LETTER

IEICE Electronics Express, Vol.15, No.3, 1–12

3D die-stacked DRAM thermal management via task allocation and core pipeline control

Changho Yoon, Jae Hoon Shim, Byungin Moon, and Joonho Kong[a)]

School of Electronics Engineering, College of IT Engineering, Kyungpook National University, 80 Daehak-ro, Buk-gu, Daegu, Korea

a) *joonho.kong@knu.ac.kr*

Abstract: A major hurdle to adopt 3D stacked DRAM is a thermal problem particularly when the DRAM dies are stacked above the processor dies. Exacerbated thermal problems in DRAM cause another problem which increases refresh rates to ensure data integrity of DRAM cells. In this paper, we propose two efficient techniques to address the thermal problem in 3D die-stacked DRAM by suppressing adverse thermal impacts from the processor die. Our thermal-aware task mapping technique allocates tasks to cores by considering computation-intensiveness of the workloads to minimize thermal interactions. The workload-aware core pipeline control technique adjusts pipeline widths (fetch and issue widths) of processor cores considering the workload characteristics. By adopting our proposed techniques, system-wide energy consumption is reduced by 7.6% while improving performance by 0.4% on average, thanks to the reduced pipeline widths and refresh rates. In terms of temperature, our techniques reduce the number of DRAM banks which exceed 85 degree Celsius by 92.8%, on average.

Keywords: 3D stacked DRAM, temperature, refresh, task allocation, pipeline control

Classification: Integrated circuits

References

[1] JEDEC SOLID STATE TECHNOLOGY ASSOCIATION, "High bandwidth memory (HBM) DRAM" (2013).

[2] "Hybrid Memory Cube, Micron Technologies," http://www.micron.com/innovations/hmc.html.

[3] JEDEC SOLID STATE TECHNOLOGY ASSOCIATION, "Wide I/O single data rate (Wide I/O SDR)" (2011).

[4] J. Kong, *et al.*: "Recent thermal management techniques for microprocessors," ACM Comput. Surv. **44** (2012) 13 (DOI: 10.1145/2187671.2187675).

[5] I. Bhati, *et al.*: "DRAM refresh mechanisms, penalties, and trade-offs," IEEE Trans. Comput. **65** (2016) 108 (DOI: 10.1109/TC.2015.2417540).

[6] A. Jain, *et al.*: "A 1.2 GHz alpha microprocessor with 44.8 GB/s chip pin

DOI: 10.1587/elex.15.20171253
Received December 19, 2017
Accepted January 9, 2018
Publicized January 24, 2018
Copyedited February 10, 2018

bandwidth," 2001 IEEE International Solid-State Circuits Conference. Digest of Technical Papers. ISSCC (Cat. No. 01CH37177) (2001) 240 (DOI: 10.1145/2187671.2187675).
[7] M. Halpern, *et al.*: "Mobile CPU's rise to power: Quantifying the impact of generational mobile CPU design trends on performance, energy, and user satisfaction," HPCA (2016) 64 (DOI: 10.1109/HPCA.2016.7446054).
[8] J.-S. Kim, *et al.*: "A 1.2 V 12.8 GB/s 2 Gb mobile Wide-I/O DRAM with 4 × 128 I/Os using TSV based stacking," IEEE J. Solid-State Circuits **47** (2012) 107 (DOI: 10.1109/JSSC.2011.2164731).
[9] S. Li, *et al.*: "McPAT: An integrated power, area, and timing modeling framework for multicore and manycore architectures," MICRO (2009) 469 (DOI: 10.1145/1669112.1669172).
[10] J. Lin, *et al.*: "Thermal modeling and management of DRAM memory systems," ISCA (2007) (DOI: 10.1145/1250662.1250701).
[11] S. Liu, *et al.*: "Hardware/software techniques for DRAM thermal management," HPCA (2011) 515 (DOI: 10.1109/HPCA.2011.5749756).
[12] D. Zhao, *et al.*: "Temperature aware thread migration in 3D architecture with stacked DRAM," ISQED (2013) 80 (DOI: 10.1109/ISQED.2013.6523594).
[13] J. Meng, *et al.*: "Optimizing energy efficiency of 3-D multicore systems with stacked DRAM under power and thermal constraints," DAC (2012) 648 (DOI: 10.1145/2228360.2228477).
[14] W. H. Lo, *et al.*: "Thermal-aware dynamic page allocation policy by future access patterns for hybrid memory cube (HMC)," DATE (2016) 1084.
[15] D. Li, *et al.*: "Adaptive thermal management for 3D ICs with stacked DRAM caches," DAC (2017) (DOI: 10.1145/3061639.3062197).
[16] M. H. Hajkazemi, *et al.*: "Wide I/O or LPDDR? Exploration and analysis of performance, power and temperature trade-offs of emerging DRAM technologies in embedded MPSoCs," ICCD (2015) 62 (DOI: 10.1109/ICCD.2015.7357085).
[17] N. Binkert, *et al.*: "The Gem5 simulator," SIGARCH Comput. Archit. News **39** (2011) 1 (DOI: 10.1145/2024716.2024718).
[18] M. Poremba, *et al.*: "NVMain 2.0: A user-friendly memory simulator to model (non-)volatile memory systems," IEEE Comput. Archit. Lett. **14** (2015) 140 (DOI: 10.1109/LCA.2015.2402435).
[19] R. Zhang, *et al.*: "HotSpot 6.0: Validation, acceleration and extension," University of Virginia, Tech. Rep. (2015).
[20] Y. Zhu, *et al.*: "Integrated thermal analysis for processing in die-stacking memory," MEMSYS (2016) (DOI: 10.1145/2989081.2989093).

1 Introduction

3D stacked DRAM has been considered as one of the key components in a data-intensive computing domain. As popular contemporary workloads (e.g., deep learning, big data, vision processing, etc.) are demanding high data bandwidth, the conventional DRAM architecture (i.e., 2D DRAM) has become hard to deliver satisfactory data bandwidth for those workloads mainly due to narrow I/O width. It necessitates a novel memory architecture that provides higher data bandwidth. To this end, 3D DRAM architecture was introduced to deliver extremely high data bandwidth. For instance, high-bandwidth memory (HBM) [1], hybrid memory cube (HMC) [2], and Wide-I/O [3] have been introduced to realize 3D die-stacked

DOI: 10.1587/elex.15.20171253
Received December 19, 2017
Accepted January 9, 2018
Publicized January 24, 2018
Copyedited February 10, 2018

DRAM architectures. Those DRAM architectures utilize through-silicon-vias (TSVs) to implement input/output interfaces, enabling much wider I/O width.

Though 3D stacked DRAM provides performance improvement for data-intensive workloads, a thermal issue is one of the most critical problems. The die-stacked architecture of 3D DRAM memories exacerbates a thermal problem because heat dissipation is more difficult in 3D than the planar structure. Particularly when DRAM dies are stacked above the processor die, the thermal problem is much more severe as processor cores generally exhibit high power density. Though the main problem of high temperature in ICs is a reliability issue [4], DRAM cells have one more important problem under high temperature: refresh rates. The typical DRAM refresh interval is 64 ms [5] while it is reduced to 32 ms (by half) when temperature is over 85°C to ensure data integrity. Since reduced refresh interval adversely affects both performance and energy-efficiency, it is crucial to manage DRAM temperature so that it does not go beyond 85°C. Without appropriate thermal management, 3D DRAM architecture would lead to energy-inefficiency and performance loss due to the increased refresh rates.

In this paper, we propose novel techniques to manage temperature of 3D DRAM memory architecture. We apply two processor-side techniques for 3D DRAM thermal management: thermal-aware task-core mapping and workload-aware pipeline control. The former assigns tasks to the cores in a thermal-aware manner while the latter controls the pipeline width (e.g., fetch width, issue width, etc.) of underlying processor cores. To minimize performance losses due to the pipeline width reduction of the cores, we consider workload characteristics (computation-intensive vs. memory-intensive). If the running workload is memory-intensive, it aggressively reduces pipeline width since the memory-intensive workloads can sufficiently hide performance losses (long memory access latency can hide performance losses from the pipeline width reduction to some extent). On the contrary, for computation-intensive workloads, we conservatively apply the pipeline width control as their performances are sensitive to pipeline width. By applying our techniques, we reduce system energy consumption by 7.6% (up to 10.7%) with 0.4% performance improvement thanks to the reduced refresh operations. In addition, the number of DRAM banks over 85°C is reduced by 92.8%, meaning that our technique has a positive impact on DRAM lifetime reliability.

2 Preliminaries

2.1 Baseline system

Our baseline system is similar to modern high performance mobile computer system. The modeled processor has eight cores and each processor core is modeled similar to modern high performance processor core [6]. The baseline core pipeline widths are set as 8 (fetch and issue widths). The rest of the core configurations is presented in Table I. Though our modeled processor core is similar to the core widely adopted in high-performance domains, modern mobile processors have already employed high-performance techniques such as wide superscalar and out-of-order execution [7]. In terms of memory hierarchy, there is a 2 MB L2 cache which is shared across the cores between the L1 caches and main memory. For 3D

DOI: 10.1587/elex.15.20171253
Received December 19, 2017
Accepted January 9, 2018
Publicized January 24, 2018
Copyedited February 10, 2018

Table I. Processor core and L2 cache specifications.

Categories	Specifications
Branch predictor	Tournament branch predictor, 4 K BTB entries
Core pipeline	8 cores, 8-way superscalar, Out-of-order execution
D-cache	64 KB, 2-cycle latency, 2-way set-associative, 64B blocksize
I-cache	32 KB, 2-cycle latency, 2-way set-associative, 64B blocksize
L2 cache	2 MB, 20-cycle latency, 8-way set-associative, 64B blocksize

die-stacked DRAM architecture, we use Wide-I/O [3]. Fig. 1 shows the floorplans of our dies ((a)~(c)) and layer configurations ((d)). In the first floor (bottommost layer), there are eight cores and L2 cache (including uncore area) in the processor while there are Wide I/O DRAM banks and peripherals in the above two dies. The heat spreader and heat sink exist in the topmost two layers as shown in Fig. 1(d).

The floorplan of the DRAM dies is generated by referring to the die photos from [8]. The core and L2 cache areas for 22 nm technology nodes are obtained from McPAT [9]. The baseline Wide-I/O DRAM has a T_{REFI} of 3.9 us under 85°C. In our system, if any of the DRAM banks is above 85°C, T_{REFI} of all the Wide-I/O DRAM banks becomes 1.95 us (an half of the conventional refresh interval) for data integrity. For design simplicity, the identical T_{REFI} is applied to all DRAM banks. The rest of the Wide-I/O parameters are set as default parameters of the Wide-I/O specifications.

2.2 Related work

Several previous studies try to reduce temperature of 2D DRAM (DIMMs: Dual In-line Memory Modules) [10, 11]. In [10], DRAM memory system is controlled by dynamic thermal management (core gating and dynamic voltage and frequency scaling). In [11], DRAM thermal management is carried out by DRAM-aware techniques (e.g., a new cache line replacement policy, new write buffer design, and page allocation techniques, etc.). However, the above mentioned techniques are geared toward the conventional 2D DRAM systems with DIMMs.

For 3D DRAM systems, a thread migration technique [12] was proposed to reduce temperatures of processors with 3D stacked DRAMs, which includes task rotation and migrations of the running threads in the system. In [13], an optimization technique for performance improvement under power and thermal constraints is proposed to decide optimal voltage and frequency settings dynamically in runtime. In [14], an operating system-level page allocation technique to optimize both performance and temperature in hybrid memory cube (HMC) is proposed. The proposed technique in [14] utilizes analytical models to estimate performance impacts of the memory access behaviors in runtime. In [15], a thermal management technique for 3D-stacked DRAM caches is proposed. The technique proposed in [15] estimates performance impact of the reduced refresh intervals and dynamically adjust frequencies of the CPU in runtime. In [16], Wide-I/O and LPDDR3 technologies are quantitatively compared and the authors proposed a stacked LPDDR3 architecture which is a compromised one of LPDDR3 and Wide-I/O. Compared to the previous studies, our proposed technique combines both task-core

DOI: 10.1587/elex.15.20171253
Received December 19, 2017
Accepted January 9, 2018
Publicized January 24, 2018
Copyedited February 10, 2018

IEICE Electronics Express, Vol.15, No.3, 1–12

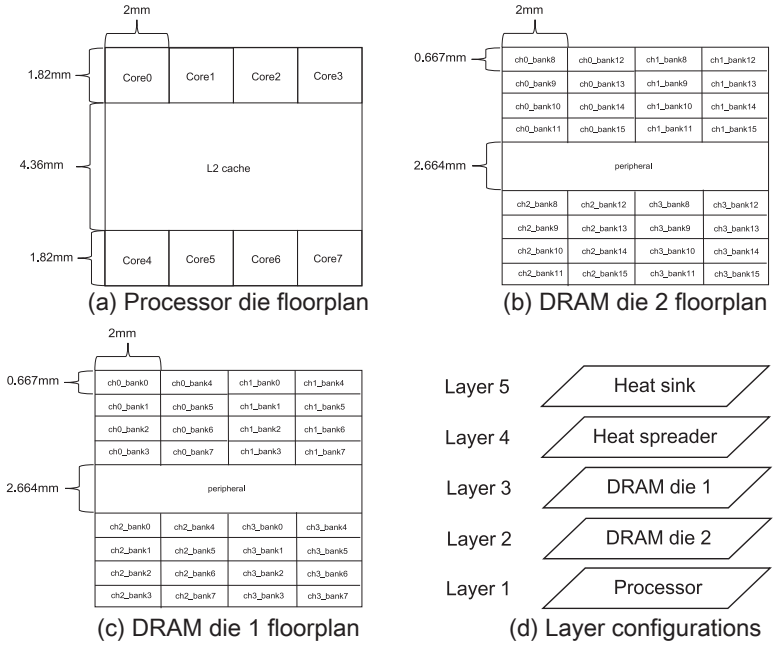


Fig. 1. Our base floorplans used in our work.

mapping and processor pipeline control for computer systems which employs 3D die-stacked DRAM.

3 Our proposed technique

3.1 Quantifying workload characteristics

In this work, we classify workloads by considering computation-intensiveness. The rationale is that memory-intensive workloads are typically cool tasks (i.e., core temperature does not increase much) while computation-intensive workloads are typically hot tasks. We utilize two metrics to gauge computation-intensiveness: workload IPC (instruction per cycle) and L2 cache misses per kilo instructions (MPKI). The IPC is often used for quantifying performance and computation-intensive workloads tend to have higher IPC while the memory-intensive workloads tend to have lower IPC. This is due to the fact that memory-intensive workloads typically have higher cache miss rates, resulting in higher number of lower-level cache and main memory accesses. This in turn translates into lower IPC of memory-intensive workloads compared to that of computation-intensive workloads. The L2 cache MPKI (in short, L2 MPKI) can also be used to represent memory access intensiveness. As the workload has higher L2 MPKI, it accesses main memory more often (i.e., memory-intensive) resulting in high load-use latency. As a result, memory-intensive workloads are often stalled by cache misses, which leads to relatively low temperature of the running core. On the contrary, computation-intensive workloads are rarely stalled due to cache misses and memory accesses, which leads to high power consumption of the core. This in turn increases temperature of the running core.

We quantify the workload characteristics as follows:

$$I_{comp} = \alpha \times IPC - \beta \times L2MPKI + \gamma \tag{1}$$

DOI: 10.1587/elex.15.20171253
Received December 19, 2017
Accepted January 9, 2018
Publicized January 24, 2018
Copyedited February 10, 2018

IEICE Electronics Express, Vol.15, No.3, 1–12

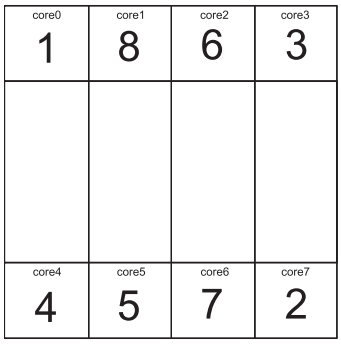


Fig. 2. The order of task assignment when we sort the I_{comp} values in the descending order.

where α, β, and γ are set to be 2, 0.01, and 1, respectively, in this work (these parameters are tunable). Please note that γ is a tuning parameter. By utilizing the I_{comp} values for each workload, we perform task-core mapping and pipeline control, which will be explained in the following subsections.

3.2 Thermal-aware task-core mapping

If a hot core has a close proximity to another hot core, those cores are likely to be even much hotter due to thermal interactions to each other. Consequently, the increased core temperature is also likely to increase DRAM temperature, which may result in shorter refresh intervals. Thus, to reduce DRAM temperature as well as processor core temperature, it is very crucial to carefully map the task to cores. In this paper, we propose a thermal-aware task-core mapping technique which takes workload characteristics into account.

For thermal-aware task-core mapping, we utilize I_{comp} of the workloads to be scheduled in the processor. Firstly, since we have eight cores in the processor, we sort eight (or less) workloads by a descending order of their I_{comp}. We then map the workloads to the cores by following a specific order (number in the cores in Fig. 2). Thus, assuming we have eight tasks to allocate, the workload with the highest I_{comp} will be allocated in Core0 (1st order) while that with the lowest I_{comp} will be allocated in Core1 (8th order). The main strategy of our task-core mapping can be summarized as follows: 1) we allocate hot tasks as far as possible so that we can minimize thermal interactions, 2) we try to allocate hot tasks to the side of the processor floorplan so that the heat generated by the cores can be dissipated well.

3.3 Workload-aware pipeline control

Along with the task-core mapping, we also propose to control the processor core pipeline width by utilizing workloads' I_{comp}. For pipeline control, we throttle fetch and issue widths of the processor cores so that the dynamic power consumption of the processor cores can be reduced. To determine the fetch and issue widths, we use the mapping rule shown in Table II. The workloads can be classified into four types according to their I_{comp} values, each of which is mapped to one of four different pipeline width configurations (1, 2, 4, and 8). As the workloads have lower I_{comp}, they tend to be memory-intensive. Thus, we try to reduce fetch and issue widths of the cores running memory-intensive workloads to reduce core temperature. The

DOI: 10.1587/elex.15.20171253
Received December 19, 2017
Accepted January 9, 2018
Publicized January 24, 2018
Copyedited February 10, 2018

IEICE Electronics Express, Vol.15, No.3, 1–12

Table II. Fetch and issue widths according to I_{comp}.

	$0 \leq I_{comp} < 1$	$1 \leq I_{comp} < 2$	$2 \leq I_{comp} < 3$	$3 \leq I_{comp}$
Width	1	2	4	8

main reason why we try to reduce pipeline width of memory-intensive workloads is to minimize performance degradation. Reducing pipeline widths of computation-intensive workloads (i.e., high I_{comp}) may result in more temperature reduction. However, it would also degrade performance of the computation-intensive workloads as their performances are much more sensitive to pipeline widths compared to those of memory-intensive workloads. In contrast, the memory-intensive workloads are less sensitive to pipeline widths as long memory latency can sufficiently hide the adverse impact of the reduced pipeline bandwidth on performance.

Per-core DVFS (dynamic voltage and frequency scaling) could be an effective solution to reduce power density in the cores. However, many mobile CPUs have not still adopted per-core DVFS since it increases design complexity (it also increases design burden in power management ICs). Instead of using per-core DVFS, we propose to use pipeline width control, which can be more widely adopted in mobile domains, because it can be implemented with very small hardware overhead.

3.4 Hardware and runtime support

Our task-core mapping and pipeline control can be performed at operating system (OS) scheduling ticks in case the list of running workloads in the processor is changed. When operating system tries to map the tasks (processes or threads) into the cores, profiled I_{comp} values for each task (i.e., workload) are loaded and OS determines the location of the tasks by using our core-task mapping technique. For the workload whose I_{comp} value has not been profiled yet, OS requires a profiling of the I_{comp} value. During the profiling we just schedule the tasks by assuming the I_{comp} of the workload is 4 (hence, pipeline width is set to be a default value 8). After determining the task-core mapping, the fetch and issue widths for each core are also determined by using our workload-aware pipeline control technique. To realize dynamic control of the fetch and issue widths, we need an additional logic in each core as shown in Fig. 3. There are 2-bit selection signal for MUX to distinguish four different widths (1, 2, 4, and 8). The operating system determines pipeline width for each core and feeds an appropriate selection signal to the MUX. The selected width from the MUX is fed into the fetch and issue control logic in each core. The fetch and issue control logic in each core limits the maximum number of instructions which can be fetched and issued within a cycle, respectively.

In terms of the memory pressure mitigation in multi-programmed environment, our technique has a positive impact on reducing memory pressure as we reduce the pipeline width of the cores which execute memory-intensive workloads. Since our technique determines pipeline widths depending on the memory-intensiveness (or computation-intensiveness), our technique can dynamically control the memory bandwidth pressure, which eventually results in a positive impact on system-wide performance.

DOI: 10.1587/elex.15.20171253
Received December 19, 2017
Accepted January 9, 2018
Publicized January 24, 2018
Copyedited February 10, 2018

IEICE Electronics Express, Vol.15, No.3, 1–12

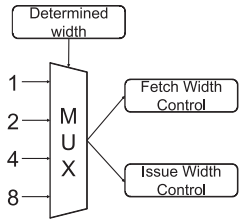


Fig. 3. Control logic diagram.

Table III. Thermal-related parameters used in this work.

Categories	Parameters used
Silicon thermal resistivity	0.01 mK/W
Metal-layer thermal resistivity	0.0025 mK/W
Ambient temperature	318.15 K
Convection thermal resistance	1.05 K/W

4 Evaluations

4.1 Evaluation framework

For performance and energy evaluation we use gem5 [17] SE (system-call emulation) mode and McPAT [9] (22 nm technology node), respectively. For DRAM power consumption, we use NVMain [18]. For thermal evaluation, we use HotSpot [19] architectural thermal modeling tool. The entire evaluation flow is shown in Fig. 4. To reflect reduced refresh intervals, when the steady state temperature of any DRAM bank exceeds 85°C (358.15 K) we re-run gem5 simulation with the reduced T_{REFI} (a refresh interval of DRAM) and re-extract power and thermal results. We use the thermal-related parameters as shown in Table III. We use the convection thermal resistance of 1.05 which corresponds to low∼mid-end cooling solution [20] as Wide-I/O is typically employed in mobile computer systems (please note that mobile systems are hard to adopt expensive cooling solutions).

For workloads, we run selected applications from SPEC2006 CPU benchmark suite. For multi-programmed environment, we generate eight workload groups by mixing the selected SPEC2006 workloads randomly. Table IV summarizes our workload groups each of which contains eight SPEC2006 benchmark applications and task-core mapping results when applying our thermal-aware task-core mapping. In the case where our task-core mappings are not applied, the tasks are mapped to the cores randomly. To improve our simulation accuracy, we fastforward 1 billion instructions and actually run 5 million instructions for each workload.

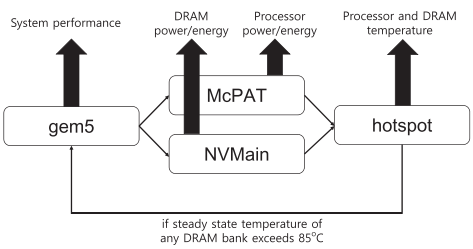


Fig. 4. Our performance-energy-thermal evaluation flow.

DOI: 10.1587/elex.15.20171253
Received December 19, 2017
Accepted January 9, 2018
Publicized January 24, 2018
Copyedited February 10, 2018

IEICE Electronics Express, Vol.15, No.3, 1–12

Table IV. Workload groups and core mappings when applying our thermal-aware task-core mapping.

	Core0	Core1	Core2	Core3	Core4	Core5	Core6	Core7
Group1	astar	mcf	gcc	bzip2	namd	cactusADM	lbm	omnetpp
Group2	povray	bwaves	gcc	calculix	sjeng	GemsFDTD	zeusmp	h264ref
Group3	astar	mcf	bwaves	h264ref	omnetpp	GemsFDTD	lbm	povray
Group4	astar	lbm	GemsFDTD	namd	calculix	cactusADM	zeusmp	omnetpp
Group5	povray	bwaves	gcc	namd	sjeng	cactusADM	zeusmp	omnetpp
Group6	astar	mcf	gcc	namd	calculix	GemsFDTD	zeusmp	omnetpp
Group7	astar	lbm	GemsFDTD	bzip2	calculix	cactusADM	bwaves	h264ref
Group8	povray	mcf	gcc	bzip2	sjeng	cactusADM	bwaves	h264ref

4.2 Energy

Fig. 5 describes energy results normalized to the baseline. Please note that Map_only and PC_only correspond to the cases where either task-core mapping (Map_only) or pipeline control (PC_only) technique is applied, respectively. Map+PC corresponds to the case where both task-core mapping and pipeline control techniques are adopted.

Thanks to our thermal-aware task-core mapping and pipeline control (Map+PC), energy consumption of the system is reduced by 7.6%, on average. When compared to the case of Map_only or PC_only, Map+PC leads to more energy reduction by 7.5% and 0.2%, respectively. In the case of Group6, one can obtain more than 10% of energy reduction compared to the baseline. The explanation for the energy reductions in the case of Map+PC is twofold. Firstly, our proposed technique throttles pipeline widths in a workload-aware manner. It eventually leads to dynamic energy reduction of the processors with negligible performance losses. Secondly, we can expect performance improvement by reducing the DRAM refreshes thanks to lower temperature. The reduced execution time (i.e., performance improvement) decreases total leakage energy consumption because the leakage energy consumption is proportional to the execution time. If our proposed technique reduces temperatures of all the DRAM banks below 85°C, we can expect energy savings thanks to the reduced refresh operations in DRAM banks.

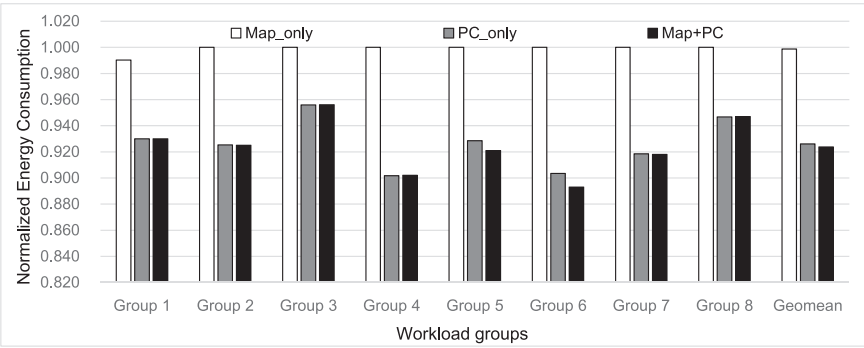


Fig. 5. Energy consumption results (normalized to that of the baseline system) in three cases: Map_only, PC_only, and Map+PC.

DOI: 10.1587/elex.15.20171253
Received December 19, 2017
Accepted January 9, 2018
Publicized January 24, 2018
Copyedited February 10, 2018

IEICE Electronics Express, Vol.15, No.3, 1–12

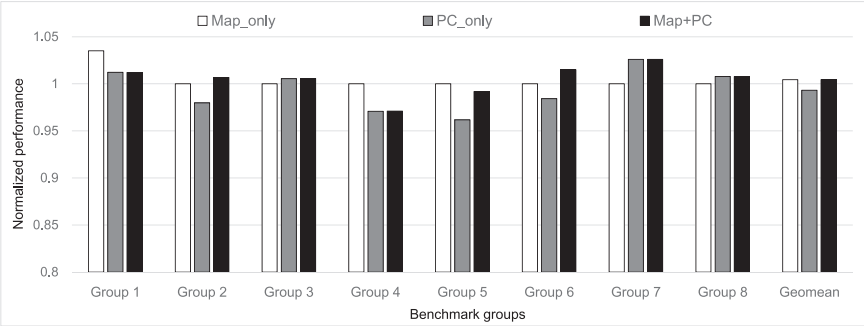


Fig. 6. Performance results (normalized to that of the baseline system) in three cases: Map_only, PC_only, and Map+PC.

4.3 Performance

Fig. 6 summarizes performance results of Map_only, PC_only, and Map+PC, which are normalized to the performance result of the baseline. Map_only leads to similar performance results compared to the Map+PC. The main reason of the performance benefit from the Map_only is reduced refresh rates (Group1) thanks to temperature reductions in DRAM banks. In the case of PC_only, performance is rather reduced compared to the baseline. This is mainly because of the reduced pipeline bandwidths. In contrast, thanks to synergistic effects of task-core mapping and pipeline width control, Map+PC improves performance of the system by 0.4% (on average) compared to the baseline.

In the case of Group7, performance improvement of Map+PC is up to 2.6%. Similarly, in the cases of Group1, Group2, Group3, Group6, and Group8, our proposed technique (Map+PC) improves performances of the system by 1.2%, 0.7%, 0.6%, 1.5%, and 0.8%, respectively. In the cases of Group1, Group2, Group6, Group7, and Group8, the main reason of performance improvement is the reduced refreshes in the Wide-I/O DRAM. In the case of Group3, the main reason of performance improvement is reduced contention to the shared resources, which is attributed to our pipeline bandwidth control. In the cases of the rest workload groups (Group4 and Group5), our proposed technique shows performance overhead (0.8%~2.9%) mainly because of the reduced pipeline widths though Map+PC results in less energy consumption compared to the baseline (as already shown in Fig. 5).

In the cases of Group2 and Group6, Map_only and PC_only show the same or worse performance compared to the baseline. On the contrary, Map+PC shows better performance compared to the baseline. It shows the synergistic effects of our proposed techniques which means combining both techniques (Map+PC) leads to the best results among four cases (baseline, Map_only, PC_only, and Map+PC).

4.4 Temperature

Our technique also contributes to the DRAM temperature reductions. Table V shows the number of DRAM banks which exceed 85°C across four different cases (baseline, Map_only, PC_only, and Map+PC).

Among 64 DRAM banks, there are non-negligible DRAM banks of which temperature is above 85°C in the case of the baseline. In the case of Map_only, we

DOI: 10.1587/elex.15.20171253
Received December 19, 2017
Accepted January 9, 2018
Publicized January 24, 2018
Copyedited February 10, 2018

IEICE Electronics Express, Vol.15, No.3, 1–12

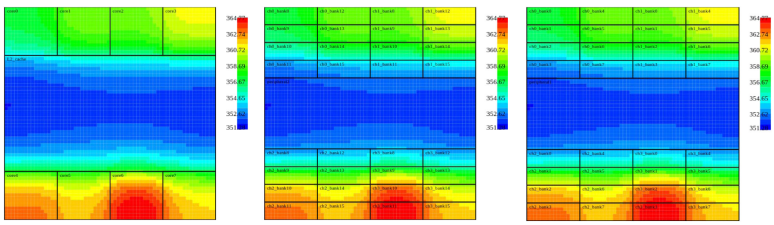
Table V. The number of DRAM banks that exceed 85°C.

	Group1	Group2	Group3	Group4	Group5	Group6	Group7	Group8
Baseline	2	8	16	17	10	8	2	3
Map_only	0	4	14	16	7	5	1	1
PC_only	0	1	11	2	1	1	0	0
Map+PC	0	0	6	2	0	0	0	0

can reduce the number of DRAM banks over 85°C by 32.9% on average. When we only adopt the pipeline control technique (PC_only), we can reduce the number of DRAM banks over 85°C by 81.3% on average. When adopting both techniques, the number of DRAM banks that exceeds 85°C is reduced by 92.8%, on average. By effectively reducing temperature of DRAM banks, our proposed technique can also expect the improvement of DRAM lifetime reliability.

Our proposed techniques are actually orthogonal to each other. Either task-core mapping or pipeline width control can be employed in the system. However, only employing either task-core mapping (Map_only) or pipeline width control (PC_only) does not lead to satisfactory temperature reduction results (in terms of the number of DRAM banks which exceed 85°C). In contrast, when applying both techniques (Map+PC), the results show that we can significantly reduce the number of DRAM banks which exceed 85°C. This in turn leads to elimination of the DRAM banks which exceed 85°C, resulting in reduced refresh rates. It implies our two techniques do not have any contradiction (one does not hurt the other) and have a synergistic effect on reducing temperatures of DRAM banks as well as processor cores. This is also presented in energy and performance results which show Map+PC leads to the best energy and performance results among three configurations (Map_only, PC_only, and Map+PC).

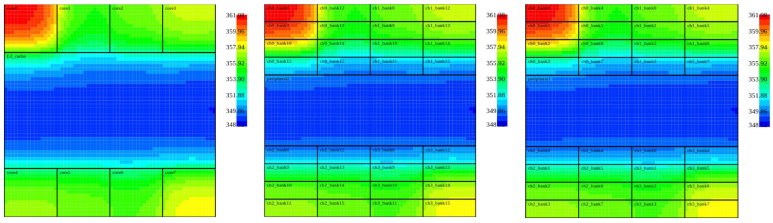
Fig. 7 and 8 show example thermal maps of each layer in the cases of baseline and our proposed technique (Map+PC), respectively, when running workload Group4. As shown in the thermal maps, our proposed technique reduces temperatures of DRAM banks (by up to 7.8°C in the case of Group4) as well as processor cores. Without our proposed technique, processor core4 and core6 becomes thermal hotspots, which eventually increases temperatures of upper layer DRAM banks. On the contrary, our proposed technique relieves thermal hotspots in processor dies (core4 and core6 becomes cool when adopting our technique), also resulting in temperature reduction in DRAM banks. As a result, it can lead to reduced refresh



(a) Processor die (layer1) (b) DRAM die 2 (layer2) (c) DRAM die 1 (layer3)

Fig. 7. Thermal maps (in K) of each layer in the case of baseline when running workload Group4.

DOI: 10.1587/elex.15.20171253
Received December 19, 2017
Accepted January 9, 2018
Publicized January 24, 2018
Copyedited February 10, 2018



(a) Processor die (layer1) (b) DRAM die 2 (layer2) (c) DRAM die 1 (layer3)

Fig. 8. Thermal maps (in K) of each layer in the case of our proposed technique when running workload Group4.

operations in DRAM banks, which in turn results in energy reduction and performance improvement of the system.

5 Conclusion

In this paper, we propose thermal-aware task-core mapping and processor pipeline control techniques to optimize energy and performance of the computer system which adopts 3D die-stacked DRAM. Both techniques have a synergistic impact on system energy and performance by reducing the temperatures in 3D DRAM banks. Our proposed technique leads to the system energy reduction of 7.6% along with performance improvement of 0.4%, on average. Considering that energy and performance are typically conflicting metrics, our technique leads to a good energy-performance trade-off. Our technique also reduces the number of DRAM banks exceed 85°C by 92.8% (on average), which can also improve DRAM lifetime reliability.

Acknowledgment

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT & Future Planning (2015R1C1A1A01051836). This research was also supported in part by Samsung Electronics.

DOI: 10.1587/elex.15.20171253
Received December 19, 2017
Accepted January 9, 2018
Publicized January 24, 2018
Copyedited February 10, 2018