CAM-based retention-aware DRAM (CRA-DRAM) for refresh power reduction

Yong Ye^{1a)}, Yuan Du¹, Weiliang Jing², Xiaoyun Li¹,
Zhitang Song¹, and Bomy Chen²

¹ State Key Laboratory of Functional Materials for Informatics, Shanghai Institute of Micro-system and Information Technology, Chinese Academy of Sciences, Shanghai, 200050, China

² Shanghai Xinchu Integrated Circuit Inc., 200122, China

a) yeyong1991@mail.sim.ac.cn

Abstract: As the main component for modern main memory system, DRAM stores data by capacitors, which must be refreshed periodically to keep the charges. As the size and speed of DRAM devices continue to increase, the overhead of refresh has caused a great power and performance dissipation. In this paper, we proposed a CAM (content-addressable memory)-based Retention-Aware DRAM (CRA-DRAM) system, a hardware implementation that uses CAM and RAM to locate and replace the leaky cells at the IO granularity. Then the entire DRAM is refreshed at a much lower rate. With IO-granularity address of leaky cells stored in CAM at the profiling stage, each access address to CRA-DRAM would be searched to determine where the data are read from or written to. We proved the IO-granularity data replacement technique is completely compatible with the JEDEC standard. The experimental results show that when the refresh period is increased by 6×, CRA-DRAM has a 82.5% refresh reduction, an average DRAM energy reduction of 29.1% and an average system performance improvement of 8.3%. Without modification to memory controller, OS and DRAM devices, CRA-DRAM is quite promising to be applied in DIMM, HBM and HMC.

Keywords: DRAM, CAM, retention time, refresh, power

Classification: Electron devices, circuits and modules

References

- [1] J. Liu, *et al.*: “RAIDR: Retention-aware intelligent DRAM refresh,” ISCA (2012) 1 (DOI: [10.1145/2366231.2337161](https://doi.org/10.1145/2366231.2337161)).
- [2] K. Kim and J. Lee: “A new investigation of data retention time in truly nanoscaled DRAMs,” IEEE Electron Device Lett. **30** (2009) 846 (DOI: [10.1109/LED.2009.2023248](https://doi.org/10.1109/LED.2009.2023248)).
- [3] R. K. Venkatesan, *et al.*: “Retention-aware placement in DRAM (RAPID): Software methods for quasi-non-volatile DRAM,” HPCA (2006) 155 (DOI: [10.1109/HPCA.2006.1598122](https://doi.org/10.1109/HPCA.2006.1598122)).
- [4] J. Mukundan, *et al.*: “Understanding and mitigating refresh overheads in high-density DDR4 DRAM systems,” ISCA (2013) 48 (DOI: [10.1145/2485922](https://doi.org/10.1145/2485922)).

- 2485927).
- [5] I. Arsovski, *et al.*: “A 32 nm 0.58-fJ/Bit/Search 1-GHz ternary content addressable memory compiler using silicon-aware early-predict late-correct sensing with embedded deep-trench capacitor noise mitigation,” *IEEE J. Solid-State Circuits* **48** (2013) 932 (DOI: [10.1109/JSSC.2013.2239092](https://doi.org/10.1109/JSSC.2013.2239092)).
 - [6] 8Gb: $\times 4$, $\times 8$, $\times 16$ DDR4 SDRAM (2015) <http://www.micron.com>.
 - [7] A. T. Do, *et al.*: “0.77 fJ/bit/search content addressable memory using small match line swing and automated background checking scheme for variation tolerance,” *IEEE J. Solid-State Circuits* **49** (2014) 1487 (DOI: [10.1109/JSSC.2014.2316241](https://doi.org/10.1109/JSSC.2014.2316241)).
 - [8] Y. Tsukamoto, *et al.*: “1.8 Mbit/mm² ternary-CAM macro with 484 ps search access time in 16 nm Fin-FET bulk CMOS technology,” *VLSIC* (2015) 274 (DOI: [10.1109/VLSIC.2015.7231286](https://doi.org/10.1109/VLSIC.2015.7231286)).
 - [9] M. Chang, *et al.*: “A 3T1R nonvolatile TCAM using MLC ReRAM with sub-1 ns search time,” *ISSCC* (2015) 318 (DOI: [10.1109/ISSCC.2015.7063054](https://doi.org/10.1109/ISSCC.2015.7063054)).
 - [10] C.-C. Lin, *et al.*: “A 25 6b-wordlength ReRAM-based TCAM with 1 ns search-time and 14 $\times$ improvement in word length-energy efficiency-density product using 2.5T1R cell,” *ISSCC* (2016) 136 (DOI: [10.1109/ISSCC.2016.7417944](https://doi.org/10.1109/ISSCC.2016.7417944)).
 - [11] T. Ohsawa, *et al.*: “A 1.5 nsec/2.1 nsec random read/write cycle 1 Mb STT-RAM using 6T2MTJ cell with background write for nonvolatile e-memories,” *VLSIC* (2015) C110.
 - [12] B. Rooseleer, *et al.*: “A 65 nm, 850 MHz, 256 kbit, 4.3 pJ/access, ultra low leakage power memory using dynamic cell stability and a dual swing data link,” *IEEE J. Solid-State Circuits* **47** (2012) 1784 (DOI: [10.1109/JSSC.2012.2191316](https://doi.org/10.1109/JSSC.2012.2191316)).
 - [13] TN-41-01: Calculating Memory System Power for DDR3 (2007).
 - [14] The gem5 Simulator (2016) <http://www.gem5.org>.
 - [15] P. Rosenfeld, *et al.*: “DRAMsim2: A cycle accurate memory system simulator,” *IEEE Comput. Archit. Lett.* **10** (2011) 16 (DOI: [10.1109/L-CA.2011.4](https://doi.org/10.1109/L-CA.2011.4)).

1 Introduction

Modern main memory is commonly composed of dynamic random access memory (DRAM) devices, which use capacitors to store data. Because of charge leakage, data stored in DRAM must be periodically read out and restored, a process called refresh. The refresh operation causes extra power overhead and downgrades performance for suspending normal access requests. These problems are expected to get worse as the process node continues to scale for higher speed and larger size. Research [1] shows that the refresh rate limits DRAM density scaling: a hypothetical 64 Gb DRAM device would spend 46% of its time and 47% of all DRAM energy for refresh operation, as opposed to typical 4 Gb devices of today that spend 8% of the time and 15% of the DRAM energy on refresh. Experimental investigation [2] indicates that most of the DRAM cells have a longer retention time while there is only a tiny fraction of leaky cells. As shown in Fig. 1, the failure cells with retention time of 512 ms and 1 s only account for about $10^{-5}\%$ and $10^{-3}\%$, respectively. However, current DRAM device refreshes all cells at a same rate (e.g. 64 ms at normal temperature range), which is determined by the leakiest cells in the DRAM device. Based on this point, some techniques have been proposed to reduce the frequency and overhead of DRAM refresh [1, 3, 4]. However, these

retention-aware refresh approaches require either OS-level involvement, or memory controller participation, or DRAM devices modification, which results in great difficulties at implementation level.

Fig. 1. CP curves of data retention failures (based on data from [2])

The goal of this paper is to minimize the number of refresh operations without OS level support, without touching memory controller which is now integrated into CPU chips, and without making modification to JEDEC standard. We firstly propose a CAM-based Retention-Aware DRAM (CRA-DRAM) system, a hardware implementation that improves DRAM refresh period by using the CAM (content-addressable memory) to locate the tiny fraction of DRAM leaky cells at IO granularity and a small-capacity RAM (random access memory) to store data from these leaky cells. DRAM refresh power is reduced significantly as the DRAM is refreshed at a uniform and much lower rate.

2 CRA-DRAM

2.1 Overview of CRA-DRAM

Fig. 2. Architecture of CRA-DRAM with a small-capacity CAM and RAM.

The idea of CRA-DRAM is simple but innovative that it applies a much longer refresh interval to the DRAM device while the leaky cells, which cannot retain their data, are replaced by a small-capacity RAM. The CAM, a special type of memory that provides a nanosecond-scale data search function by comparing the search data

with all the stored data in a single clock cycle [5], is used to achieve fast address search to determine where the effective data are located, as shown in Fig. 2. Note that in DRAM devices, read and write share the same address bus, as well as the bidirectional DQ bus. And the colored lines just represent the datapath. Unlike previous techniques, CRA-DRAM realizes an IO-granularity data replacement. As DDR4 SDRAM supports IO data width of $\times 4$, $\times 8$ and $\times 16$, the data replacement keeps the same data width with the DRAM device. If the retention time of any bits in the IO data is less than the target refresh period we set for the entire DRAM device, the IO data would be transferred into RAM.

Assume that the information of IO addresses that contains DRAM leaky cells has been obtained at the profiling stage and stored in the CAM. Access to DRAM and address search in CAM are handled simultaneously. If a read operation is applied to CRA-DRAM, where the read data are from is determined by the read address, each of which would be searched in the CAM array. If the read address is matched in CAM, data are read from RAM, otherwise data are read from DRAM. The main challenge from our read scheme is the high requirement for ultrafast search speed in CAM. Firstly, where the read data come from is determined by the match result that generates data selection signal to the multiplexer. Secondly, the read address to RAM also depends on the match result. That's why CAM is used as the dedicated search engine. Write operation to CRA-DRAM is similar to read operation. Where the write data are written to is determined by the write address, which would also be searched in the CAM array. The only challenge is that once matched the write latency to RAM cannot be exceed the normal DRAM write latency. The challenges for read and write operation will be further discussed later.

2.2 CAM design for CRA-DRAM

Fig. 3. CAM Design for CRA-DRAM.

As the specific application in CRA-DRAM, the CAM design should be modified. According to latest JEDEC DDR4 standard, the address for DRAM access is composed by bank group address, bank address, row address and column address. With an ACT (activate) command, an entire row of data (called one page) is moved from the array of DRAM cells into the sense amplifiers (SAs). The time it takes is known as t_{RCD} . Then subsequent column access commands would move column data from SAs to IO data bus through the I/O gating logic. So a multi-segment search (MSS) scheme for CAM design is proposed to reduce search overhead, as

shown in Fig. 3. When an ACT command is issued to move one row data to SAs, the first segment search with row address (including address of bank group, bank, and row) is also performed. Only if the first MLs are matched, the corresponding second MLs would be precharged preparing for the second segment search along with following read or write operations with specific column addresses. If an active row has no leaky cells, there would be no second segment search for no matched results in the first segment search. Hence much power will be saved by MSS for leaky cells just account for a tiny part. For a 32 Gb ($\times 8$) DRAM device, the total number of row address including bank address and bank group address is 22 bits. The number of column address is always 10 bits. So the total bit width of a CAM search word including the two segments would be 32 bits. The size of CAM entries and RAM is determined by the distribution of leaky cells.

2.3 Timing challenges to CRA-DRAM

Fig. 4. A read cycle and a write cycle for DRAM devices.

As explained above, CAM search result determines where the data are read from or write to, which may have a negative impact on performance or violate the DDR4 protocol. Below we will prove this concern would not happen based on specific timing analysis. First, let's get to know some crucial timing parameters during DRAM access procedure. Here we ignore the additive latency (AL) which is supported to make command bus and data bus efficient for sustainable bandwidths. As shown in Fig. 4, it takes a t_{RCD} time to bring a row of data from the DRAM array to the SAs. Then, a column read command places the requested data on the data bus, which costs a CAS latency of t_{CAS} or t_{CL} . In the case of a write operation, data are provided by the memory controller, driven through the data bus, passed by the I/O gating multiplexors, overdrive the SAs, and finally stored into the DRAM cells. t_{CWL} is the column write latency between the internal write command and the availability of the first bit of input data on the data bus. The write recovery time, t_{WR} , is the minimum time interval between the end of a write data burst and the start of a precharge command, allowing SAs to restore data to a DRAM row. So from the point of the memory controller, the minimum time for a read or a write operation is determined by t_{CL} and t_{CWL} , respectively. From Micron 8 Gb SDRAM product manuals [6], t_{CL} and t_{CWL} are both a dozen of clock cycles, over 10 ns.

Next, we get to know the search delay (t_{SD}) of CAM and access delay (t_{AD}) of RAM. CAM can be divided into two types, volatile CAMs or non-volatile CAMs (nvCAMs). From recently published high-quality papers with silicon test results, we have made a summary in Table I. As we can see that all the proposed CAMs or nvCAMs have a search time of few nanosecond or sub nanosecond. The fastest search speed can reach 1.25 G sps (search per second). And the search data width can reach up to 144 bits, which is more than enough to meet our requirement in

CRA-DRAM. Once an address is matched in CAM, it will take another t_{AD} time to access the RAM (also known as SRAM). As we know, SRAM or non-volatile SRAM (nvSRAM) is a kind of ultra fast memory that operates about at GHz. For example, the 1 Mb nvSRAM proposed in [11] can operate in 1.5 ns/2.1 ns random read/write cycle. Thus, for worst-case consideration, the worst delay is the sum of t_{SD} and t_{AD} , which still has much margin compared with DRAM access latency. The following condition is satisfied:

$$t_{SD} + t_{AD} < t_{CL} \text{ or } t_{CWL} \quad (1)$$

So the timing challenges to CRA-DRAM need not be considered completely. The memory controller gets no sense of the difference between conventional DRAM system and CRA-DRAM system, meaning that CRA-DRAM can immediately run on existing systems without any modifications to the memory controller or upper levels.

Table I. Summary of recently published volatile and non-volatile CAMs

	[5] EPLC 2013 JSSC	[7] ABC 2014 JSSC	[8] Layout 2015 VLSIC	[9] 3T1R 2015 ISSCC	[10] 2.5T1R 2016 ISSCC
Configuration	2 K × 640	128 × 128	128 × 80	2 × 64 × 64	64 × 256
Process	32 nm SOI	65 nm CMOS	16 nm FinFET	90 nm CMOS	65 nm CMOS
Supply	0.95 V	1.2 V	0.8 V	1 V	1 V
Search speed	1 ns	1.07 ns	1.25 G sps	0.96 ns	1 ns
Energy (fJ/bit/search)	0.58	0.77	N.A.	0.51	0.28–0.71
Standby power (μW/Kb)	N.A. volatile	~0.375 volatile	~4.8 volatile	0 non-volatile	0 non-volatile

2.4 Energy challenges to CRA-DRAM

The additional energy consumption caused by CAM/RAM is inevitable in our memory system. The challenge is whether this part is considerable to the reduced refresh power of DRAM devices. The total energy of CAM/RAM ($E_{CAM/RAM}$) can be calculated as below:

$$E_{CAM/RAM} = E_{S_CAM} + E_{bg_CAM} + E_{A_RAM} + E_{bg_RAM} \quad (2)$$

where E_{S_CAM} , E_{bg_CAM} , E_{A_RAM} and E_{bg_RAM} denote the search energy of CAM, the background energy of CAM, the access energy of RAM and the background energy of RAM, respectively. As the search energy per bit per search (E_{FOM}) has been given in Table II and III, we can calculate E_{S_CAM} as below:

$$E_{S_CAM} = E_{FOM} \times N_{bits} \times N_{access} \quad (3)$$

where N_{bits} means the total number of bits in CAM array and N_{access} represents the access number to the main memory system i.e., the number for address search operations. From our simulation results, the maximum access number to our CRA-DRAM system in fixed execution cycles is at the scale of 10^8 . According to the

16Kb CAM proposed in [7], we can estimate that E_{S_CAM} is about 1.26 mJ. Assuming that the RAM energy per access is E_{unit} , E_A can be expressed as below:

$$E_{A_RAM} = E_{unit} \times N_{match} \quad (4)$$

where N_{match} indicates the total number of match results from CAM, namely, the total times for RAM access. As the leaky cells only accounts for a small proportion in the total DRAM cells, N_{match} must be much smaller than N_{access} , even less than 1%. From the results of [12] that 256 Kb SRAM operates up to a speed of 850 MHz while only consuming 4.3 pJ/access for a word length of 32 bits. So it is estimated that E_{A_RAM} is only about 4.3 μ J with 1% match probability.

The background power or the standby power of CAM and RAM is about 0.375 and 0.1 μ W/Kb from the results of [7] and [12], which can be negligible compared to the power of massive DRAM devices. So E_{bg_CAM} and E_{bg_RAM} can be ignored.

Next, we evaluate the refresh energy to draw a comparison. The equation below from [13] is used to calculate the energy caused by each refresh command:

$$E_{REF} = (IDD5 - IDD3N) \times t_{RFC} \times VDD \quad (5)$$

where t_{RFC} represents the time taken for one refresh operation. We use the timing and IDD current values based on [6]. The values are as follows: $IDD5 = 225$ mA, $IDD3N = 55$ mA, $t_{RFC} = 640$ ns (for 32 Gb DRAM device assumed by [4]), $VDD = 1.2$ V. The energy consumed by one refresh operation is 130.6 nJ. At a fixed execution cycles (t_{Fixed}) for simulation, the total refresh energy (E_{total}) is given as below:

$$E_{total} = E_{REF} \times \frac{t_{Fixed}}{t_{REFI}} \quad (6)$$

where t_{REFI} is the average refresh interval between two refresh commands. In normal temperature range, the refresh period (t_{RP}) is 64 ms and the default refresh rate is to carry out fixed 8192 refresh commands in each bank, so t_{REFI} is equal to 7.8 μ s. Therefore, the total refresh energy in the fixed execution cycles is about 34.3 mJ in normal temperature range and doubled to 68.6 mJ in extended temperature range. If we increase the t_{REFI} by one time, we can save the refresh energy by half of E_{total} . Compared with the extra CAM/RAM energy, we have the conclusion that our CRA-DRAM can indeed reduce the total power of the memory system. It also should be noted that the MSS scheme can significantly reduce search overhead. In addition, we can see that the main energy in $E_{CAM/RAM}$ is the CAM search energy. The RAM access energy is so small that the match probability almost has no effect on E_{total} . Thus in the following evaluation the match probability is set as 1%.

3 Evaluation methodology

To evaluate our CRA-DRAM, we use an standard X86 platform provided by gem5 simulator [14] with a cycle-accurate DRAM timing model named DRAMSim2 [15]. Benchmarks are drawn from SPEC CPU2006. The system configuration is listed in Table II. Each simulation is run for fixed 2.048 billion cycles, since refresh timing is based on wall time [1]. As CRA-DRAM is a high-performing mechanism that executes more instructions and therefore generates more memory accesses, we

report DRAM system power as energy per access to achieve a fair comparison. The DRAM device capacity we simulated is set as 32 Gb with 8-bit IO width, and the timing parameters are referred to the datasheet from [6]. The default value of t_{RFC}/t_{REFI} is set to 640 ns/7800 ns at normal temperature range.

Table II. Evaluation system configuration

Component	Specifications
Processor	timing-simple X86 CPU, 4 GHz
L1-Icache & L1-Dcache	32 KB, 2-way associative, 64 B block size
L2-cache	256 KB, 8-way associative, 64 B block size
L3-cache	2 MB, 16-way associative, 64 B block size

Table III. CAM/DRAM configuration for evaluation

t_{RP}/t_{REFI}	Failure IO data	CAM size	SRAM size
128 ms/15.6 μ s	30	1 Kb	0.25 Kb
256 ms/31.2 μ s	1000	32 Kb	8 Kb
384 ms/46.8 μ s	6500	200 Kb	50 Kb
512 ms/64.2 μ s	30000	1000 Kb	250 Kb

We also include the energy consumption of CAM/DRAM in our evaluation based on the result of [7] and [12]. Another concern is the sizes of the CAM/DRAM. As the refresh period increases, more leaky cells need to be replaced by RAM, resulting in larger CAM entries and RAM array. For evaluation, we have increased the refresh period to 128 ms (2 $\times$), 256 ms (4 $\times$), 384 ms (6 $\times$) and 512 ms (8 $\times$). The number of failure IO data in DRAM devices are given in Fig. 1. For worst case consideration that the leaky cells are all located on different IO data, the required sizes of CAM and RAM are listed in Table III.

4 Simulation results

The baseline is the conventional DRAM system with 64 ms refresh period at normal temperature range. We will compare the simulation results of CRA-DRAM with the baseline and RAIDR [1] in the following aspects.

Fig. 5. (a) Number of refreshes and (b) MPKI of last level cache.

4.1 Auto-refresh reduction

As introduced above, CRA-DRAM is completely compatible with auto-refresh mechanism while RAIDR only supported row-granularity refreshes. The number of refreshes performed by RAIDR and this work under normal temperature range is given in Fig. 5(a). A mechanism that issues refresh commands every 31.2 μ s ($4\times$) instead of every 7.8 μ s would reduce refreshes by 75%, just as CRA-DRAM does, while RAIDR only provides a 74.6% refresh reduction because there are still rows to be refreshed every 64 ms. And the difference would be more obvious if further increasing the refresh period. When the refresh period is set to 384 ms ($6\times$), CRA-DRAM can cut down the refreshes by 87.5%. According to the last level cache misses per 1000 instructions (MPKI) [1], the benchmarks are divided into two classes, memory-intensive (MPKI > 5) and non-memory-intensive (MPKI < 5), as shown in Fig. 5(b). This information would be used for following analysis.

4.2 Energy reduction

The energy reduction is shown in Fig. 6. As t_{REFI} is increased by $2\times$, $4\times$, $6\times$ and $8\times$ in CRA-DRAM, the energy consumption cost by refreshes in fixed cycles is significantly decreased. Specifically, when t_{REFI} is increased to 46.8 μ s and 62.4 μ s, the average share of refresh energy in total energy consumption is dropped from 29.2% to 6.79% and 5.16%, respectively. When the refresh period is increased by $6\times$ and $8\times$, CRA-DRAM has an average energy reduction of 29.1% and 29.9%, respectively. Compared with RAIDR, CRA-DRAM has a great advantage over energy reduction. It should be noted that when we increased t_{REFI} by $8\times$, the potential for energy reduction is very limited compared with $6\times t_{REFI}$, as seen from Fig. 6. When increasing t_{REFI} from $6\times$ to $8\times$, the total energy of memory-intensive benchmarks, like *lbm*, would be risen for the increasing energy of CAM/RAM exceeds the reduced DRAM refresh energy. For non-memory-intensive benchmarks, the total energy is still trending downwards because these benchmarks have less times of memory access, which is proportional to the energy of CAM/RAM. Balanced the energy reduction and the cost for CAM/RAM, which t_{REFI} is better for configuration depends on the system and the applications.

Fig. 6. Normalized energy per access.

4.3 Performance improvement

As the refresh interval is increased by several times, much refresh cycle time (t_{RFC}) is saved for performance improvement. The normalized IPC of CRA-DRAM has been illustrated in Fig. 7. We can see that some benchmarks have a great growth of IPC with increasing refresh interval while the other benchmarks have a limited

growth. This can be explained by the memory access intensity of benchmarks. By contrasting Fig. 5(b) and Fig. 7, we can learn that the non-memory-intensive benchmarks always have a small growth because these benchmarks are cache-heavy and have less access requirement for the DRAM system. In other words, the DRAM system is waiting for access mostly and thus there is little space for improvement. But for memory-intensive benchmarks, memory access becomes a bottleneck so that CRA-DRAM plays an effective role to save refresh time for memory access. When the refresh period is extended by 4 $\times$, 6 $\times$ and 8 $\times$, CRA-DRAM consistently provides a significant performance gain over conventional 1 $\times$ auto refresh, averaging a 7.5%, 8.3% and 8.7% improvement, respectively, which is much better than RAIDR (4.1%). We can also infer that CRA-DRAM can achieve a greater power and performance improvement under extended temperature range for the refresh overhead is twice as it in normal temperature range.

Fig. 7. Normalized IPC

4.4 Applicability to DIMM, HBM and HMC

In this section, we will briefly discuss the applicability of CRA-DRAM to DIMM, High Bandwidth Memory (HBM) and Hybrid Memory Cube (HMC). DIMM is the most common type of computer memory module which is made up of a small circuit board that holds several DRAM devices. HBM is a stacked DRAM approach that promises a major boost in bandwidth (performance), along with a reduction in circuit board area. Each DRAM die is interconnected to the one below it and ultimately to the base logic die with Through-Silicon Vias (TSVs) and micro-bumps (μ Bumps). HMC leverages a 3D array of chips connected by TSVs with a logic controller embedded into the wafer. The cube is then attached directly to the CPU in a “Short Reach” configuration. While these 3D stacking DRAMs get a huge jump in throughput and bandwidth, they have not yet resolved refresh overhead. As the operating temperature of 3D stacking DRAMs will be over 90 $^{\circ}$ C, leading to reduced refresh interval, which will further aggravate the decline caused by refresh. Thus, refresh in HBM or HMC is likely to be a big issue. CRA-DRAM is applicable to DIMM, HBM and HMC.

As DRAM devices are mainly designed for cost (density), the addition of CAM/RAM can be a standalone chip on the DIMM board, as shown in Fig. 8(a). A non-volatile memory (NVM), such as phase change memory (PCM), may be required to store the profiling data. Once power on, the first thing is to transfer the profiling information to CAM cells. Another benefit of the NVM is that it can

replace the on-board EEPROM, which is always required to store some timing parameters, manufacturer, serial number and other useful information about the module. Moreover, if the NVM has sufficient capacity, it can also serve as the backup storage to build NVDIMM. For nvCAM/nvRAM, which can be also fabricated with standard CMOS process, the profiling data would be just stored in local nvCAM. In HBM and HMC, small-capacity CAM/RAM and control logic can be embedded on the large logic die area as a macro IP, not a standalone chip, as shown in Fig. 8(b). A certain automatic build-in self-test (BIST) module can be used to do on-line test periodically for DRAM retention time profiling in order to save the testing cost. In summary, we believe that CRA-DRAM is quite promising to be applied in DIMM, HBM and HMC for power reduction and performance improvement.

Fig. 8. The implementation of CRA-DRAM in DIMM, HBM and HMC.

5 Conclusion

We present a CAM-based Retention-Aware DRAM (CRA-DRAM) system to reduce the refresh energy with performance improved a lot at the same time. CRA-DRAM uses a small-capacity CAM/RAM to replace the IO-granularity data that contain leaky cells, and then increases the overall refresh interval of DRAM. To our knowledge, CRA-DRAM is the first work to propose such a hardware solution based on DRAM retention time without modification to OS, memory controller, or JEDEC standard. Our experimental evaluation shows that CRA-DRAM is effective and estimable in improving the DRAM power and performance and it is quite tolerant from variability in temperature and in DRAM cell retention time. The simplicity and flexibility of CRA-DRAM make it potentially applicable to future main memory system.

Acknowledgments

This work is supported by National Key Basic Research Program of China (2013CBA01900, 2010CB934300, 2011CBA00607, 2011CB932804, 2013CBA01904), the “Strategic Priority Research Program” of the Chinese Academy of Sciences (XDA09020402).