

DRBAC: Dynamic Row Buffer Access Control for Power and Performance of DRAM Systems

Dong-Ik Jeon and Ki-Seok Chung

Abstract—Performance of dynamic random access memory (DRAM) has been steadily improved to overcome the concern that the DRAM access time may become the performance bottleneck of a system. Besides, DRAM power consumption has become a critical issue in mobile and server systems. The open page policy is widely used to minimize the memory access latency and the power consumption of the activate and the precharge commands. In this paper, we analyze DRAM power and performance according to memory request characteristics of applications. Especially, we observe that the row buffer access control influences the overall performance and power consumption. Further, the power-delay product (PDP) is sensitive to the row buffer hit ratio and the memory request frequency. Thus, we propose a method called dynamic row buffer access control (DRBAC) that changes the row buffer access limit dynamically based on the memory request characteristics. From simulation results, it is verified that DRBAC reduces the PDP value by up to 17.8% compared to the conventional method for various benchmarks. Therefore, we conclude that the proposed DRBAC is very effective for low power and high performance DRAM systems.

Index Terms—DRAM, memory controller, memory scheduling, energy-aware system

I. INTRODUCTION

Today, dynamic random access memory (DRAM) has been widely employed as the main memory in various digital systems such as mobile devices and servers. Over the past few decades, DRAM has been developed toward capacity increase as the required size of the main memory has gone up. In contrast, limited bandwidth of DRAM has become one of the most critical issues due to both massive amount of data transfer and heterogeneous data access patterns induced by multiple types of processing units. Besides, DRAM has become a major source of power consumption because DRAMs dissipate a considerable amount of background power in order to retain its data [1, 2]. C. Lefurgy, et al. estimated that DRAMs consumed up to 40% of the total commercial server system energy [3]. In addition, A. Carroll, et al. estimated that DRAM power consumption in a smartphone occupied with a fraction ranged from 5% to 30% of the overall system power consumption [4]. Therefore, power and performance optimization for the DRAM memory controller is crucial. Among lots of techniques to improve the DRAM performance, the open page policy with a row buffer has been most widely used. In the open page policy, the row buffer keeps the activated row open until the address of the next request is known while intentionally delaying the close (precharge) operation. If consecutive memory requests make accesses to the same row data, the memory access latency can be significantly reduced. In addition, if the memory controller employs the first ready-first come first serve (FR-FCFS) scheduling which prioritizes memory requests that hit in the activated row buffer over other requests, DRAM performance can be significantly

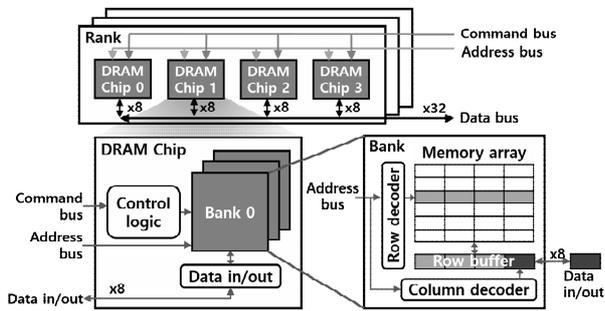


Fig. 1. DRAM-based main memory structure.

improved [5]. However, activating a row buffer constantly not only consumes additional power but also incurs some risk to have memory request starvation which is a situation that some commands are not able to be issued for a long time. Thus, the memory controller ensures that the number of consecutive memory requests that access the same row buffer should be limited.

To observe how much the row buffer access control will influence the DRAM performance and power consumption, extensive simulations are carried out and the results are analyzed in terms of the power-delay product (PDP). As a result, different PDP changing patterns according to memory request characteristics have been observed. Therefore, we propose a novel method called dynamic row buffer access control (DRBAC) which sets the maximum count of consecutive row buffer accesses dynamically to achieve the best trade-off between power consumption and performance. Simulation results under various experimental conditions verify that the proposed method is very effective.

II. OPEN PAGE POLICY

Fig. 1 shows a DRAM-based main memory structure, which is hierarchically composed of ranks, DRAM chips, and banks. Row and column addresses of memory requests share the address bus in order to minimize the number of DRAM in/out pins as shown in the figure. DRAM operates by issuing three commands sequentially: the activate command, fetching row data corresponding to the decoded row address from a cell array to the row buffer; the read/write command, exchanging data with an external device by choosing data in the row buffer for the column address; the precharge command, restoring the row data from the row

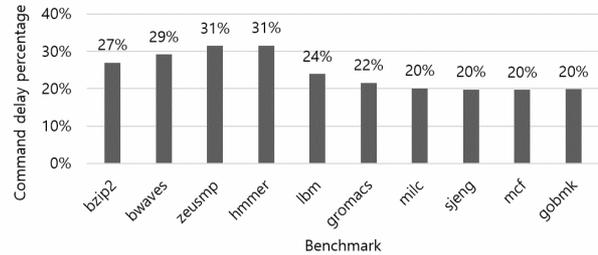


Fig. 2. The probability of commands with row buffer miss to be delayed due to the FR-FCFS scheduling in SPEC CPU2006 benchmarks.

buffer to the cell array and charging the bit lines to the reference voltage. In order to reduce the execution latency of these commands, the open page policy and the FR-FCFS scheduling are commonly used. These methods allow to skip the activate and the precharge commands per memory request by retaining the row buffer data, if the row address of the following memory request remains unchanged. However, execution of other commands may be significantly delayed by the FR-FCFS scheduling until execution of all the commands with row buffer hits are completed, which causes system performance degradation due to the memory request starvation. Fig. 2 shows how often commands with row buffer misses are delayed by the FR-FCFS scheduling under the real memory system environment with SPEC CPU2006 benchmarks. As shown in the figure, the probability of the delayed execution is about 24%, which means that some delay happens once in every four executions. Therefore, the memory controller should limit the maximum count of consecutive accesses to the same row buffer to prevent the starvation to ensure fairness in memory requests.

III. RELATED WORK

The open page policy is effective in the situation where the next memory request is highly likely to have a row buffer hit, that is, high spatial locality. On the other hand, the close page policy works better in the opposite situation. Thus, many studies have attempted to find the best trade-off by utilizing a memory access history table or a predictor. Park and Xu [6, 7] proposed a page policy control based on the saturated up/down counter prediction with reference to the memory access history. A similar approach was used in the access based

predictor (ABP)-based row-buffer closure policy [8], where the total number of memory accesses with row buffer hits was stored in a history table. When to close the row buffer was predicted by this history table. Stankovic et al. [9] proposed a close-page predictor to predict the row buffer closing time by a zero live time predictor and a dead time predictor. Xie et al. [10] proposed an application-aware page policy (AAPP) which profiled application characteristics, and then used these characteristics to make a decision on a proper policy. All of these proposals have concentrated on the prediction about the policy conversion between the open page policy and the close page policy. Since the proposed DRBAC attempts to find the best trade-off between power consumption and performance, the proposed DRBAC is orthogonal to other approaches. And therefore, DRBAC can be employed along with the aforementioned existing methods.

On the other hand, while the average memory access latency and the power consumption are reduced by the open page policy because execution of some commands are skipped with row buffer hits, the average memory throughput will be improved by employing bank interleaving. In the bank interleaving, consecutive memory reads and writes are allowed to access different memory banks in turn. As a result, memory requests are processed through multiple banks in parallel at the same time. Hence, the throughput is significantly improved. In many commercial memory systems, both the open page policy and the bank interleaving are employed to improve the system performance and the power consumption. However, in this paper, our goal is to evaluate the net effect due to the proposed DRBAC when we employ first-ready first-come first-service (FR-FCFS) scheduling to maximize the open page policy efficiency. Thus, the performance improvement due to bank interleaving is not taken into account. However, the proposed dynamic access control method can be used with bank interleaving together.

IV. DRBAC: DYNAMIC ROW BUFFER ACCESS CONTROL

Although the open page policy reduces power consumption as well as latency by skipping the activate and the precharge commands, DRAM suffers from

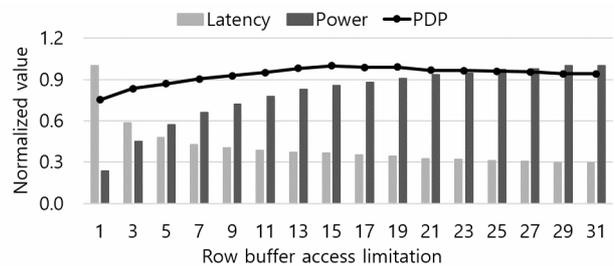


Fig. 3. Results of the row buffer access control for *mcf* benchmark.

significant power consumption due to the structure of the row buffer separated from the cell array. According to a DDR3 SDRAM datasheet (MT41J256M8) from Micron Technology Inc., the amount of the background energy consumption per one clock cycle is 23 mJ for a deactivated row buffer, whereas that for an activated row buffer is 33 mJ, which is bigger by 43% [11]. Thus, the memory controller deactivates the row buffer to avoid the unnecessary energy consumption when the command queue does not have any command with the row buffer hit or the count of row buffer accesses is over the limit. This deactivation reduces not only the row buffer power consumption but also the precharge command overhead that takes to close a row buffer before issuing a new command.

It is well-known that power consumption and performance of DRAM memory systems depend on memory request characteristics of applications. For example, in the case of applications with a lot of memory requests per unit time, the growing rate of DRAM power consumption is very steep. This is because the duration of row buffer activation increases rapidly with a lot of commands in a command queue. On the other hand, in the case of applications with relatively few memory requests, power consumption overhead is small because of automatic deactivation of the row buffer. In order to verify this observation under a real memory system environment, we have conducted experiments using benchmarks with various memory request frequencies in the SPEC CPU2006.

Fig. 3 shows experimental results of *mcf* benchmark that has a relatively high memory request frequency among SEPC CPU2006 benchmarks. The values in Fig. 3 are normalized by the maximum value to observe the net change in power consumption, latency, and PDP, respectively. As shown in the figure, the bigger the

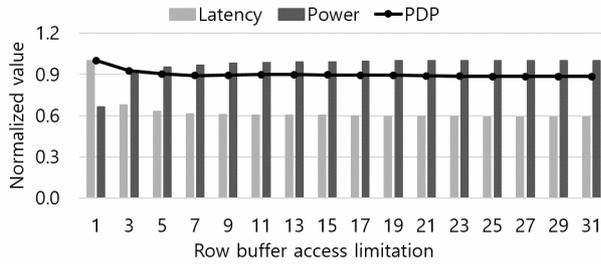


Fig. 4. Results of the row buffer access control for *hammer* benchmark.

maximum count of the row buffer access gets, the smaller the average latency becomes because the activate and the precharge commands are skipped. On the other hand, power consumption increases due to long duration of row buffer activation. The PDP value is small when the maximum count of row buffer accesses is small. This result strongly implies that the increase in power consumption overshadows the performance improvement of the open page policy for a benchmark like *mcf* which has a high memory request frequency.

Fig. 4 shows the result of the *hammer* benchmark that has a relatively low memory request frequency under the same experimental environment for *mcf*. Similar to *mcf* results, latency decreases and power consumption increases, as the maximum count of row buffer accesses gets bigger. However, the PDP variation curve shows the opposite result to the *mcf* benchmark. The reason why the opposite tendency is revealed is mainly due to automatic deactivation of the row buffer with the low memory request frequency. Furthermore, the row buffer hit ratio, which is the probability that the row address of a new request coincides with the one for data in the activated row buffer, also influences this tendency. Because memory requests with a high row buffer hit ratio make the row buffer stay in activation, the power consumption is larger than that for requests with low row buffer hit ratios. For example, two consecutive memory requests with row buffer misses consume 0.483 W, while row buffer hit requests consume 0.497 W according to a DDR3 SDRAM datasheet from Micron Technology Inc. [11]. Accordingly, reducing power consumption by decreasing the maximum count of row buffer accesses is more effective when the row buffer hit ratio is high. From these results, we claim that variation of power consumption and performance is dependent on the row buffer activation frequency. As a result, it is necessary to

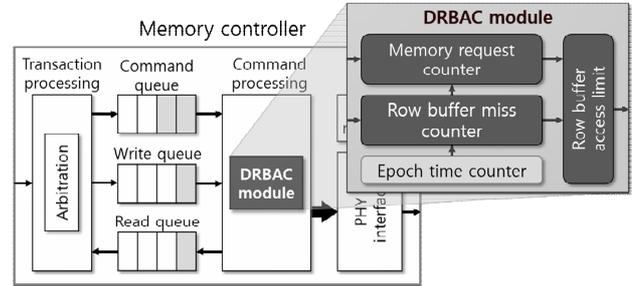


Fig. 5. The memory controller structure for DRBAC.

make a decision on the maximum count of row buffer accesses (row buffer access limit) according to memory request characteristics for a power-critical DRAM system.

Therefore, in this paper, we propose a dynamic row buffer access control method called DRBAC. Fig. 5 shows the structure of a memory controller which employs the proposed DRBAC. One memory request (transaction) is converted into three DRAM commands, such as activate, read/write, and precharge. The converted commands are rescheduled by an arbitration scheme for a better performance, and then stored in the command queue. Command issues should be carried out by obeying the timing parameters, which commonly indicate the minimum time intervals between successive command issues, in order to guarantee stable execution of the command. The command processing determines the time when the command is issued to DRAM. The DRBAC module is added in the command processing, because the module counts the number of row buffer misses as well as memory requests by monitoring the issued command. Algorithm 1 shows the pseudo code of the algorithm to determine the row buffer access limit in DRBAC. In the case of row buffer misses, the memory controller has to issue the precharge and activate commands to get a newly requested data. As a result, the number of row buffer misses will correspond to the number of the activate or precharge commands. In every epoch, DRBAC determines the row buffer access limit based on the memory request frequency and the row buffer hit ratio, as shown in line 7 to line 22. Since every memory request must have one read or write command, the memory request frequency is derived by the number of read/write commands per epoch. The row buffer hit ratio is also derived by the memory request and the row buffer miss counter. After the row buffer access limit is

Algorithm 1. Pseudo code of the DRBAC algorithm

```

1: command_queue.issue ( )
2: ...
3: if issued_cmd == activate
4:   buffer_miss ++
5: else if issued_cmd == read or write
6:   mem_req ++
7: if (current_cycle % epoch) == 0
8:   if (mem_req / epoch) >= 0.05
9:     buffer_hit = (mem_req - buffer_miss) / mem_req;
10:    switch (buffer_hit)
11:      case: >= 80
12:        buffer_limit = 1; break;
13:      case: >= 60
14:        buffer_limit = 9; break;
15:      case: >= 40
16:        buffer_limit = 16; break;
17:      case: >= 20
18:        buffer_limit = 23; break;
19:      default:
20: buffer_limit = 31;
21:   else
22:     buffer_limit = 31;

```

Table 1. The row buffer access control rule

Memory request characteristics		Row buffer access limit
0.05 ≤ memory request frequency	80 ≤ row buffer hit	1
	60 ≤ row buffer hit < 80	9
	40 ≤ row buffer hit < 60	16
	20 ≤ row buffer hit < 40	23
	row buffer hit < 20	31
memory request frequency < 0.05		31

updated, all counters and values are reset for the next epoch. The rule of row buffer access limit DRBAC is summarized in Table 1. The row buffer access limit is determined by extensive performance evaluation with respect to various applications.

V. EXPERIMENTS

1. Experimental Environment

In order to verify effectiveness of the proposed method, we implement a simulator which combines the SimpleScalar tool set [12] with the DRAMSim2 [13] simulator to conduct experiments with commercially available memory systems. We have compiled the SPEC

Table 2. Simulation environment

Parameter	Value
Processor	2-GHz, out-of-order, Alpha ISA, fetch / decode / issue 4 / 4 / 4, LSQ 8-entry, RUU 16-entry
Last level cache (unified)	256KB, LRU, 4-way associative, 1024sets, block size : 64Byte
DRAM device	Micron DDR3-1333 (MT41J256M8 revision M) 2Gb: 256 Meg x 8 Configuration
Row buffer policy	Open page, FR-FCFS [5]
Address mapping	channel:row:rank:bank:column
Burst length	8
Transaction/Command queue	32-entry, rank queuing structure
Power-down mode	0 cycle threshold, tXP cycle penalty

CPU2006 benchmark suite for Alpha instruction set architecture (ISA) to run simulations and execute the benchmark suite until 500 million instructions are completely executed. The user-defined epoch is set to 100,000 DRAM clock cycles. In order to maximize memory performance, we set the memory burst length to 8, which is the last level cache block size. Detailed simulation environment is summarized in Table 2.

2. Experimental Results

Since the proposed DRBAC sets the maximum count of row buffer accesses dynamically, it is possible to flexibly control memory accesses under various execution conditions. In order to verify the effectiveness, we analyze memory request characteristics using various SPEC CPU2006 benchmarks.

Fig. 6 shows experimental results of SPEC CPU2006 memory request characteristics. The results are depicted in terms of two factors: row buffer hit and memory request frequency. The row buffer hit represents efficiency of the open page policy. The memory request frequency is measured by the number of memory requests divided by the number of executed DRAM clock cycles, and it shows how often the row buffer is activated. As shown in Fig. 6, all benchmarks show unique memory request characteristics. Among them, four distinctive benchmarks were chosen to conduct experiments: *bzip2*, *hmmmer*, *sjeng*, and *gobmk*.

Fig. 7 shows simulation results of the PDP comparison of DRBAC with a conventional static access control method with respect to various row buffer access controls

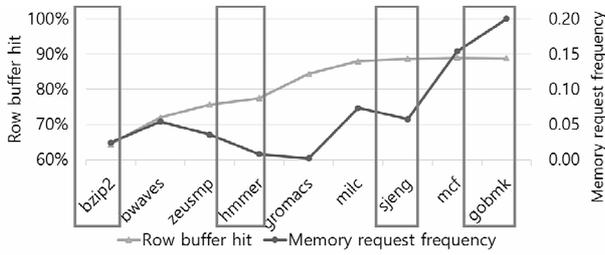


Fig. 6. Row buffer hit ratio and memory requests frequency of SPEC CPU2006 benchmarks.

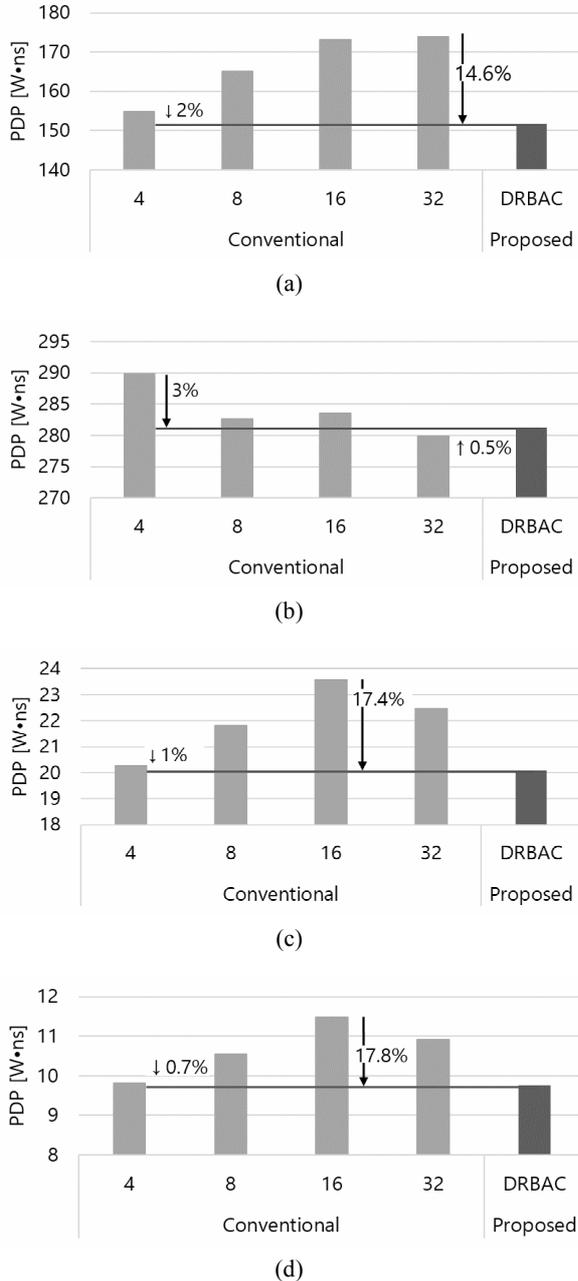


Fig. 7. The PDP comparison of DRBAC with row buffer access control (a) *bzip2*, (b) *hmmer*, (c) *sjeng*, (d) *gobmk* benchmark.

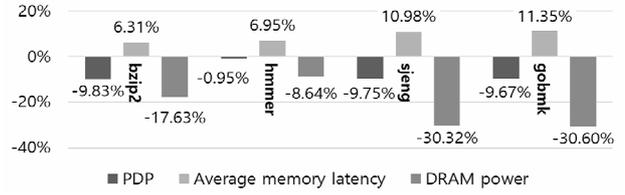


Fig. 8. The average rate of DRBAC PDP, latency and power changes with respect to conventional methods respectively in benchmarks.

for *bzip2*, *hmmer*, *sjeng* and *gobmk*, respectively. In the conventional memory controller, the maximum count of row buffer accesses is statically fixed. The fixed count is determined based on several system characteristics. Thus, we compare DRBAC with four different static methods with 4, 8, 16, and 32 as the maximum count. As shown in the figure, the DRBAC PDP value is less than those of conventional methods by up to 17.8%, but some conventional methods show almost the same result as the DRBAC PDP value, especially the method with the maximum count of 32 for *hmmer* benchmark. However, the best maximum count with the lowest PDP value is different for each benchmark. That is, although a static method with a certain maximum count number may outperform DRBAC for some benchmarks, it works poorly for the others. In contrast, DRBAC has shown consistently good results for all benchmarks, because DRBAC sets an appropriate row buffer access control adaptively based on memory request characteristics.

Fig. 8 shows the memory latency overhead and the power reduction of DRBAC on average in respective benchmarks. The result shows that the DRBAC average memory latency is increased by up to 11.35% compared with conventional methods, but this only takes DRAM performance into consideration. The overall performance overhead due to DRBAC is reduced by more than half of those results from the system viewpoint. That is, the performance overhead is relatively small. Therefore, it can be claimed that DRBAC is a very effective method to get the best DRAM trade-off for power and performance.

VI. CONCLUSION

The DRAM-based main memory has often become a performance bottleneck due to massive data processing and heterogeneous memory request patterns in high

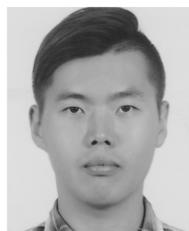
performance systems. In addition, DRAM dissipates large amount of power, and especially, the amount of background power consumption is significant. Therefore, we proposed a novel scheme called dynamic row buffer access control (DRBAC) to find the best trade-off between power consumption and performance. DRBAC dynamically adjusts the row buffer access control to take advantage of application-specific memory request characteristics to find the best trade-off between power and performance. DRBAC module added in a memory controller determines the maximum count of row buffer accesses depending on memory request frequency and row buffer hit ratio. The simulation results show that the PDP value of DRBAC is improved by up to 17.8% compared to that of the conventional method. Therefore, DRBAC is claimed to be a highly effective low power technique in DRAM-based main memory systems.

ACKNOWLEDGMENTS

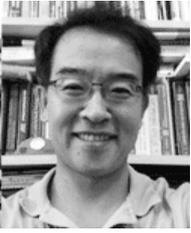
This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education(NRF-2015R1D1A1A09061079).

REFERENCES

- [1] M. Kondo and H. Nakamura, "Reducing Memory System Energy by Software-Controlled On-Chip Memory," *IEICE Transactions on Electronics*, pp. 580-588, 2003.
- [2] H. Zhu, et al, "Formal Model for the Reduction of the Dynamic Energy Consumption in Multi-Layer Memory Subsystems," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, pp. 3559-3567, 2008.
- [3] C. Lefurgy, et al, "Energy Management for Commercial Servers," *IEEE Computer*, vol. 36, pp. 39-48, Dec. 2003.
- [4] A. Carroll and G. Heiser, "An Analysis of Power Consumption in a Smartphone," *USENIX annual technical conference*, 2010.
- [5] S. Rixner, et al, "Memory Access Scheduling," in *Proceedings of International Symposium on Computer Architecture*, pp. 128-138, 2000.
- [6] S.-I. Park and I.-C. Park, "History-Based Memory Mode Prediction for Improving Memory Performance," in *Proceedings of International Symposium on Circuits and Systems*, vol. 5, pp. 185-188, 2000.
- [7] Y. Xu, A. S. Agarwal, and B. T. Davis, "Prediction in Dynamic SDRAM Controller Policies," in *Proceedings of International Workshop on Embedded Computer Systems: Architectures, Modeling, and Simulation*, pp. 128-138, 2009.
- [8] M. Awasthi, et al, "Prediction Based DRAM Row-Buffer Management in the Many-Core Era," in *Proceedings of International Conference on Parallel Architectures and Compilation Techniques*, pp. 183-184, 2011.
- [9] V. V. Stankovic and N. Z. Milenkovic, "DRAM Controller with a Close-Page Predictor," in *Proceedings of International Conference on Computer as a Tool EUROCON*, vol. 1, pp. 693-696, 2005.
- [10] M. Xie, et al, "Page Policy Control with Memory Partitioning for DRAM Performance and Power Efficiency," in *Proceedings of IEEE International Symposium on Low Power Electronics and Design*, pp. 298-303, 2013.
- [11] Micron Technology Inc, "DDR3 SDRAM (MT41J256M8) Datasheet," 2014, http://www.micron.com/~/media/documents/products/datasheet/dram/ddr3/2gb_ddr3_sdram.pdf.
- [12] T. Austin, E. Larson, and D. Ernst, "SimpleScalar: An Infrastructure for Computer System Modeling," *IEEE Computer*, pp. 59-67, 2002.
- [13] P. Rosenfeld, E. C. Balis, and B. Jacob, "DRAMSim2: A Cycle Accurate Memory System Simulator," *IEEE Computer Architecture Letters*, vol. 10, no. 1, pp. 16-19, 2011.



Dong-Ik Jeon received his B.S. in Electronics & Communication Engineering from Hanyang University, Ansan, Korea in 2012, and he is currently working toward a Ph.D. in Electronics and Computer Engineering from Hanyang University, Seoul, Korea. His research interests include the DRAM memory controller, memory architecture, and hybrid memory cube (HMC).



Ki-Seok Chung received his B.S. in Computer Engineering from Seoul National University, Seoul, Korea in 1989, and his Ph.D. in Computer Science from the University of Illinois at Urbana-Champaign in 1998. He was a Senior R&D

Engineer at Synopsys, Inc. in Mountain View, CA from 1998 to 2000, and was a Staff Engineer at Intel Corp. in Santa Clara, CA from 2000 to 2001. He also worked as an Assistant Professor at Hongik University, Seoul, Korea from 2001 to 2004. Since 2004, he has been a professor at Hanyang University, Seoul, Korea. His research interests include low power embedded system design, multi-core architecture, image processing, reconfigurable processor and DSP design, SoC-platform based verification and system software for MPSoC.