power calculator," in *Proc. IEEE Int. Memory Workshop (IMW)*, May 2018, pp. 1–4.

[13] C. W. Smullen, V. Mohan, A. Nigam, S. Gurumurthi, and M. R. Stan, "Relaxing non-volatility for fast and energy-efficient STT-RAM caches," in *Proc. IEEE Int. Symp. High Performance Comput. Architecture (HPCA)*, Feb. 2011, pp. 50–61.

[14] A. Jog, A. K. Mishra, C. Xu, Y. Xie, V. Narayanan, R. Iyer, and C. R. Das, "Cache revive: Architecting volatile STT-RAM caches for enhanced performance in CMPs," in *Proc. Des. Autom. Conf (DAC)*, Jun. 2012, pp. 243–252.

[15] Y. Kim, M. Kang, L. R. Varshney, and N. R. Shanbhag, "Generalized water-filling for source-aware energy-efficient SRAMs," *IEEE Trans. Commun.*, vol. 66, no. 10, pp. 4826–4841, Oct. 2018.

[16] ——, "SRAM bit-line swings optimization using generalized waterfilling," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2018, pp. 1670–1674.

[17] K. K. Chang, A. Kashyap, H. Hassan, S. Ghose, K. Hsieh, D. Lee, T. Li, G. Pekhimenko, S. Khan, and O. Mutlu, "Understanding latency variation in modern DRAM chips: Experimental characterization, analysis, and optimization," in *Proc. ACM SIGMETRICS Int. Conf. Meas. Model. Comput. Syst.*, Jun. 2016, pp. 323–336.

[18] J. Liu, B. Jaiyen, Y. Kim, C. Wilkerson, and O. Mutlu, "An experimental study of data retention behavior in modern DRAM devices: implications for retention time profiling mechanisms," in *Proc. ACM/IEEE Annu. Int. Symp. Comput. Archit. (ISCA)*, Jun. 2013, pp. 60–71.

[19] J. Kim and M. C. Papaefthymiou, "Block-based multiperiod dynamic memory design for low data-retention power," *IEEE Trans. VLSI Syst.*, vol. 11, no. 6, pp. 1006–1018, Dec. 2003.

[20] X. Yang and K. Mohanram, "Unequal-error-protection codes in SRAMs for mobile multimedia applications," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design (ICCAD)*, Nov. 2011, pp. 21–27.

[21] C. Hou, J. Li, C. Lo, D. Kwai, Y. Chou, and C. Wu, "An FPGA-based test platform for analyzing data retention time distribution of DRAMs," in *Proc. Int. Symp. VLSI Design, Autom., and Test*, Apr. 2013, pp. 1–4.

[22] D. P. Palomar and J. R. Fonollosa, "Practical algorithms for a family of waterfilling solutions," *IEEE Trans. Signal Process.*, vol. 53, no. 2, pp. 686–695, Feb. 2005.

[23] R. M. Corless, G. H. Gonnet, D. E. G. Hare, D. J. Jeffrey, and D. E. Knuth, "On the Lambert W function," *Adv. Comput. Math.*, vol. 5, no. 1, pp. 329–359, Dec. 1996.

[24] P. Bonami, L. T. Biegler, A. R. Conn, G. Cornuéjols, I. E. Grossmann, C. D. Laird, J. Lee, A. Lodi, F. Margot, N. Sawaya, and A. Wächter, "An algorithmic framework for convex mixed integer nonlinear programs," *Discrete Optimization*, vol. 5, no. 2, pp. 186–204, May 2008.

[25] P. Bonami, M. Kilinç, and J. Linderoth, "Algorithms and software for convex mixed integer nonlinear programs," in *Proc. Mixed Integer Nonlinear Programming*, Nov. 2012, pp. 1–39.

[26] A. Amer and E. Dubois, "Fast and reliable structure-oriented video noise estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 1, pp. 113–118, Jan. 2005.