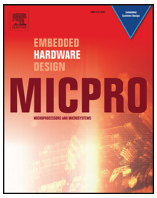
Microprocessors and Microsystems 42 (2016) 100–112

Contents lists available at ScienceDirect

Microprocessors and Microsystems

journal homepage: www.elsevier.com/locate/micpro



Exploration of temperature-aware refresh schemes for 3D stacked eDRAM caches

Young-Ho Gong[a], Jae Min Kim[b], Sung Kyu Lim[c], Sung Woo Chung[a,*]

[a] Department of Computer Science, Korea University, Seoul 136-713, Korea
[b] Samsung Electronics DS, San#24 Nongseo-Dong, Giheung, Gyeonggi-do 446-711, Korea
[c] School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332-0250, USA

ARTICLE INFO

Article history:
Received 26 August 2014
Revised 20 May 2015
Accepted 2 January 2016
Available online 8 February 2016

Keywords:
Cache
eDRAM
Refresh interval
3D Microprocessors
Temperature

ABSTRACT

Recent studies have shown that embedded DRAM (eDRAM) is a promising approach for 3D stacked last-level caches (LLCs) rather than SRAM due to its advantages over SRAM; (i) eDRAM occupies less area than SRAM due to its smaller bit cell size; and (ii) eDRAM has much less leakage power and access energy than SRAM, since it has much smaller number of transistors than SRAM. However, different from SRAM cells, eDRAM cells should be refreshed periodically in order to retain the data. Since refresh operations consume noticeable amount of energy, it is important to adopt appropriate refresh interval, which is highly dependent on the temperature. However, the conventional refresh method assumes the worst-case temperature for all eDRAM stacked cache banks, resulting in unnecessarily frequent refresh operations. In this paper, we propose a novel temperature-aware refresh scheme for 3D stacked eDRAM caches. Our proposed scheme dynamically changes refresh interval depending on the temperature of eDRAM stacked last-level cache (LLC). Compared to the conventional refresh method, our proposed scheme reduces the number of refresh operations of the eDRAM stacked LLC by 28.5% (on 32 MB eDRAM LLC), on average, with small area overhead. Consequently, our proposed scheme reduces the overall eDRAM LLC energy consumption by 12.5% (on 32 MB eDRAM LLC), on average.

1. Introduction

As the process technology scales down, microprocessor performance improves dramatically, especially due to the reduced gate delay. However, the wire delay is not much reduced compared to the gate delay, since it is more affected by wire length. Among components in a microprocessor, caches occupy the largest area since microprocessors have multi-megabyte last-level cache (LLC). The large cache area leads to long wire delay that accounts for a substantial portion of cache access time [6]. To alleviate the long wire delay, many studies have proposed 3D microprocessor design which uses vertical interconnection by through-silicon-vias (TSVs) between dies [23–25].

There are two representative 3D microprocessor design alternatives [23]: (i) cache-on-core and (ii) core-on-core. In the case of cache-on-core 3D microprocessor design, LLC dies are vertically stacked on top of the CPU core die. Since TSV-based 3D interconnection provides shorter wire length than 2D wire interconnection, it reduces the cache access time of cache-on-core 3D microprocessors significantly, compared to conventional 2D microprocessors. In the case of core-on-core 3D microprocessor design, each die consists of functional blocks of a CPU core as well as a fragment of LLC. In each die, the functional blocks and the fragment of LLC are connected by 2D wire interconnection. In order to improve performance, functional blocks (such as arithmetic units, register file, and etc.) are split into each die. Furthermore, the functional blocks far from each other in the 2D microprocessor design are connected vertically to improve performance [17]. However, the performance improvement of core-on-core 3D microprocessors is insignificant compared to that of cache-on-core 3D microprocessors, since the wire delay reduction between functional blocks is not so large compared to that in the case of cache access [16]. In addition, core-on-core 3D microprocessor design is more likely to face thermal problem than cache-on-core 3D microprocessor design, since vertically stacked functional blocks are much hotter than vertically stacked caches [18,20]. Note that the increased power density of 3D microprocessors causes higher on-chip temperature [28]. Thus, many recent studies have focused on cache-on-core 3D microprocessor design [1,5,24,25].

In conventional 2D microprocessors, most caches consist of SRAM cells to provide high performance. However, the high

[*] Corresponding Author. Tel.: +82 2 3290 3571.
E-mail addresses: kyh555@korea.ac.kr (Y.-H. Gong), joist@gmail.com (J.M. Kim), limsk@ece.gatech.edu (S.K. Lim), swchung@korea.ac.kr (S.W. Chung).

http://dx.doi.org/10.1016/j.micpro.2016.01.010
0141-9331/© 2016 Elsevier B.V. All rights reserved.

performance comes at the expense of large area and high leakage power. Especially, among components in the conventional 2D microprocessor, SRAM based LLC has the largest area and the highest leakage power due to its large capacity. To alleviate large area and high leakage power of SRAM based LLC, many studies have proposed embedded DRAM (eDRAM) based LLC [5,13,34]. Compared to SRAM cells, eDRAM cells have smaller area and less leakage power, since they consist of less number of transistors than SRAM cells. Hence, recent studies have shown that using eDRAM stacked LLC is a promising approach for 3D microprocessors rather than using SRAM stacked LLC [5,34].

Different from the SRAM cells, eDRAM cells need to be refreshed periodically to preserve their data. The eDRAM cell *retention time* (the maximum time the stored data can be retained without any refresh operation) varies depending on temperature [32]. As eDRAM cell temperature gets higher, its leakage current also increases. Eventually, the eDRAM cell needs more frequent refresh operations to preserve its data.

With the conventional refresh method, all eDRAM stacked cache banks are refreshed once every predefined constant *refresh interval* (the period between the beginnings of two successive refresh operations), which ensures the data integrity even at the highest operating temperature. In fact, the eDRAM stacked cache banks under lower temperature can retain their data with longer refresh interval than the predefined constant refresh interval. In other words, the conventional refresh method causes a lot of unnecessary refresh operations for the eDRAM stacked cache banks under low temperature. Such unnecessary refresh operations eventually result in energy/performance loss.

In practice, recent low-power DRAM chips (which are used for main memory systems) adopt a temperature-aware refresh method, which is called Temperature Compensate Self Refresh (TCSR) [38]. The TCSR changes the refresh interval of the DRAM chip depending on temperature of the whole DRAM chip. However, the TCSR cannot reduce refresh overhead significantly, since the TCSR does not have fine-grained temperature-aware refresh intervals; it has only a few (two or four) refresh intervals depending on temperature of whole DRAM chip. Compared to DRAM chips (for main memory systems), eDRAM stacked caches must have higher peak temperature in runtime due to the impact of 3D stacking on heat radiation ability. According to the temperature analysis of a recent computer system [36], the peak temperature of the DRAM chip is below 60 °C. However, the peak temperature of the CPU core is nearby 80 °C. Assuming that the eDRAM stacked caches are applied to the system, they have much higher peak temperature than DRAM chips. In addition, eDRAM stacked caches have a considerable spatial temperature variation across eDRAM cache banks, since each eDRAM cache bank has different heat radiation ability depending on distance from the heat spreader. In order to effectively reduce refresh overhead of eDRAM stacked caches, we should adopt more fine-grained temperature-aware refresh intervals. Moreover, to further reduce refresh overhead, we should vary refresh intervals of eDRAM cache banks depending on temperature of each cache bank. However, since the TCSR does not use fine-grained temperature-aware refresh intervals and does not apply different refresh intervals to different banks, it is not appropriate for reducing refresh overhead of eDRAM stacked caches.

In this paper, we analyze the retention time of eDRAM stacked cache banks depending on temperature. Since temperature of each eDRAM stacked cache bank varies, the retention time can be different for different cache banks. Note that an eDRAM cell under lower temperature retains its data much longer without any refresh operation. Considering the different retention time of eDRAM stacked cache banks due to temperature changes, we propose a novel temperature-aware refresh scheme for 3D stacked eDRAM caches. We use thermal sensors for detecting temperature of each cache bank. Depending on temperature from thermal sensors, the scheme proposed in this paper applies temperature-aware refresh interval to eDRAM LLC. In addition, our scheme dynamically changes the refresh interval, depending on run-time temperature. As a result, our scheme significantly reduces the number of refresh operations.

The rest of this paper is organized as follows. In Section 2, we present motivational study why the temperature should be considered in the 3D stacked eDRAM caches. In Section 3, we present a review of related works. In Section 4, we propose a novel temperature-aware refresh scheme. In Section 5, we provide our evaluation methodology and evaluation results, in the perspective of temperature, refresh interval, number refresh operations, and energy consumption. Lastly, we conclude our paper and discuss future work in Section 6.

2. Motivation

The thermal problem is more serious in 3D microprocessors than in 2D microprocessors due to the following reasons:

(1) Temperature of each die is likely to increase along with the distance from heat spreader, since the die far from the heat spreader has low heat radiation ability. For example, in Fig. 1, the die in layer 4 has lower heat radiation ability than that in layer 1.
(2) Temperature of each die is affected by temperature of adjacent dies. For example, in Fig. 1, when the die in layer 2 becomes hot, the adjacent dies (layer 1 and layer 3) are also heated up.

Fig. 2(a)–(c) show thermal maps of 8 MB (1-die), 16 MB (2-die stacked), and 32 MB (4-die stacked) eDRAM stacked LLC, respectively; note that the L2 cache is used for the LLC in our paper. Each cache die is 8 MB eDRAM cache composed of eight 1 MB eDRAM cache banks. As shown in Fig. 2(c), the cache die on top of the stack is hottest due to lower heat radiation ability.

The cache banks shown in Fig. 2(a)–(c) have different retention times due to the temperature difference. To analyze how temperature affects eDRAM retention time, we investigate the relation between temperature and retention time. According to [11], retention time of an eDRAM cell is written as follows:

$$T_{RETENTION} \propto \frac{Cs\Delta V_{SN}}{I_{LEAK}} \quad (1)$$

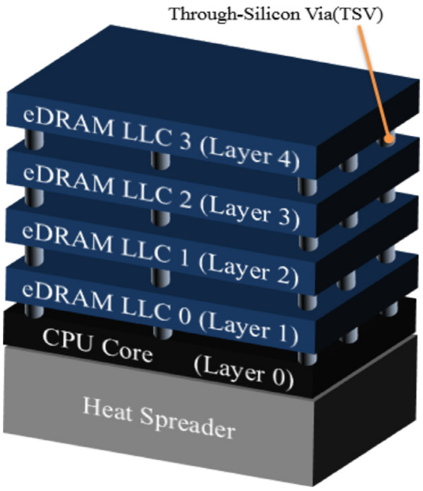


Fig. 1. 3D microprocessor design with eDRAM stacked caches.(Note that this represents a cache-on-core 3D microprocessor design).

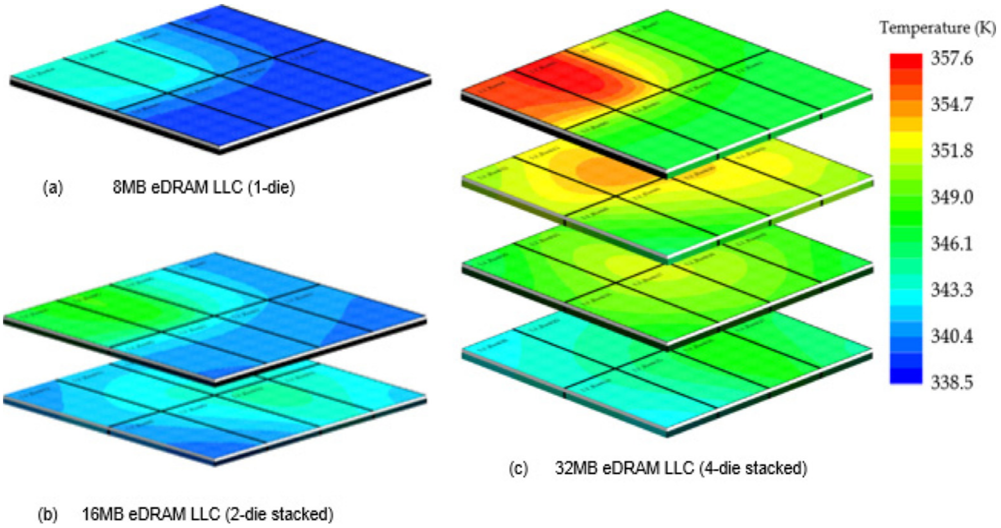


Fig. 2. Thermal maps of 3D stacked eDRAM caches. Note that these thermal maps are the results from the evaluation of Group 1. Detailed parameters for HotSpot thermal simulation are described in Table I. The list of applications (Group 1) used in the evaluation is described in Table III.

where C_S, V_{SN}, and I_{LEAK} are the capacitance of the eDRAM cell, the voltage of the eDRAM cell and the leakage current of the eDRAM cell, respectively.

As shown in Eq. (1), retention time of an eDRAM cell is inversely proportional to its leakage current. The leakage current of the eDRAM cell is written as follows [35]:

$$\boldsymbol{I_{LEAK}} \propto \boldsymbol{\mu_0} \cdot \boldsymbol{W} \cdot \boldsymbol{v_t^2} \tag{2}$$

where $\mu 0$, W, and $\upsilon t = kT/q$ are the constant for a given device, the device width, and the thermal voltage, respectively (k, T, and q are the Boltzmann constant in Joules/K, temperature, and the electrical charge on the electron, respectively.).

The thermal voltage in Eq. (2) is the average electron-volts (eV, a unit of energy) at a given temperature. As temperature increases, the thermal voltage increases. As shown in Eq. (2), the leakage current of the eDRAM cell is quadratically proportional to the thermal voltage. Since the thermal voltage linearly increases along with cell temperature, the leakage current of the eDRAM cell increases quadratically with cell temperature. To summarize, the retention time of the eDRAM cell rapidly decreases as the cell temperature increases. Therefore, different refresh interval should be used depending on the temperature, in order to reduce the number of refresh operations. Nevertheless, the conventional refresh method applies same refresh interval to all cache banks assuming that temperature of all cache banks is always high [38]. According to [3,33], the conventional eDRAM cache adopts 100 μs refresh interval for all cache banks assuming that temperature of all cache banks is always 368 K (95 °C). However, as shown in Fig. 2(a)–(c), even the hottest cache bank is 357.6 K, which is 10.4 K lower than 368 K. Thus, the cache banks can retain their data, though they are refreshed with the refresh interval longer than 100 μs. Consequently, the number of refresh operations can be reduced. Furthermore, when we apply different refresh intervals to different eDRAM stacked cache banks depending on temperature, the number of refresh operations can be further reduced compared to using only one refresh interval for all cache banks.

To extract quantitative relation between eDRAM retention time and temperature, we use CACTI [32] based on the fact that eDRAM cache has 100 μs retention time at 368 K (95 °C) [3,33]. We estimate the leakage current of 1 MB eDRAM cache bank at various temperatures and calculate the retention time of 1 MB eDRAM cache bank based on (1). Fig. 3 shows retention time of the 1 MB eDRAM cache bank depending on the temperature.

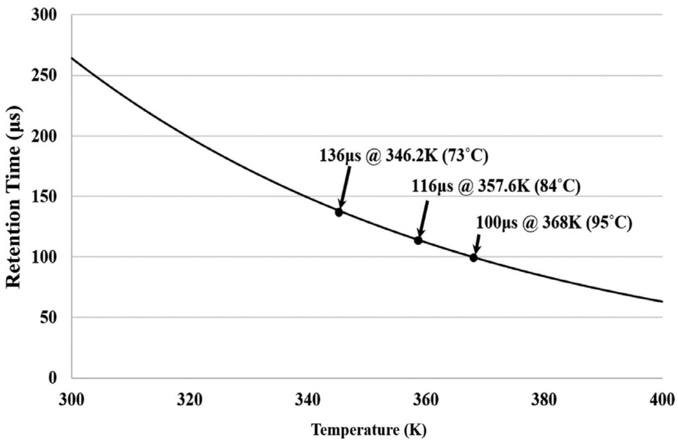


Fig. 3. Retention time of the 1 MB eDRAM cache bank.

As shown in Fig. 3, retention time dramatically decreases as temperature increases. Based on Fig. 3, we further analyze the retention time difference across eDRAM stacked cache banks shown in Fig. 2(c). In the case of the 32 MB eDRAM stacked LLC (Fig. 2(c)), temperature of the coolest cache bank and temperature of the hottest cache bank are 346.2 K and 357.6 K, respectively. According to Fig. 3, retention time of the coolest cache bank is 136 μs which is 36% longer than 100 μs (the conventional eDRAM refresh interval). Even in the hottest cache bank, the retention time (116 μs) is 16% longer than 100 μs. Since the retention times shown in Fig. 3 are based on the worst-case refresh interval of the conventional eDRAM (100 μs at 95 °C), all eDRAM cells in our eDRAM model have retention times longer than the retention time shown in Fig. 3 at any temperature. In this case, longer refresh interval can be used to reduce refresh overhead, while still retaining the cell data. Hence, the scheme proposed in this paper exploits the opportunity of energy savings by adopting temperature-aware refresh intervals.

3. Related work

There have been many previous studies on low-power refresh for DRAM chips used for main memory systems. Since eDRAM, which is the target memory architecture in this paper, is similar to

DRAM, we review previous studies on low-power refresh for DRAM chips and briefly compare them with the scheme proposed in this paper.

3.1. Refresh schemes considering temperature

In this section, we review refresh schemes considering temperature-dependent retention time. Though the refresh schemes presented in this section seem to be similar to the scheme proposed in this paper, there are differences between our scheme and the refresh schemes.

Recent low-power DRAM chips adopt Temperature Compensate Self Refresh (TCSR) scheme [38]. The TCSR applies different refresh interval to the DRAM chip depending on operating temperature range. The operating temperature range is divided into several sections. For example, in the Micron low-power DRAM, the operating temperature is divided into two sections [38]: (i) normal temperature range (from 0 °C to 85 °C) and (ii) high temperature range (from 85 °C to 105 °C). At normal temperature range, the DRAM chip is refreshed with nominal refresh interval (e.g., 64 ms). When the DRAM chip is at high temperature range, the DRAM chip is refreshed with reduced refresh interval (e.g., half of the nominal refresh interval). However, the TCSR cannot reduce refresh overhead significantly, since it does not have fine-grained temperature-aware refresh intervals. Compared to the TCSR, the scheme proposed in this paper divides the operating temperature more fine-grained. Our scheme adopts temperature-aware refresh interval for all eDRAM cache banks at 1 °C temperature granularity. Hence, our scheme further reduces refresh overhead compared to the TCSR, which will be described in Section 5.5.

Sadri et al. [29] proposed Bank-wise Refresh for wide-I/O DRAM chips. Their scheme applies different refresh intervals to different DRAM banks depending on temperature. Though their design philosophy is similar to the scheme proposed in this paper, there are differences as follows.

(1) Our scheme is designed for 3D stacked eDRAM-based LLC. On the other hand, Bank-wise Refresh is designed for wide-I/O DRAM-based main memory system. There are several important differences between eDRAM-based LLC and wide-I/O DRAM-based main memory. Firstly, while the unit of LLC access is a cache line of several bytes, the unit of main memory access is a page of several kilobytes. Since eDRAM-based LLC and wide-I/O DRAM-based main memory are significantly different in the unit of access, they are entirely different in access behavior. Secondly, eDRAM-based LLC is much hotter than wide-I/O DRAM-based main memory since it is much closer to CPU cores which are the hottest components in the computer system. Thirdly, eDRAM-based LLC should have much shorter refresh interval than wide-I/O DRAM-based main memory, since it has much shorter retention time than wide-I/O DRAM-based main memory at the same temperature. Due to these differences, Bank-wise Refresh is not scalable to eDRAM-based LLC.

(2) Even if we assume that Bank-wise Refresh is used for eDRAM-based LLC, our scheme is more robust to rapid temperature change than their scheme. Since recent microprocessors adjust Dynamic Voltage and Frequency Scaling (DVFS) of each CPU core at least every 10 ms, temperature of the CPU core die can be rapidly changed on the order of millisecond [14,19]. In order to adjust refresh intervals depending on temperature change, the temperature sensing interval should be much lower than 10 ms. Thus, our scheme monitors temperature every 30 µs. When temperature of a cache bank increases 1 °C for 30 µs, our scheme refreshes the cache bank immediately and updates the refresh interval of the cache bank. In addition, considering temperature sensor error, our scheme adds 3 °C temperature margin to the measured temperature value, when it determines temperature-aware refresh interval. Thus, our scheme prevents data loss even in the case of rapid temperature change; the detailed algorithm of our scheme will be described in Section 4.3. Contrary to our scheme, Bank-wise Refresh adjusts much longer temperature sensing intervals depending temperature (from 32 ms at 40 °C to 1 ms at 100 °C). Furthermore, their scheme does not consider temperature sensor error. Therefore, in the case of rapid temperature change, their scheme may lose data.

(3) Our scheme changes refresh intervals more sensitive to temperature than their scheme. Our scheme adopts different refresh intervals at 1 °C temperature granularity. Though our scheme uses fine-grained temperature-aware refresh intervals, it guarantees data integrity, since it adopts sufficient temperature margin and temperature sensing interval which is short enough to detect rapid temperature change. On the contrary, their scheme has only thirteen refresh intervals depending on temperature from 35 °C to 105 °C. Since our scheme applies more optimal refresh intervals to cache banks, it reduces more refresh overhead compared to their scheme, which will be described in Section 5.5.

3.2. Other refresh schemes

In this section, we review refresh schemes which exploit the other characteristics of DRAM (excluding temperature-dependent retention time).

Ghosh and Lee [4] proposed Smart Refresh for DRAM chips. Their scheme is based on the characteristic of the DRAM access. Whenever a DRAM row is accessed, the DRAM row does not need to be refreshed until another refresh interval. In other words, a DRAM read operation has the same effect on the DRAM row as a refresh operation. To utilize the refresh effect of DRAM access, their scheme uses refresh counters for all DRAM rows. Whenever a DRAM row is accessed, the refresh counter for the DRAM row is reset, since the corresponding DRAM row is refreshed by the access. However, their scheme causes large area overhead, because it needs refresh counters as much as the number of DRAM rows. Furthermore, their scheme does not consider the impact of temperature on retention time. Since retention time varies depending on temperature, the number of refresh operations can be significantly reduced with the consideration of the impact of temperature on retention time. Contrary to their scheme, the scheme proposed in this paper adopts different refresh intervals depending on temporal and spatial temperature variation. Therefore, our scheme effectively reduces unnecessary refresh operations in eDRAM stacked caches. In addition, our scheme does not need to have refresh counters for all eDRAM rows since the spatial temperature variation across eDRAM rows is negligible. Hence, our scheme has much smaller area overhead compared to their scheme.

Stuecheli et al. [31] proposed Elastic Refresh for JEDEC DRAM chips. Their scheme mitigates performance overhead due to refresh by delaying refresh operations; the JEDEC DRAM chips can postpone or issue in advance up to eight refresh commands [37]. In their scheme, when the memory systems have heavy read/write traffic, refresh operations are delayed until either the memory rank is idle or the retention time is imminent. Generally, since DRAM chips use much longer nominal refresh interval (64 ms) than eDRAM stacked caches (100 µs), it is possible for DRAM chips to process many read/write operations between refresh operations by delaying refresh operations. However, since eDRAM stacked caches should be refreshed with much shorter refresh interval than DRAM chips, eDRAM stacked caches can have only a few read/write operations between refresh operations by delaying refresh operations. Thus, the performance overhead reduction of their scheme must be insignificant in eDRAM stacked caches. On the other hand, since the scheme proposed in this paper reduces considerable refresh operations in eDRAM stacked caches by exploiting temperature-aware refresh intervals, it can effectively reduce performance overhead due to refresh.

Liu et al. [22] proposed RAIDR for DRAM technology. Their scheme exploits the impact of process variation on DRAM cell to reduce refresh operations. Generally, DRAM uses 64 ms for the refresh interval [7,21]. However, due to the process variation, different DRAM cells have different retention times at the same temperature [22]. Their scheme classifies all DRAM rows into several refresh interval sections based on the retention time of DRAM rows. For example, DRAM rows with retention time from 128 ms to 192 ms are refreshed with 128 ms refresh interval. However, the data stored in the cell can be threatened when temperature increases during execution, since their scheme does not consider temperature. Different from their scheme, the scheme proposed in this paper adopts the refresh interval considering temperature. Our scheme changes refresh intervals adequately depending on run-time temperature changes, by monitoring temperature changes of each cache bank. Thus, our scheme guarantees data integrity though temperature increases rapidly.

Nair et al. [26] proposed Refresh Pausing for JEDEC DRAM chips. Similar to Elastic Refresh [31] which is based on JEDEC DRAM chips, their scheme reduces latency overhead due to refresh by pausing refresh operations. Obviously, in order to avoid data loss, pausing refresh operations should be performed with the consideration of the retention time. As we mentioned before, since eDRAM has much shorter retention time than DRAM, it is impractical to pause refresh operations for a long time in eDRAM. In addition, eDRAM stacked caches have much more read/write operations than DRAM main memory systems. In that case, refresh operations cannot be paused indefinitely, as they also mentioned in their paper [26]. Due to these reasons, Refresh Pausing is not appropriate for reducing latency overhead due to refresh in eDRAM stacked caches. On the contrary, since the scheme proposed in this paper can reduce unnecessary refresh operations by adopting temperature-aware refresh intervals, it can reduce latency overhead due to refresh in the eDRAM stacked caches.

4. Temperature-aware refresh scheme

4.1. Temperature-aware refresh methods

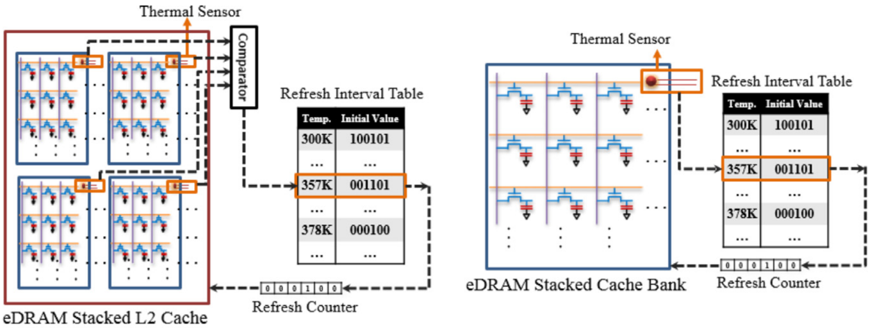
We propose two dynamic methods as follows: (i) peak temperature-aware refresh method (PT_Refresh) and (ii) temperature variation-aware refresh method (TV_Refresh).

Since the conventional refresh method takes the conservative approach by assuming the worst-case temperature, the predefined constant refresh interval is very short (100 μs at 95 °C) [3,33]. On the other hand, PT_Refresh dynamically detects the actual peak temperature among the cache banks by using thermal sensors. When the peak temperature during execution is lower than 368 K (95 °C), PT_Refresh uses the temperature-aware refresh interval, which is longer than 100 μs. As the peak temperature changes during the runtime, PT_Refresh changes the temperature-aware refresh interval dynamically, depending on temperature changes. Hence, PT_Refresh reduces a lot of unnecessary refresh operations compared to the conventional refresh method.

Furthermore, when the spatial temperature variation across eDRAM stacked cache banks becomes larger, the number of refresh operations can be further reduced, by using TV_Refresh. Note that PT_Refresh uses just one refresh interval for all the cache banks in the whole stacked cache dies. Different from PT_Refresh, TV_Refresh adopts different refresh intervals for different eDRAM stacked cache banks, depending on temperature of each cache bank. Similar to PT_Refresh, TV_Refresh also changes the temperature-aware refresh intervals dynamically, depending on temperature changes. Consequently, TV_Refresh further reduces the number of refresh operations compared to PT_Refresh, when there is high spatial temperature variation.

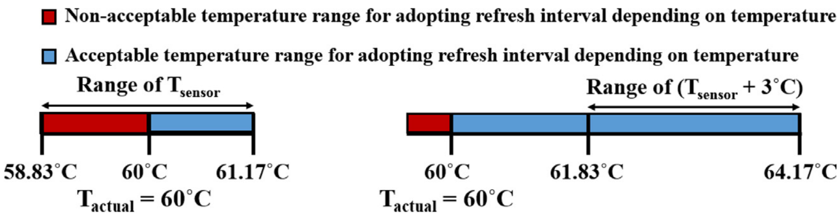
4.2. Hardware structure

Fig. 4 shows the hardware structure for our proposed scheme. As shown in Fig. 4(a) and (b), both PT_Refresh and TV_Refresh require a thermal sensor for each eDRAM stacked cache bank. However, PT_Refresh only needs one refresh counter and one refresh interval table for a whole eDRAM LLC, since all eDRAM stacked cache banks have same refresh interval which corresponds to the refresh interval for the hottest cache bank among all eDRAM stacked cache banks. On the other hand, TV_Refresh requires a refresh counter and a refresh interval table for each eDRAM stacked bank, since each eDRAM stacked cache bank has different refresh intervals depending on its own temperature. Though recent studies on adaptive refresh scheme (which considers process variation-dependent retention time without reflecting temperature-dependent retention time) adopt different refresh intervals at a row granularity [4,22], our proposed scheme adopts temperature-aware refresh interval at whole cache granularity (PT_Refresh) and bank granularity (TV_Refresh), since it is impractical to place a thermal sensor (in both schemes) and a refresh counter (in the case of TV_Refresh) for each eDRAM row due to area overhead. The refresh counter value is decremented by 1, every 10 μs. Therefore, in order to cover the whole range of eDRAM refresh interval (0 μs–270 μs, based on Fig. 3), the refresh counter is a 5-bit timeout counter ($270\,\mu s/10\,\mu s = 27 < 2^5$). In PT_Refresh, whenever the



(a) Peak Temperature-aware Refresh (PT_Refresh)
(b) Temperature Variation-aware Refresh (TV_Refresh)

Fig. 4. Hardware structure of our proposed scheme.


🟥 Non-acceptable temperature range for adopting refresh interval depending on temperature
🟦 Acceptable temperature range for adopting refresh interval depending on temperature
Range of T_{sensor}
58.83°C
60°C
61.17°C
T_{actual} = 60°C
Range of (T_{sensor} + 3°C)
60°C
61.83°C
64.17°C
T_{actual} = 60°C

(a) Without temperature margin (b) With temperature margin (+3°C)

Fig. 5. Temperature measurement range.

refresh counter value reaches zero, all eDRAM stacked cache banks are refreshed and the refresh counter is initialized. On the other hand, in TV_Refresh, each eDRAM stacked cache bank is refreshed individually, when the corresponding refresh counter value reaches zero. At the same time, the corresponding refresh counter is initialized. The initial value of the refresh counter should be changed depending on temperature, since it is determined by the refresh interval. To change the initial value of the refresh counter depending on temperature, the refresh interval table contains 5-bit initial values corresponding to temperatures from 300 K to 400 K at 1 °C temperature granularity. The values in the refresh interval table are based on the worst-case refresh interval of the conventional eDRAM (100 μs at 95 °C). Thus, our proposed scheme ensures that temperature-aware refresh interval at certain temperature T is shorter than the shortest retention time at the temperature *T*. In other words, since our proposed scheme considers the shortest retention time, we are sure that the impact of process variation on eDRAM cells is considered. In the following section, we describe the algorithm to determine temperature-aware refresh interval for our proposed scheme in detail.

4.3. Algorithm overview

In our proposed scheme, we determine refresh intervals depending on temperature measured by thermal sensors. Therefore, it is most important to accurately measure temperature by using thermal sensors. However, since all thermal sensors have measurement error, we should consider the measurement error when we determine refresh interval depending on temperature measured by the thermal sensor. We assume that our proposed scheme considers the thermal sensor proposed by Ituero et al [9]. The thermal sensor has an error range of ± 1.17 °C [9].

Fig. 5(a) shows the temperature measurement range considering the measurement error (±1.17 °C) when the actual temperature (T_{actual}) is 60 °C. As shown in Fig. 5(a), when T_{actual} is 60 °C, the temperature measured by the thermal sensor (T_{sensor}) is between 58.83 °C (T_{actual} −1.17 °C) and 61.17 °C (T_{actual} +1.17 °C). When T_{sensor} is higher than or equal to T_{actual}, the temperature-aware refresh interval (determined by our proposed scheme) is shorter than or equal to the retention time. In this case, though our proposed scheme incurs a little more refresh operations compared to the optimal case (using optimal refresh interval), it guarantees data integrity. On the other hand, when T_{sensor} is lower than T_{actual}, the eDRAM cells lose their data since the temperature-aware refresh interval is longer than the retention time. Therefore, our scheme should add at least 1.17 °C temperature margin to the measured temperature, to retain the data regardless of the measurement error. In our proposed scheme, as shown in Fig. 5(b), we conservatively add 3 °C temperature margin to the measured temperature (2 °C margin for the measurement error and 1 °C additional safe margin), when we determine temperature-aware refresh interval. Therefore, we ensure that the temperature-aware refresh interval in our scheme is always shorter than the retention time in any case.

Furthermore, our scheme also considers the rapid change in temperature, between refresh intervals. Note that the eDRAM cells may lose data when the temperature is rapidly increased, just after a refresh operation. In order to avoid such situation, our proposed scheme monitors the temperatures of all the cache banks every 30 μs. When our proposed scheme detects that T_{sensor} increases more than 1 °C for 30 μs, it performs a refresh operation immediately, and updates the temperature-aware refresh interval depending on the increased temperature. According to [3], the worst-case refresh interval of the conventional eDRAM is 100 μs at 95 °C. In other words, eDRAM data is retained for at least 100 μs, in normal operating temperature (<95 °C). According to the retention time distribution depicted in Fig. 3, in the case of 400 K (127 °C), the retention time (63 μs) is much longer than 30 μs. Therefore, monitoring the temperature change every 30 μs is enough to prevent data loss even in the case of rapid temperature change. Consequently, since our proposed scheme considers the error of thermal sensors and the rapid change in temperature, it definitely ensures that the refresh interval determined by our proposed scheme is shorter than the retention time.

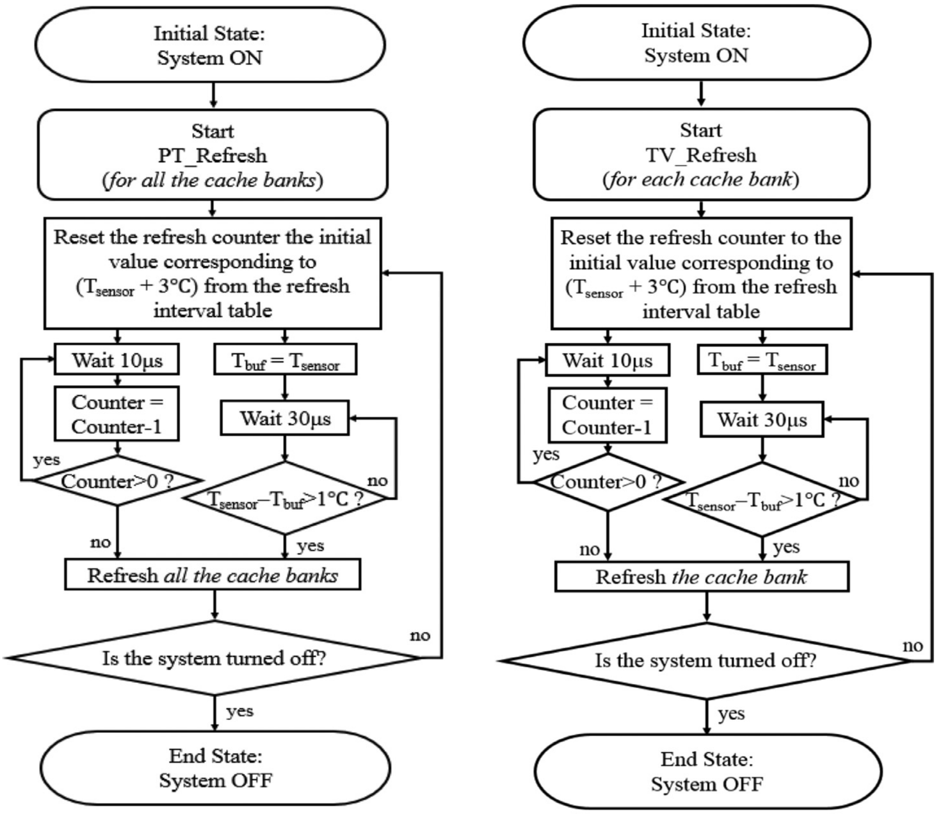
Fig. 6 shows the algorithm for our proposed scheme. As shown in Fig. 6(a), in PT_Refresh, the refresh counter is decremented by 1, every 10 μs. Whenever the refresh counter reaches zero, all the cache banks are refreshed immediately. As we mentioned before, while the refresh counter is being decremented, it is possible that the temperatures of the cache banks increase rapidly. Thus, PT_Refresh monitors the temperatures of all the cache banks, every 30 μs. Among the temperatures obtained from the thermal sensors, the maximum temperature is sent to the refresh interval table. When the maximum temperature increases more than 1 °C for 30 μs, all the cache banks are refreshed immediately. After that, the initial value of refresh counter is selected, considering the new temperature value.

In TV_Refresh, as shown in Fig. 6(b), similar algorithm is applied to each eDRAM cache bank. Whenever the refresh counter reaches zero, the corresponding cache bank is refreshed immediately. At the same time, TV_Refresh monitors the temperature measured by the thermal sensor, every 30°μs. When the temperature increases more than 1 °C for 30 μs, the corresponding cache bank is refreshed immediately. After that, the initial value of the refresh counter is selected, considering the new temperature value.

5. Evaluation

5.1. Evaluation methodology

We apply our proposed scheme to the 8 MB (1-die), 16 MB (2-die stacked), and 32 MB (4-die stacked) eDRAM LLC. An eDRAM cache die consists of eight 1 MB eDRAM cache banks. We use M-sim simulator [30] which is derived from SimpleScalar toolset [39] and integrated with advanced Wattch framework [2]. Table 1 describes the system parameters for our simulation. Note that original Wattch framework only provides process technology parameters from 800 nm to 100 nm, while our evaluation


Initial State:
System ON
Start
PT_Refresh
(for all the cache banks)
Reset the refresh counter the initial value corresponding to (T_{sensor} + 3°C) from the refresh interval table
Wait 10µs
Counter = Counter-1
yes
Counter>0 ?
no
$T_{buf} = T_{sensor}$
Wait 30µs
$T_{sensor} - T_{buf} > 1°C$?
no
yes
Refresh all the cache banks
Is the system turned off?
no
yes
End State:
System OFF
Initial State:
System ON
Start
TV_Refresh
(for each cache bank)
Reset the refresh counter to the initial value corresponding to (T_{sensor} + 3°C) from the refresh interval table
Wait 10µs
Counter = Counter-1
yes
Counter>0 ?
no
$T_{buf} = T_{sensor}$
Wait 30µs
$T_{sensor} - T_{buf} > 1°C$?
no
yes
Refresh the cache bank
Is the system turned off?
no
yes
End State:
System OFF

(a) PT_Refresh (b) TV_Refresh

Fig. 6. Algorithm for our proposed scheme.

Table 1
System parameters.

Parameter	Value		
Initial temperature	50 °C		
Process technology	32 nm		
Number of cores	4 (1 application per core)		
Frequency	3 GHz		
Number of instructions	1B (500 M are fast-forwarded)		
L1 cache (Private)	I$: 64 KB / D$: 64 KB 4-way set associative, SRAM		
LLC (shared)	8 MB 1-die	16 MB 2-die stacked	32 MB 4-die stacked
	8-way set associative, eDRAM		
Number of cache banks (LLC)	8	16	32
TSV specification	50 µm pitch, > 1 K CU TSVs		

Table 2
eDRAM LLC specification.

Parameter		Value
Process technology		32 nm
Cache bank size		1 MB eDRAM
Retention time		100 µs at 95 °C
Energy parameters	Dynamic read energy per access	0.64 nJ
	Leakage power	27.38 mW
	Refresh power	1.83 mW
	Routing energy	8.75 pJ

Table 3
Groups of applications.

	Core 0	Core 1	Core 2	Core 3
Group 01	astar	bzip2	dealII	omnetpp
Group 02	astar	dealII	gobmk	bzip2
Group 03	bzip2	dealII	lbm	astar
Group 04	libquantum	dealII	omnetpp	gobmk
Group 05	libquantum	omnetpp	mcf	dealII
Group 06	libquantum	dealII	astar	omnetpp
Group 07	lbm	astar	omnetpp	bzip2
Group 08	lbm	astar	mcf	omnetpp
Group 09	gobmk	libquantum	mcf	bzip2
Group 10	gobmk	libquantum	mcf	astar

targets on 32 nm process technology. Thus, we added the 32 nm process technology parameters to Wattch by using the scaling method described in [12]. In addition, we use HotSpot thermal simulation tool [8] to evaluate temperatures of eDRAM stacked cache banks. We use a floorplan of Alpha 21364-like processor for thermal simulation, which is scaled down to 32 nm process technology. The detailed specification of eDRAM LLC is described in Table 2, which is derived from CACTI 6.5 [32] based on the conventional eDRAM model [3].

For benchmark applications, we use eight different LLC sensitive applications [10] which have higher misses per kilo instructions (MPKI) than the other applications from SPEC CPU2006 benchmark suite [40]. This is a conservative approach to show our scheme is beneficial at any case. Note memory intensive applications further raise the LLC temperature compared to the other applications. Since our scheme adopts different refresh interval depending on temperature, it is less beneficial, when the actual temperature is close to the worst-case temperature (95 °C). We organize ten groups of applications. Each group consists of four randomly selected applications and different applications are assigned to different cores. Table 3 shows the groups of applications used in our evaluation.

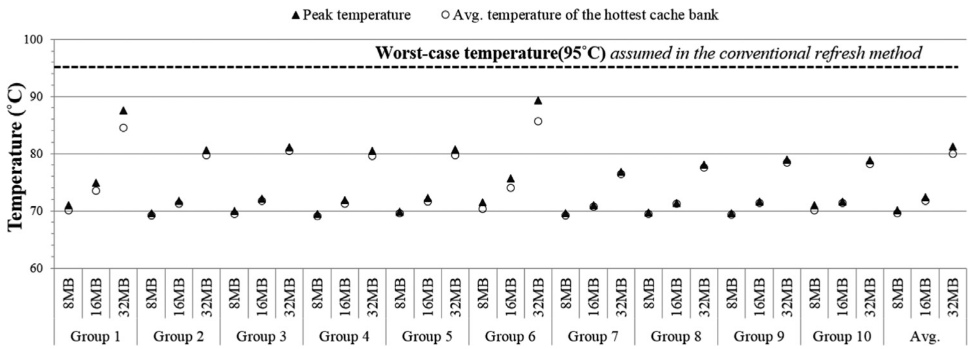


Fig. 7. Temperature of the hottest cache bank during execution. Note that the peak temperature is the highest temperature in eDRAM LLC during execution.

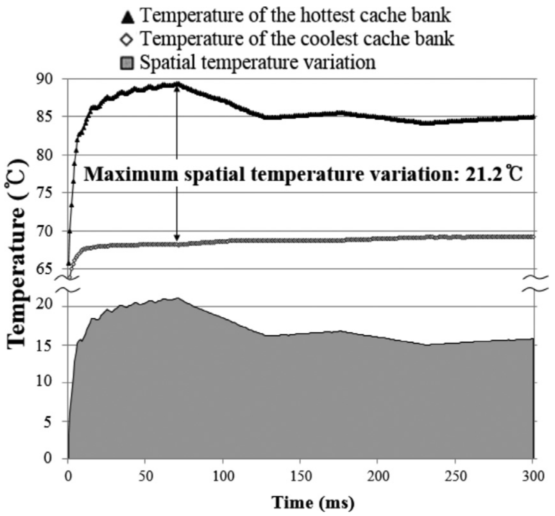


Fig. 8. Runtime temperature of the hottest/coolest cache bank. (Group 6 on 32 MB eDRAM LLC).

5.2. Temperature/retention time

We evaluate temperature behavior of eDRAM cache banks in the three different cache configurations (8 MB, 16 MB, and 32 MB eDRAM LLC) when executing ten groups of applications.

Fig. 7 shows the temperature of the hottest cache bank during execution. Note that PT_Refresh adopts the refresh interval based on the temperature of the hottest bank. As shown in Fig. 7, the average temperature of the hottest cache bank gets higher as the number of stacked cache dies increases (in other words, cache size increases). Nevertheless, the average temperature of the hottest cache bank is much lower than 95 °C (the worst-case temperature assumed in the conventional refresh method), in all cases. Even the peak temperature is at most 89.4 °C (Group 6 on 32 MB eDRAM LLC), which is 5.6 °C lower than 95 °C. The temperature of the hottest cache bank in the 8 MB (1-die), 16 MB (2-die stacked), and 32 MB (4-die stacked) eDRAM LLC is 69.6 °C, 71.8 °C, and 80.02 °C, on average, respectively. Hence, the retention time is expected to be much longer than 100 µs.

Furthermore, while the hottest cache bank is at most 89.4 °C, temperatures of the other cache banks are expected to be much lower. For example, Fig. 8 shows the runtime temperature of the hottest/coolest cache bank when executing Group 6 on 32 MB eDRAM LLC. Note that we define the spatial temperature variation as the temperature difference between the hottest cache bank and the coolest cache bank.

As shown in Fig. 8, spatial temperature variation is steadily large. The maximum spatial temperature variation is 21.2 °C in this case. Thus, we evaluate the spatial temperature variation of eDRAM cache banks in Fig. 9. As shown in Fig. 9, there are considerable spatial temperature variations in all cases. Especially, the spatial temperature variation gets larger, as the number of stacked cache dies increases. Hence, the retention time is expected to be different even across the cache banks.

Fig. 10 shows the retention time of eDRAM cache banks. As shown in Fig. 10, in the case of the 8 MB eDRAM LLC, the retention time of the hottest cache bank is 136.7 µs, on average, which is 36.7% longer than 100 µs. Even in the case of the 32 MB eDRAM LLC which is much hotter than 8 MB eDRAM LLC, the retention time of the hottest cache bank is 117.9 µs, on average, which is 17.9% longer than 100 µs. Note that PT_Refresh adopts the refresh interval based on the retention time of the hottest cache bank. As the number of stacked cache dies increases, the retention time difference becomes larger, due to the spatial temperature variation. For example, as shown in Fig. 10, in the case of 32 MB eDRAM LLC, the retention time difference is 15.6 µs, on average. Note that

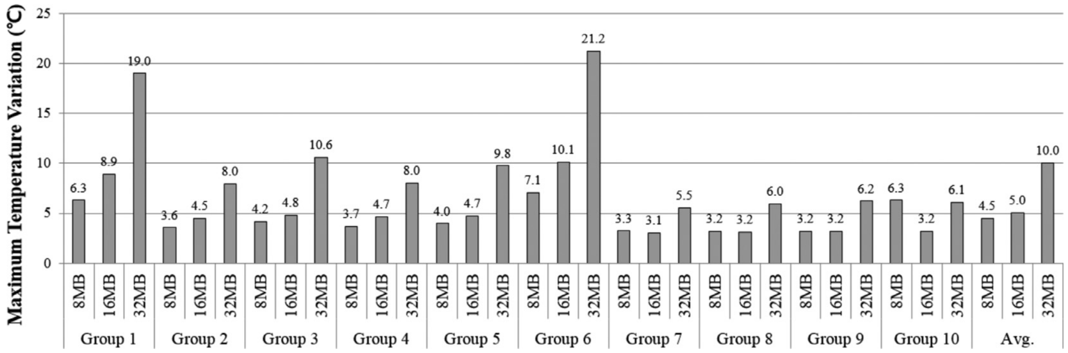


Fig. 9. Maximum spatial temperature variation of eDRAM cache banks.

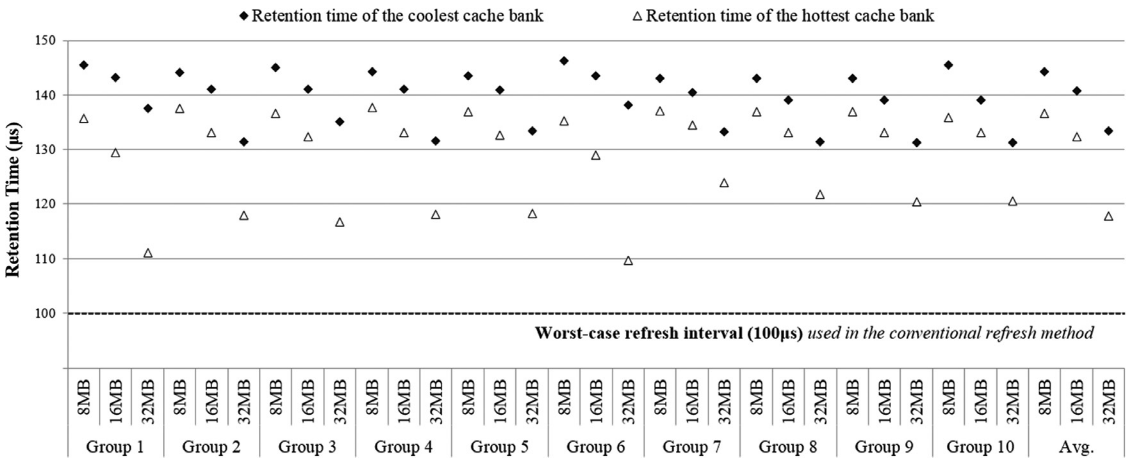


Fig. 10. Retention time difference across eDRAM stacked cache banks during execution. Note that retention times of the hottest cache bank and the coolest cache bank are estimated at the time when the retention time difference across eDRAM stacked cache banks is largest during execution.

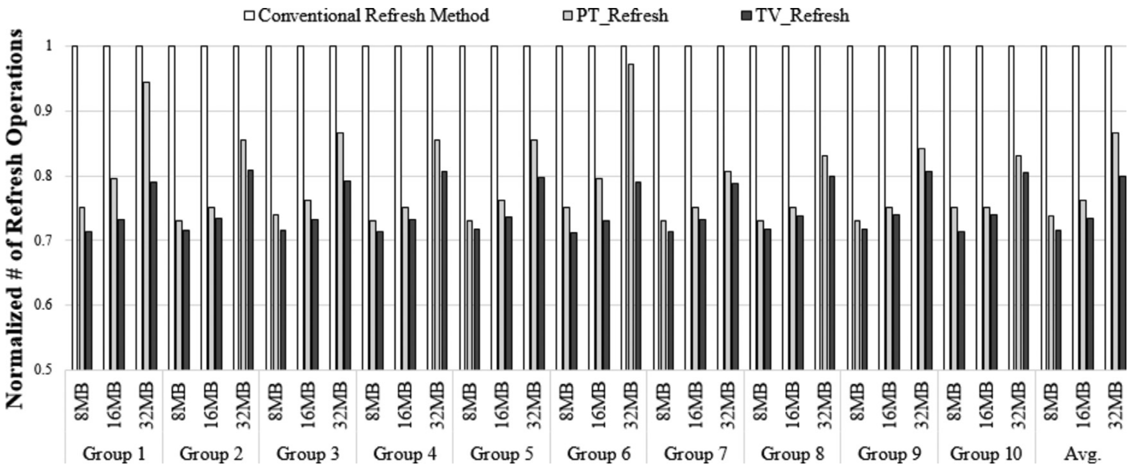


Fig. 11. Normalized number of refresh operations. Note that all results are normalized to the conventional refresh method using 100 µs refresh interval for all cache banks.

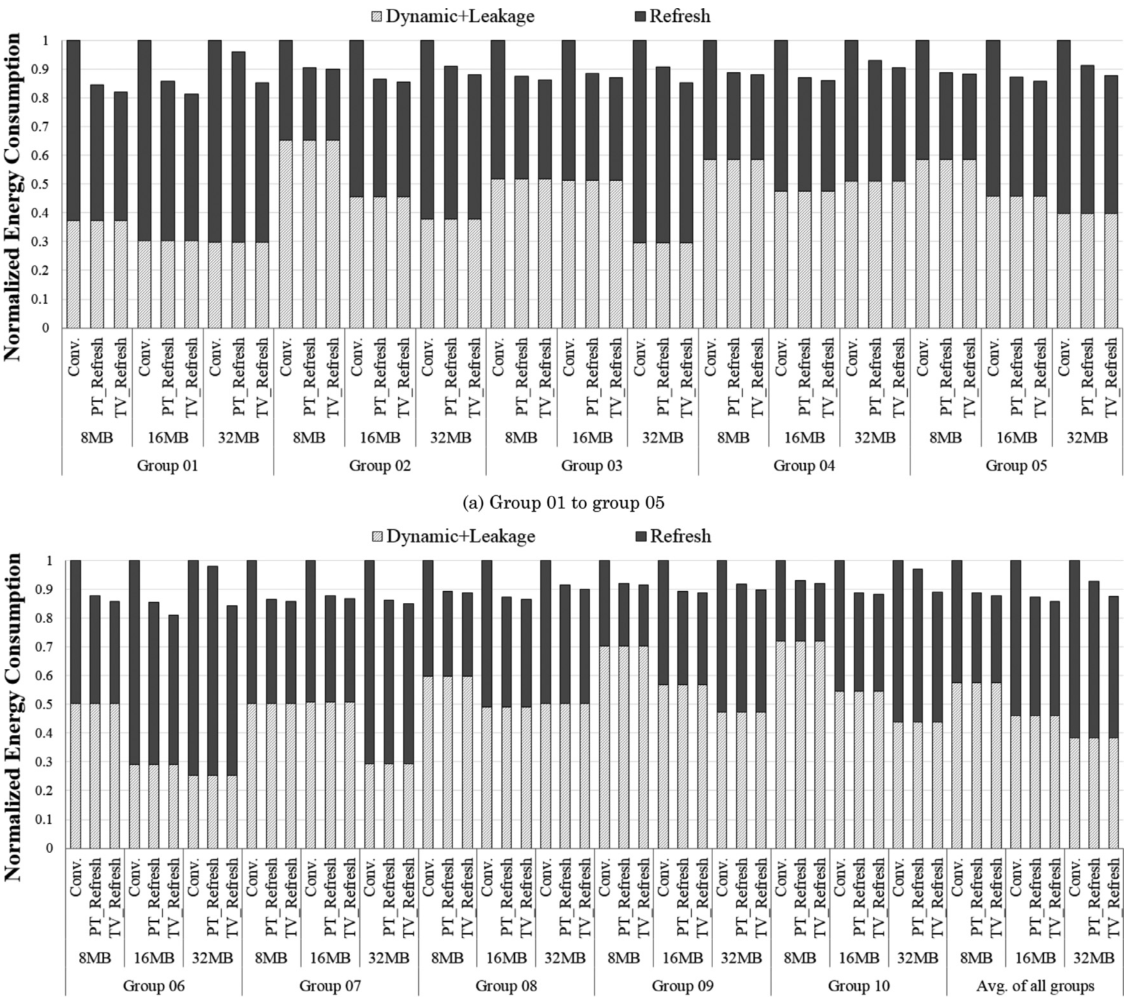
the shortest retention time is 117.9 µs, on average. In this case, TV_Refresh is likely to further reduce the number of refresh operations compared to PT_Refresh, since it considers retention times of each cache bank, while PT_Refresh only considers the retention time of the hottest cache bank**.**

5.3. Number of refresh operations

In this section, we evaluate PT_Refresh and TV_Refresh compared to the conventional refresh method using 100 µs refresh interval for all cache banks, in terms of number of refresh operations. We evaluate the number of refresh operations when the applications are running for 300 ms. Note that, in the case of the conventional refresh method, the number of refresh operations is 24.0 K, 48.0 K, and 96.0 K in the 8 MB, 16 MB, and 32 MB eDRAM LLC, respectively. Fig. 11 shows the number of refresh operations normalized to that of conventional refresh method.

As we explained, the conventional refresh method uses 100 µs refresh interval for all cache banks assuming that temperature of all cache banks is 368 K (95 °C). On the other hand, PT_Refresh dynamically adopts different refresh interval depending on the peak temperature (the maximum temperature among the cache banks, at a given time). Consequently, PT_Refresh further reduces the number of refresh operations as the temperature gets lower. As shown in Fig. 11, PT_Refresh reduces the number of refresh operations by 26.3%, on average, on the 8 MB eDRAM LLC, since the peak temperature is lowest. On the other hand, the reduction rate is smaller in the larger eDRAM LLC, since the peak temperature is high. Nevertheless, PT_Refresh reduces 13.4% of refresh operations even on the 32 MB eDRAM LLC, on average, since the peak temperature is still much lower than 368 K (95 °C). Note that when we adopt PT_Refresh, the number of refresh operations is 17.7 K, 36.6 K, and 83.1 K in the 8 MB, 16 MB, and 32 MB eDRAM LLC, respectively.

While the PT_Refresh only considers the peak temperature, TV_Refresh considers not only the peak temperature but also the spatial temperature variation. Consequently, TV_Refresh reduces more number of refresh operations when the spatial temperature variation is larger. Since the larger eDRAM LLC has more spatial temperature variation, TV_Refresh further reduces the number of refresh operations on 32 MB eDRAM LLC (by up to 18.3%, and 6.7% on average), compared to the PT_Refresh. On the other hand, when there is small spatial temperature variation (such as the case of 8 MB eDRAM LLC), PT_Refresh (which only considers the peak temperature) is good enough to reduce unnecessary refresh operations. Thus, in the case of the 8 MB eDRAM LLC, PT_Refresh and TV_Refresh significantly reduce the number of refresh operations by 26.3% and 28.5%, on average, respectively. Note that when we adopt TV_Refresh, the number of refresh operations is 17.1 K, 35.3 K, and 76.6 K in the 8 MB, 16 MB, and 32 MB eDRAM LLC, respectively.



(a) Group 01 to group 05

(b) Group 06 to group 10 and average of all groups

Fig. 12. Normalized energy consumption of eDRAM LLC. Note that all results are normalized to the conventional refresh method using 100 µs refresh interval for all cache banks.

5.4. Energy consumption

In PT_Refresh, as the number of stacked cache dies increases, the refresh reduction rate becomes much smaller due to the increased peak temperature (Fig. 11). However, the portion of the refresh energy consumption in the LLC energy consumption increases with the number of stacked cache dies. Therefore, when adopting PT_Refresh, the eDRAM LLC energy reduction is still significant on the 32 MB eDRAM LLC.

As shown in Fig. 12, PT_Refresh reduces the average eDRAM LLC energy consumption by 11.1% (8 MB), 12.7% (16 MB), and 7.4% (32 MB), respectively, compared to the conventional refresh method. The energy reduction is highest in the 16 MB eDRAM LLC, since the refresh energy reduction is nearly as much as that of 8 MB eDRAM LLC, while the refresh energy portion increases greatly.

Different from PT_Refresh, the refresh reduction rate of TV_Refresh only decreases slightly though the number of stacked dies increases (Fig. 11). On the other hand, the refresh energy portion increases with the number of stacked cache dies, same as in PT_Refresh. Hence, TV_Refresh reduces significant portion of total eDRAM LLC energy consumption as the number of stacked dies increases. As shown in Fig. 12, TV_Refresh reduces the average eDRAM LLC energy consumption by 12.1% (8 MB), 14.3% (16 MB), and 12.5% (32 MB), respectively, compared to the conventional refresh method. In the case of 32 MB eDRAM LLC, the average energy saving of TV_Refresh is 5.1% larger, even compared to PT_Refresh.

5.5. Comparison with the other temperature-aware refresh schemes

We compare our proposed TV_Refresh with TCSR [38], Bank-wise Refresh [29], and Ideal_TV_Refresh. Note that we assume that Ideal_TV_Refresh always gets an exact temperature of a cache bank from the thermal sensor without sensor measurement error. Thus, Ideal_TV_Refresh refreshes each cache bank with its own optimal refresh interval, which corresponds to its retention time at the actual temperature (without considering temperature margin). In addition, we evaluate TCSR assuming that the eDRAM stacked cache adopts the retention time at 85 °C as the refresh interval of whole eDRAM cache below 85 °C, and adopts the half of the retention time at 85 °C as the refresh interval of whole eDRAM cache above 85 °C.

Fig. 13 shows the energy reduction of TCSR, Bank-wise Refresh, TV_Refresh, and Ideal_TV_Refresh, compared to the conventional refresh method, when we execute ten groups of applications described in Table 3. Note that the energy consumption of TV_Refresh and Ideal_TV_Refresh includes the energy overhead of TV_Refresh (the energy consumed by refresh interval tables, thermal sensors, and refresh counters), which will be described in Section 5.6. As shown in Fig. 13, though Bank-wise Refresh

Table 4
Overheads.

	Response time	Energy (per read access)	Area
Comparator [15]	< 210 ps	5.7 nJ	16 μm^2
Refresh interval table (CAM) [27]	3.9 ns	57 fJ	0.0029 mm^2
Thermal sensor [9]	< 28.7 µs	0.64 nJ	0.0016 mm^2
Refresh counter	negligible	negligible	negligible
Summary (on 32 MB eDRAM LLC)			
PT_Refresh	< 28.8 µs	< 1.3% of the LLC energy	< 4.6% of the LLC area
TV_Refresh		< 0.5% of the LLC energy	

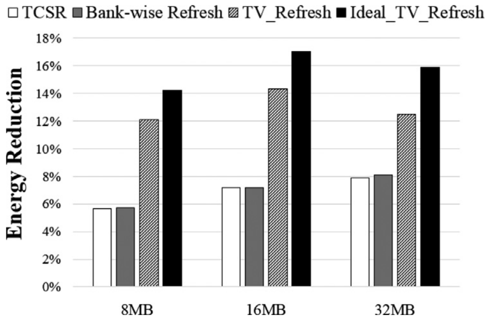


Fig. 13. Average energy reduction compared to the conventional refresh method. Note that all results are normalized to the conventional refresh method using 100 µs refresh interval for all cache banks.

uses temperature-aware refresh intervals, it reduces slightly more energy than TCSR, since it has only small number of (thirteen) refresh intervals depending on temperature from 35 °C to 105 °C. Contrary to Bank-wise Refresh, our proposed TV_Refresh uses more fine-grained temperature-aware refresh intervals. Hence, TV_Refresh reduces 6.4%, 7.1%, and 4.3% more average energy consumption than Bank-wise Refresh in the 8 MB, 16 MB, and 32 MB eDRAM LLC, respectively. As shown in Fig. 13, TV_Refresh has less energy reduction rate in the 32 MB eDRAM LLC, compared to that in the 16 MB eDRAM LLC; temperatures of cache banks increase with eDRAM LLC capacity, as shown in Fig. 7. Due to the increased temperature, retention times of cache banks decrease exponentially with eDRAM LLC capacity, as shown in Fig. 10. Consequently, when we use TV_Refresh, temperature-aware refresh intervals for the cache banks in the 32 MB eDRAM LLC are much closer to 100 µs (the conventional refresh interval) than those in the 8 MB and 16 MB eDRAM LLC. Thus, in the case of TV_Refresh, the 32 MB eDRAM LLC has less energy reduction rate, compared to the 16 MB eDRAM LLC. However, TV_Refresh still reduces significant energy consumption of 32 MB eDRAM LLC, compared to the conventional refresh method.

The average energy reduction of TV_Refresh is 2.1%, 2.7%, and 3.5% less than that of Ideal_TV_Refresh in the 8 MB, 16 MB, and 32 MB eDRAM LLC, respectively. The energy reduction differences between our proposed TV_Refresh and Ideal_TV_Refresh are due to additional 3 °C temperature margin of our proposed TV_Refresh, considering temperature sensor error. However, in order to prevent data loss, it is indispensable to have temperature margin, when we determine refresh interval based on the value of temperature sensor.

5.6. Overheads

We estimate the overhead of our proposed scheme in the perspective of response time, energy, and area, which is described in Table 4. We assume that the refresh interval table is composed of content-addressable memory (CAM) [27]. In addition, we estimate the elapsed time to refresh an eDRAM cache bank to verify that the refresh operations can be finished before the retention time expires.

Firstly, we analyze response time of the components used in our proposed scheme. The eDRAM model used in our paper [3] assumes that the worst-case temperature is 95 °C. In the worst-case scenario, our proposed scheme changes refresh interval every 100 µs assuming that the temperature of all the cache banks is 95 °C. Thus, the sum of response time of all the components is reasonable as long as it is shorter than 100 µs. As described in Section 4.3, our proposed scheme monitors temperatures of all the cache banks every 30 µs. Our response time analysis shows that it is appropriate to use 30 µs period for temperature monitoring. As shown in Table 4, the sum of response time of the comparator, the refresh interval table, and the refresh counter is negligible since it is much shorter than 0.1 µs. The thermal sensor has the longest response time among the components. According to [9], the response time of the thermal sensor becomes faster as temperature increases (from 28.7 µs at 40 °C to 1.2 µs at 110 °C). Though the thermal sensor has considerable response time (28.7 µs) at the worst case, the sum of response time of all the components is still acceptable, since it is less than 30 µs.

Secondly, we analyze energy consumption of the components used in our proposed scheme. As described in Table 4, the comparator consumes 5.7 nJ per access [15]. In our simulation, the energy consumption of the comparator is 1.1% of the eDRAM LLC energy consumption (32 MB), on average. The refresh interval table has small read access energy (9.5 fJ per bit) [27]. Since we just need to access several bits in the refresh interval table, the energy overhead of the refresh interval table (57 fJ per access) is negligible. The thermal sensor consumes 0.64 nJ per read access. When using 32 MB eDRAM LLC, the energy overhead of PT_Refresh is less than 1.3% of the eDRAM LLC energy consumption (1.1% for comparators, much less than 0.01% for the refresh interval tables and 0.1% for the thermal sensors). Compared to PT_Refresh, TV_Refresh needs more refresh interval tables but it does not require comparators. Thus, the energy overhead of TV_Refresh is less than 0.5% of the eDRAM LLC energy consumption (0.4% for the refresh interval tables and 0.1% for the thermal sensors), which is negligible compared to the significant energy saving (12.5%, on average).

Thirdly, we estimate area overhead of the components used in our proposed scheme. As described in Table 4, the comparator occupies 16 μm^2 which is negligible compared to the cache bank area (0.1 mm^2). The refresh interval table requires 64 Bytes (101*5 bits = 505 bits ≈ 64 Bytes). When the refresh interval table is composed of CAM array, its area is 0.0029 mm^2, which is 2.9% of the cache bank area. The thermal sensor occupies 0.0016 mm^2 which is 1.6% of the cache bank area [9]. In the case of the refresh counter, PT_Refresh needs a refresh counter for a whole eDRAM LLC (5 bits). On the other hand, TV_Refresh needs a refresh counter for each eDRAM stacked cache banks. Thus, in the 32 MB eDRAM LLC, TV_Refresh requires 20 Bytes (5 bits per bank * 32 banks = 160 bits = 20 Bytes) for the refresh counters. Even in that case, the required capacity (20 Bytes) is still negligible compared to the whole eDRAM LLC capacity (32 MB). When we

Table 5
The elapsed time to refresh an eDRAM cache bank in the 8 MB eDRAM cache.

	8 MB eDRAM cache (8 banks, 8-way, and 32B cache line)
H-tree input delay (①)	0.162 ns
Wordline precharge delay (②)	0.163 ns
Bitline precharge delay (③)	0.740 ns
$T_{\text{refresh\_1st\_cache\_line}}$ (①+②+③=④)	1.065 ns
$T_{\text{refresh\_each\_of\_the\_other\_cache\_lines}}$ (②+③=⑤)	0.903 ns
$T_{\text{refresh\_an\_eDRAM\_cache\_bank}}$ (④*1+⑤*4095)	3.698 µs (1.065 ns*1+0.903 ns*4095)

assume that the components are routed using global signal nets, the routing area overhead depends on the number of components and chip size. However, we anticipate negligible area occupied by these global nets because they are non-timing critical and thus require no buffering or global metal layers. This is mainly because the time constant for on-chip temperature change is on the order of micro- to milli-seconds, not nano-seconds. Even if TSVs are used in these global nets, the RC parasitic impact is still negligible; the parasitic values are on the order of single digit ohm and single digit femto-farad. Consequently, the area overhead is less than 4.6% of the cache area, as shown in Table 4. In summary, PT_Refresh and TV_Refresh are capable of changing refresh interval depending on temperature with small response time, energy, and area (including routing area) overhead.

Finally, we verify that the refresh operations can be finished before the retention time expires. To calculate the elapsed time to refresh an eDRAM cache bank, we evaluate the time components of the 8 MB eDRAM cache (8-banks, 8-way set associative, and 32B cache line size) by using CACTI 6.5, since the eDRAM LLCs have 8 MB eDRAM cache per cache die. Table 5 shows the elapsed time to refresh an eDRAM cache bank in the 8 MB eDRAM cache. As described in Table 5, the elapsed time to refresh the first cache line is 1.065 ns, since it consists of H-tree input delay (0.162 ns), wordline precharge delay (0.163 ns), and bitline precharge delay (0.740 ns). After the first cache line is refreshed, the elapsed time to refresh each of the other cache lines is 0.903 ns, since refreshing each of the other cache lines consumes wordline precharge delay and bitline precharge delay. In summary, since the number of cache lines is 4096 (1 MB / (32B * 8-way)), the elapsed time to refresh an eDRAM cache bank is 3.698 µs (1.065 ns*1 + 0.903 ns*4095). The elapsed time to refresh an eDRAM cache bank (3.698 µs) is much shorter than the worst-case retention time (100 µs at 95 °C) of our eDRAM model. Thus, we are sure that refresh operations are always finished before retention time expires.

6. Conclusion

In this paper, we proposed a novel temperature-aware refresh scheme to reduce unnecessary refresh operations. Since our proposed scheme adopts fine-grained temperature-aware refresh intervals considering peak temperature (PT_Refresh) and spatial temperature variation (TV_Refresh), it reduces significant refresh energy overhead. Compared to the conventional refresh method, our proposed scheme reduces refresh operations by 28.5% (on 32 MB eDRAM LLC), on average, without losing data stored in eDRAM. As a result, our proposed scheme reduces the overall eDRAM LLC energy consumption by 12.5% (on 32 MB eDRAM LLC), on average, with small area overhead. Even in the case of eDRAM LLC larger than 32 MB, which has cache dies more than 4 dies, our proposed scheme will be effective to reduce refresh energy overhead since the larger eDRAM LLC has the higher peak temperature and the larger spatial temperature variation. We expect that our proposed refresh scheme will be used for eDRAM stacked caches.

Acknowledgment

Sung Woo Chung is the corresponding author of this paper. This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP) (No. 2015R1A2A1A15055435 and No. 2014R1A2A1A11054390) and Korea University.

References

[1] S. Borkar, 3D integration for energy efficient system design, in: Proceedings of the 48th Annual Design Automation Conference (DAC), June 2011, pp. 214–219.
[2] D. Brooks, V. Tiwari, M. Martonosi, Wattch: a framework for architectural-level power analysis and optimizations, in: Proceedings of the 27th Annual International Symposium on Computer Architecture (ISCA), 28, May 2000, pp. 83–94.
[3] M.-T. Chang, P. Rosenfeld, S.-L. Lu, B. Jacob, Technology comparison for large last-level caches (L3Cs): low-leakage SRAM, low write-energy STT-RAM, and refresh-optimized eDRAM, in: Proceedings of the IEEE International Symposium on High Performance Computer Architecture (HPCA), February 2013, pp. 143–154.
[4] M. Ghosh, H.-H.S. Lee, Smart refresh: an enhanced memory controller design for reducing energy in conventional and 3D die-stacked DRAMs, in: Proceedings of the IEEE/ACM International Symposium on Microarchitecture (MICRO), 2007, pp. 134–145.
[5] J. Golz, J. Safran, B. He, D. Leu, M. Yin, T. Weaver, A. Vehabovic, Y. Sun, A. Cestero, B. Himmel, G. Maier, C. Kothandaraman, D. Fainstein, J. Barth, N. Robson, T. Kirihata, K. Rim, S. Iyer, 3D stackable 32 nm high-K/Metal gate SOI embedded DRAM prototype, in: Proceedings of the IEEE International Symposium on VLSI Circuits, June 2011, pp. 228–229.
[6] Y.-H. Gong, H.B. Jang, S.W. Chung, Performance and cache access time of SRAM-eDRAM hybrid caches considering wire delay, in: Proceedings of the International Symposium on Quality Electronic Design (ISQED), March 2013, pp. 524–530.
[7] C. Hou, J. Li, C. Lo, D. Kwai, Y. Chou, C. Wu, An FPGA-based test platform for analyzing data retention time distribution of DRAMs, in: Proceedings of the IEEE International Symposium on VLSI Design, Automation, and Test (VLSI-DAT), April 2013, pp. 1–4.
[8] W. Huang, S. Ghosh, S. Velusamy, K. Sankaranarayanan, K. Skadron, M.R. Stan, HotSpot: A compact thermal modeling methodology for early-stage VLSI design, IEEE Trans. Very Large Scale Integr. (VLSI) Syst. 14 (5) (May 2006) 501–513.
[9] P. Ituero, M. López-Vallejo, C. López-Barrio, A 0.0016mm2 0.64nJ leakage-based CMOS temperature sensor, Sensors 2013 13 (9) (2013) 12648–12662.
[10] A. Jaleel, Memory Characterization of Workloads using Instrumentation-Driven Simulation, Intel Corporation, VSSAD, 2010 , http://www.jaleels.org/ajaleel/workload/.
[11] S. Jin, J. Yi, J.H. Choi, D.G. Kang, Y.J. Park, H.S. Min, Prediction of data retention time distribution of DRAM by physics-based statistical simulation, IEEE Trans. Electron Devices 52 (November 2005) 2422–2429.
[12] A.B. Kahng, B. Li, L.-S. Peh, K. Samadi, ORION 2.0: A fast and accurate NoC power and area model for early-stage design space exploration, in: Proceedings of the IEEE/ACM International Conference on Design, Automation and Test in Europe (DATE), April 2009, pp. 423–428.
[13] R. Kalla, B. Sinharoy, W.J. Starke, M. Floyd, Power7: IBM's next-generation server processor, IEEE Micro Mag. 30 (March-April 2010) 7–15.
[14] K. Kang, J. Kim, S. Yoo, C.-M. Kyung, Temperature-aware integrated DVFS and power gating for executing tasks with runtime distribution, IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. 29 (2) (2010) 1381–1394.
[15] K. Khalil, M. Abbas, M. Abdelgawad, A low-power low-delay dispersion comparator for high-speed level-crossing ADCs, in: Proceedings of the Saudi International Electronics, Communications and Photonics Conference (SIECPC), April 2013, pp. 1–6.
[16] C. Kim, D. Burger, S.W. Keckler, An adaptive, non-uniform cache structure for wire-delay dominated on-chip caches, in: Proceedings of the IEEE International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS), October 2002, pp. 211–222.
[17] D.H. Kim, S.K. Lim, Through-silicon-via-aware delay and power prediction model for buffered interconnects in 3D ICs, in: Proceedings of the ACM/IEEE International Workshop on System Level Interconnect Prediction (SLIP'10), June 2010, pp. 25–32.
[18] J. Kong, S.W. Chung, K. Skadron, Recent thermal management techniques for microprocessors, ACM Comput. Surv. 44 (3) (June 2012).
[19] J.S. Lee, K. Skadron, S.W. Chung, Predictive temperature-aware DVFS, IEEE Trans. Comput. 59 (1) (2010) 127–133.
[20] F. Li, C. Nicopoulos, T. Richardson, Y. Xie, V. Narayanan, M. Kandemir, Design and management of 3D chip multi processors using network-in-memory, in: Proceedings of the IEEE/ACM International Symposium on Computer Architecture (ISCA), 34, May 2006, pp. 130–141.
[21] J. Liu, B. Jaiyen, Y. Kim, C. Wilkerson, O. Multu, An experimental study of data retention behavior in modern DRAM devices: implications for retention time profiling mechanisms, in: Proceedings of the IEEE/ACM International Symposium on Computer Architecture (ISCA), June 2013.

[22] J. Liu, B. Jaiyen, R. Veras, O. Mutlu, RAIDR: retention-aware intelligent DRAM refresh, in: Proceedings of the IEEE/ACM International Symposium on Computer Architecture (ISCA), June 2012, pp. 1–12.
[23] G.H. Loh, Y. Xie, B. Black, Processor design in 3D die-stacking technologies, IEEE Micro Mag. 27 (May-June 2007) 31–48.
[24] G.L. Loi, B. Agrawal, N. Srivastava, S. Lin, T. Sherwood, K. Banerjee, A thermally-aware performance analysis of vertically integrated (3-D) processor-memory hierarchy, in: Proceedings of the IEEE/ACM/EDAC Design Automation Conference (DAC), July 2006, pp. 991–996.
[25] J. Meng, K. Kawakami, A.K. Coskun, Optimizing energy efficiency of 3-D multicore systems with stacked DRAM under power and thermal constraints, in: Proceedings of the IEEE/ACM/EDAC Design Automation Conference (DAC), June 2012, pp. 648–655.
[26] P. Nair, C.-C. Chou, M.K. Qureshi, Refresh pausing in DRAM memory systems, ACM Trans. Archit. Code Optim. 11 (1) (2014) 10–35.
[27] K. Pagiamtzis, A. Sheikholeslami, Content-addressable memory (CAM) circuits and architectures: a tutorial and survey, IEEE J. Solid-State Circ. 41 (3) (March 2006) 712–727.
[28] K. Puttaswamy, G.H. Loh, Thermal analysis of a 3D die-stacked high-performance microprocessor, in: Proceedings of the ACM Great Lakes Symposium on VLSI (GLSVLSI'06), April-May 2006, pp. 19–24.
[29] M. Sadri, M. Jung, C. Weis, N. When, L. Benini, Energy optimization in 3D MPSoCs with wide-I/O DRAM using temperature-variation aware bank-wise refresh, in: Proceedings of the IEEE/ACM International Conference on Design, Automation and Test in Europe (DATE), poster session, March 2014.
[30] J.J. Sharkey, D. Ponomarev, K. Ghose, M-sim: A Flexible, Multithreaded Architectural Simulation Environment," Technical Report CS-TR-05-DP01, Department of Computer Science, State University of New York at Binghamton, October 2005.
[31] J. Stuecheli, D. Kaseridis, H.C. Hunter, L.K. John, "Elastic refresh: techniques to mitigate refresh penalties in high density memory, in: Proceedings of the IEEE/ACM International Symposium on Microarchitecture (MICRO), December 2010, pp. 375–384.
[32] S. Thoziyoor, N. Muralimanohar, J.H. Ahn, N.P. Jouppi, CACTI 5.1, Technical Report HPL-2008-20, HP Laboratories, 2008.
[33] Y. Wang, U. Arslan, N. Bisnik, R. Brain, S. Ghosh, F. Hamzaoglu, N. Lindert, M. Meterelliyoz, J. Park, S. Tomishima, K. Zhang, Retention time optimization for eDRAM in 22 nm tri-gate cmos technology, in: Proceedings of the IEEE International Electron Devices Meeting, December 2013, pp. 240–243.
[34] M. Wordeman, J. Silberman, G. Maier, M. Scheuermann, A 3D system prototype of an eDRAM cache stacked over processor-like logic using through-silicon vias, in: Proceedings of the IEEE International Solid-State Circuits Conference, February 2012, pp. 186–187.
[35] Y. Zhang, D. Parikh, K. Sankaranarayanan, K. Skadron, M. Stan, HotLeakage: A Temperature-aware Model of Subthreshold and Gate Leakage for ArchitectsTR-CS-2003-05, University of Virginia, Dept. of Computer Science, March 2003.
[36] Intel D54250WYKH Haswell NUC Kit with 2.5" Drive Slot Mini-Review, http://www.anandtech.com/show/8028/intel-d54250wykh-haswell-nuc-kit-with-25-drive-slot-minireview/5/ (accessed 18.05.14).
[37] JESD79-3F, JEDEC Committee JC-42.3, 2012.
[38] MT46H32M16LC, "Mobile Low-Power DDR SDRAM, 512Mb: x16, x32 Mobile LPDDR SDRAM Features" http://www.micron.com/ (accessed 14.01.09).
[39] SimpleScalar toolset, http://www.simplescalar.com/ (accessed 22.03.11).
[40] SPEC CPU2006, http://www.spec.org/cpu2006/ (accessed 07.09.11).

Young-Ho Gong received the BS degree in the Division of Computer and Communication Engineering from Korea University, in 2012. He is currently a Ph.D. student in the Department of Computer Science, Korea University. His research interests include energy-efficient cache architecture and low-power DRAM refresh.

Jae Min Kim received the BS and PhD degrees in Computer Science from Korea University, in 2010 and 2015, respectively. He is currently a research engineer in the Samsung Electronics DS. His research interests include user-aware design, computer architecture and low-power design.

Sung Kyu Lim received the BS, MS, and PhD degrees from the Computer Science Department, University of California, Los Angeles (UCLA), in 1994, 1997, and 2000, respectively. He is currently a professor in the School of Electrical and Computer Engineering, Georgia Institute of Technology. His research interests include the architecture, circuit, and physical design for 3-D ICs and 3-D system-in-packages.

Sung Woo Chung received the BS, MS, and PhD degrees in Electrical Engineering and Computer Science from Seoul National University, in 1996, 1998, and 2003, respectively. He is currently a professor in the Department of Computer Science, Korea University. His research interests include low-power design, temperature-aware design, and user-aware design.