ELectronics EXpress

LETTER

IEICE Electronics Express, Vol.14, No.24, 1–12

Energy-efficient heterogeneous memory system for mobile platforms

Dongsuk Shin[1,2a)], **Hakbeom Jang**[1b)], **and Jae W. Lee**[3c)]

[1] *College of Information and Communication Engineering, Sungkyunkwan University, 2066 Seobu-ro, Jangan, Suwon, Gyeonggi-do 16419, Republic of Korea*

[2] *System LSI Division, Samsung Electronics Company, DSR, Samsung-ro, Giheung, Yongin, Gyeonggi-do 17113, Republic of Korea*

[3] *Department of Computer Science and Engineering, Seoul National University, 1 Gwanak-ro, Gwanak-gu, Seoul 08826, Republic of Korea*

a) *dsshin@skku.edu*

b) *hakbeom@skku.edu*

c) *jaewlee@snu.ac.kr*

Abstract: The bandwidth demands from mobile application processors (APs) have been consistently growing to run multiple compute- and memory-intensive workloads concurrently. The Low-Power Double-Data Rate (LPDDR) DRAM family has been the de-facto standard for main memory, whose operating frequency has been scaled up to meet these demands. However, frequency scaling poses many design challenges due to limited power budget, timing, and power/signal integrity issues. JEDEC has recently released the WideIO2 DRAM standard to provide high bandwidth at a low frequency. However, WideIO2 DRAM cannot completely replace LPDDR devices because it cannot provide enough capacity and bandwidth by itself. Thus, this paper proposes a heterogeneous mobile memory system (HMMS) using both types of DRAM devices. HMMS employs an efficient page migration scheme to serve more requests from energy-efficient WideIO2 DRAM devices. HMMS uses a small on-chip page location table at each DRAM controller to track page remappings without requiring a master table in off-chip DRAM. To minimize bandwidth and energy wastes from excessive page migrations, HMMS selects strong hot pages for migration, adjusts the page migration threshold dynamically and employs a migration stop mechanism. Our evaluation using 10 multi-programmed workloads demonstrates that HMMS improves the performance and energy-delay product (EDP) by 13% and 21%, respectively, over the baseline heterogeneous memory system with no migrations.

Keywords: mobile DRAM, heterogeneous memory, LPDDR4, WideIO2

Classification: Integrated circuits

DOI: 10.1587/elex.14.20171002
Received September 27, 2017
Accepted November 10, 2017
Publicized November 30, 2017
Copyedited December 25, 2017

References

[1] https://en.wikipedia.org/wiki/Apple_mobile_application_processors.
[2] JEDEC LPDDR4 SDRAM standard. http://www.jedec.org/standards-documents/docs/jesd209-4b.
[3] Samsung Semiconductor, LPDDR4: Evolution for new mobile world, Memcon 2013. http://www.memcon.com/pdfs/proceedings2013/track1/LPDDR4_Evolution_for_a_New_Mobile_World.pdf.
[4] JEDEC WideIO2 SDRAM standard. https://www.jedec.org/standards-documents/docs/jesd229-2.
[5] 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).
[6] Y.-H. Kim *et al.*: "POP structure using PCB cavity," http://www.kipo.go.kr.
[7] J. Sim, *et al.*: "Transparent hardware management of stacked DRAM as part of memory," IEEE/ACM International Symposium on Microarchitecture (MICRO) (2014) (DOI: 10.1109/MICRO.2014.56).
[8] C. C. Chou, *et al.*: "CAMEO: A two-level memory organization with capacity of main memory and flexibility of hardware-managed cache," IEEE/ACM International Symposium on Microarchitecture (MICRO) (2014) (DOI: 10.1109/MICRO.2014.63).
[9] Y. Lee, *et al.*: "A fully associative, tagless DRAM cache," International Symposium on Computer Architecture (ISCA) (2015) 211 (DOI: 10.1145/2749469.2750383).
[10] E. Bolotin, *et al.*: "Designing efficient heterogeneous memory architectures," IEEE Micro **35** (2015) 60 (DOI: 10.1109/MM.2015.72).
[11] M. R. Meswani, *et al.*: "Heterogeneous memory architecture: A HW/SW approach for mixing die-stacked and off-package memories," Proc. IEEE 21st Int. Symp. High Performance Computer Architecture (HPCA) (2015) 126 (DOI: 10.1109/HPCA.2015.7056027).
[12] M.-C. Hsieh: "Advanced flip chip package on package technology for mobile application," International Conference on Electronic Packaging Technology (ICEPT) (2016) 486 (DOI: 10.1109/ICEPT.2016.7583181).
[13] J. H. Ahn, *et al.*: "McSimA+: A manycore simulator with application-level+ simulation and detailed microarchitecture modeling," Int. Symp. Performance Analysis of Systems and Software (ISPASS) (2013) 74 (DOI: 10.1109/ISPASS.2013.6557148).
[14] JEDEC High Bandwidth Memory (HBM) DRAM standard. http://www.jedec.org/standards-documents/docs/jesd235.
[15] nVidia, Tesla P100 white paper. https://images.nvidia.com/content/pdf/tesla/whitepaper/pascal-architecture-whitepaper.pdf.
[16] Micro, Hybrid memory cube - Documentation. https://www.micron.com/products/hybrid-memory-cube.

1 Introduction

Today's mobile devices are required to run many complex applications concurrently, which increases bandwidth pressure from the memory system. For example, Apple's A9X and A10 Fusion application processors representing the state-of-the-art, integrate four LPDDR4 dies with a CPU package to provide a massive bandwidth of up to 51.2 GB/s [1]. This trend is likely to continue in the foreseeable

DOI: 10.1587/elex.14.20171002
Received September 27, 2017
Accepted November 10, 2017
Publicized November 30, 2017
Copyedited December 25, 2017

future with an increase in display resolutions and emergence of novel AI applications that are known to be memory intensive, such as intelligent personal assistant, image/video classification, natural language processing, and so on.

LPDDR DRAM [2] is the de-facto standard for mobile memory, and its I/O frequency has been continuously scaled up to satisfy these demands. Today's LPDDR4 DRAM device is clocked at 1.6 GHz, and there is only limited headroom for future frequency scaling. Furthermore, frequency scaling imposes many design challenges, such as increased power consumption, tighter timing margin, and signal/power integrity issues. According to an LPDDR4 power report [3], the ratio of core power to I/O power is almost 1:1. Thus, even with lower-voltage I/O interface, the I/O power still accounts for a significant portion of total DRAM power due to high operating frequency and on-die termination (ODT).

Scaling the number of DRAM channels is not viable, either, as it would make the mobile AP SoC pad-limited, which in turn increases the die size and package size. Furthermore, more LPDDR channels can pose additional challenges for assembling a CPU die and DRAM dies in a single package-on-package (PoP) structure due to more congested inter-die connections. If off-package LPDDR4 devices are used, it would increase the package size significantly with more solder balls for CPU-DRAM communication.

To address this problem JEDEC has released the WideIO2 DRAM standard [4] to provide more bandwidth without frequency scaling. The CPU die and WideIO2 DRAM dies are connected by through-silicon vias (TSVs) [5], which have much lower RC-delay than the conventional off-package DRAM channels. However, this is only a half solution as more DRAM dies do not translate to higher DRAM bandwidth due to TSVs shared by all DRAM dies. According to the standard, the peak bandwidth of WideIO2 DRAM is limited to 51.2 GB/s regardless of the number of stacked DRAM dies.

To scale mobile DRAM bandwidth beyond what is offered by a single DRAM type, we propose to use both LPDDR4 and WideIO2 DRAM devices. Fig. 1 demonstrates a PoP structure (for LPDDR4 DRAM) augmented with TSV channels (for WideIO2 DRAM) [6]. LPDDR4 has two channels per die, and WideIO2 8 channels per stack [2, 4]. Since each channel has a peak bandwidth of 6.4 GB/s regardless of the DRAM type, the heterogeneous memory system can provide up to 76.8 GB/s if two LPDDR4 dies are used together with a stack of WideIO2 dies.

This memory configuration is different from the conventional "near-far" memory system in high-performance servers. In a server environment, the near

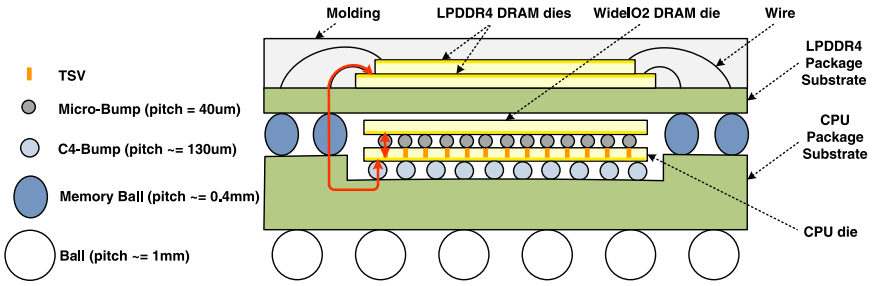

Fig. 1. PoP structure with TSV-SIP

DOI: 10.1587/elex.14.20171002
Received September 27, 2017
Accepted November 10, 2017
Publicized November 30, 2017
Copyedited December 25, 2017

(fast) memory may have 8× higher bandwidth than the far (slow) memory and/or much lower latency [7, 8, 9, 10, 11]. In this setup it is a good strategy to migrate hot DRAM pages to near memory whenever possible, to maximize the number of requests it services. However, this is not the case for the LPDDR4-WideIO2 hybrid memory as both devices have a narrow bandwidth gap and comparable access latency. If the system uses 2 LPDDR4 dies, their bandwidth is a half of the WideIO2 stack; if it uses 4 LPDDR4 dies, the bandwidth is the same for both. Thus, the simple migration strategy for a near-far memory server is suboptimal for a mobile device. Moreover, excessive migrations can significantly degrades the energy efficiency, which is a first-class design objective for the mobile device.

This paper proposes Heterogeneous Mobile Memory System (HMMS), an energy-efficient hybrid memory system composed of both WideIO2 and LPDDR2 DRAM devices. It has a flat address space to maximize the usable memory capacity (instead of using near memory as cache) and employs an efficient data migration scheme with no modifications to the OS. HMMS uses a small on-chip page location table to track page migrations and does not maintain any master table in off-chip DRAM. To prevent excessive DRAM page migrations, HMMS selects strong hot pages for migration, adjusts the page migration threshold dynamically and even prevents migrations if they are not beneficial. Our evaluation with 10 multi-programmed workloads demonstrates that HMMS improves the instructions-per-cycle (IPC) and energy-delay product (EDP) by 13% and 21%, respectively, compared to the baseline heterogeneous memory system with no DRAM page migrations.

2 Background and motivation

Today's mobile APs integrate 10's to 100's of IP blocks on a single chip and can be as large as 10 mm × 10 mm in die size. As the chip size gets bigger, it is becoming more challenging to place and route some 300 memory I/O connections within a PoP structure, where a typical package size is about 14 mm × 14 mm or 15 mm × 15 mm [12]. To scale DRAM bandwidth without increasing the package size, it is appealing to augment a PoP structure (for LPDDR4 DRAM) with TSV channels (for WideIO2 DRAM). Fig. 2 shows total DRAM bandwidth for varying the number of LPDDR4 channels with and without a stack of WideIO2 DRAM devices. To achieve >70 GB/s bandwidth, 12 LPDDR4 channels are needed as a single channel has a data rate of 6.4 GB/s. This will increase the package size due to additional power and signal paths. However, one can provide comparable bandwidth by using only 4 LPDDR4 channels together with a WideIO2 DRAM stack. As shown in Fig. 1, an AP SoC die can be connected to both WideIO2 DRAM via TSVs and LPDDR4 DRAM via either memory balls (for a PoP structure) or conventional solder balls (for on-board DRAM devices).

Fig. 3 shows the two logical organizations for heterogeneous memory systems (HMS) composed of fast stacked DRAM (e.g., WideIO2) and slow off-package DRAM (e.g., LPDDR4): (a) hardware-managed cache [9, 10] and (b) flat physical memory [7, 8, 10, 11]. The cache organization is software-transparent and does not require any change to the software stack. However, all accessed cache blocks

DOI: 10.1587/elex.14.20171002
Received September 27, 2017
Accepted November 10, 2017
Publicized November 30, 2017
Copyedited December 25, 2017

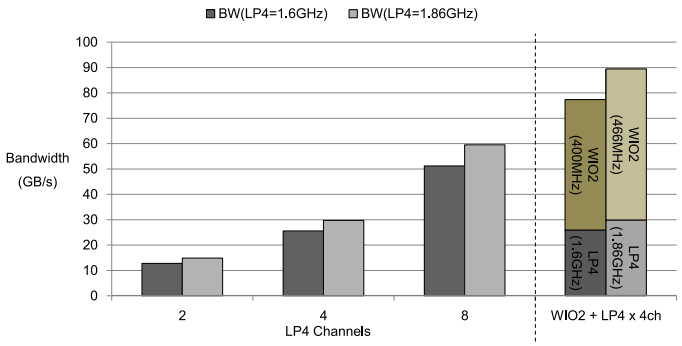

Fig. 2. Total bandwidth when using LPDDR4 and WideIO2

should be allocated to the stacked DRAM first before they are accessed by the CPU. This can be very slow as cache tags are often stored in DRAM together with data, which has long latency. Besides, the usable capacity is reduced due to duplicated copies in both fast and slow DRAM regions.

The second organization is flat physical address space, where both DRAM regions are mapped to the same physical address space. While it allows the user to use the entire address space, it requires modifications to the OS and/or memory controllers to efficiently use fast stacked DRAM. Typically, hot DRAM pages are tracked by either hardware or software and migrated to the fast memory region, while cold pages are to the slow region. In this way the system can serve more requests from the fast, energy-efficient stacked DRAM to improve performance and energy efficiency.

The primary challenge in designing a heterogeneous memory system is how to effectively migrate hot (cold) pages to the fast (slow) region. If the bandwidth and latency gap between fast and slow regions is small as in mobile DRAMs, the cost of migration can easily outweigh its benefits. Migration at a coarse granularity (say, DRAM page size) [7, 11] has advantage of better exploiting DRAM row buffer locality over fine granularity (say, CPU cache block size) [8]. Since the existing migration heuristics are designed for heterogeneous memory devices with a relatively large performance gap [7, 8, 9, 10, 11], they can be suboptimal for a LPDDR4-WideIO2 heterogeneous memory system. Thus, we propose a novel page migration technique customized for resource-constrained mobile devices.

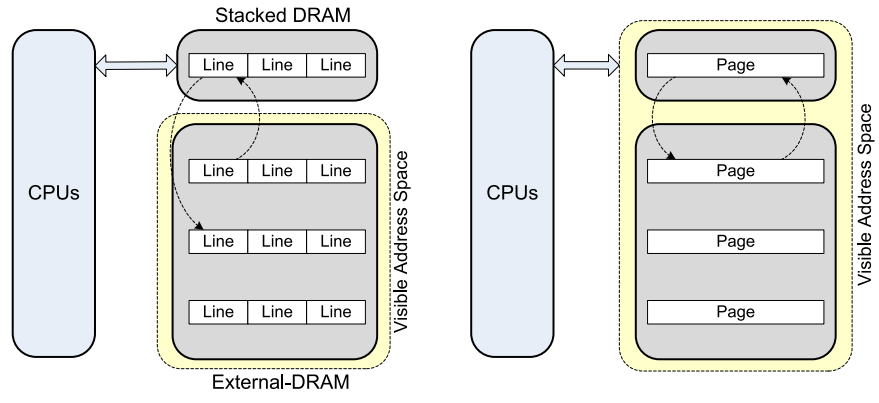

Fig. 3. Two logical organizations of HMS

DOI: 10.1587/elex.14.20171002
Received September 27, 2017
Accepted November 10, 2017
Publicized November 30, 2017
Copyedited December 25, 2017

3 HMMS: architecture and design

To minimize bandwidth and energy waste caused by excessive migrations, page migrations and remapping table accesses must be controlled judiciously. HMMS employs a throttling mechanism to achieve this. Since LPDDR4 and WideIO2 DRAM devices have the same page size of 2 KB, we choose migration at a page granularity to best exploit row buffer locality.

Page Location Table (PLT) Organization. PLT is a table that maintains mappings for migrated pages. In prior work PLT is organized in a hierarchical manner with a master PLT in DRAM and a small on-die PLT cache [7, 8, 9, 11]. To remove the latency cost for an on-die PLT miss, HMMS uses a small on-die PLT only with no master PLT in DRAM, where a PLT miss implies no migration for the requested page.

Fig. 4 shows the organization of PLT assuming a 2:1 capacity ratio for LPDDR4 and WideIO2. At most one pair of WideIO2 and LPDDR4 pages can be swapped within a *page group*, which is a set of disjoint pages with equal distances (N). In the example configuration, a page group has 8 sets, where each set has one WideIO2 page and two LPDDR4 pages shown in gray. PLT consists of two tables. The Selective Remapping Table maintains the status of migration for each page group. The Candidate Remapping Table tracks the hottest LPDDR4 page for future migration. The size of PLT (N) is derived from Eq. (1), where CW is the capacity of WideIO2 and S is the size of the page group. For example, if CW = 2 GB and $S = 8$, N is equal to 128 K.

$$N = \frac{CW}{2\,\text{KB} \times S} \tag{1}$$

PLT Operations. Fig. 5 illustrates the operation of PLT when $S = 8$. All entries in PLT are initialized to zero (no migration). When a memory request arrives, the address is compared with the *Remap_Tag* of the corresponding entry in the Selective Remapping Table and *Candi_Tag* in the Candidate Remapping Table. *Remap_Tag* keeps the ID of the set in which a migration has happened. *Candi_Tag*,

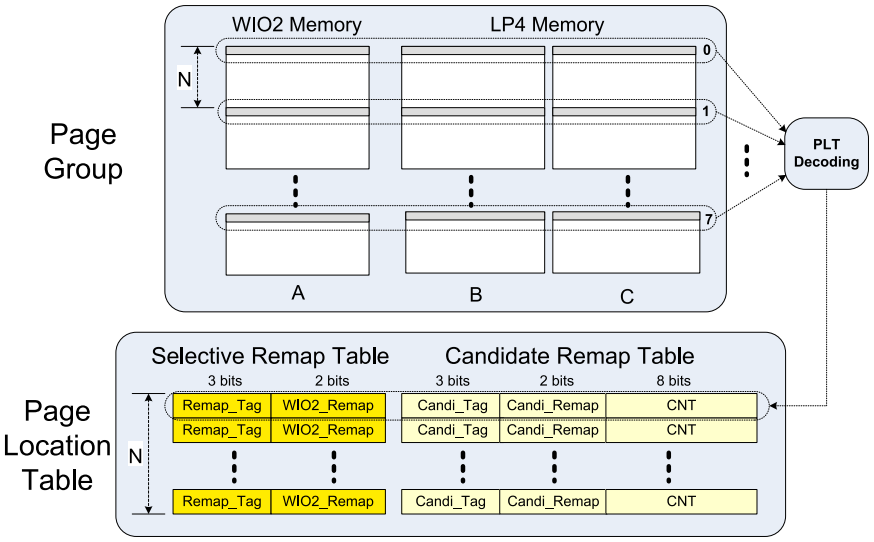

Fig. 4. PLT organization

DOI: 10.1587/elex.14.20171002
Received September 27, 2017
Accepted November 10, 2017
Publicized November 30, 2017
Copyedited December 25, 2017

together with *Candi_Remap*, keeps the ID of the hottest LPDDR4 page. If the requested page matches *Remap_Tag*, the request is forwarded to either WideIO2 or LPDDR4 in the set depending on the value of *WIO2_Remap*. The two bits of *WIO2_Remap* encode where the original WideIO2 page has been migrated to within the set: `2'b00` = WideIO2 (Column A), `2'b01` = LPDDR4 (Column B) and `2'b10` = LPDDR4 (Column C). Thus, if there is no migration, *WIO2_Remap* is set to `2'b00`. In Fig. 5 3B is a hot page that is initially placed in LPDDR4 but migrated to WideIO2. This page has competed successfully against the other 15 LPDDR4 pages in the page group to secure the WideIO page originally taken by Page 3A.

If the requested page does not match *Remap_Tag*, there is no migration for the set and the request goes to the original page. If the memory request falls on LPDDR4, *Candi_Tag* and *Candi_Remap* are looked up. If the page matches, the candidate remapping counter (*CNT* in Fig. 4) is incremented by one. If this value goes above a threshold, the page is identified as hot and a migration procedure is invoked to swap the LPDDR4 page with the WideIO2 page in the same set. Once completed, *Remap_Tag* and *WIO2_Remap* are updated accordingly. If the requested page differs from the candidate LPDDR4 page being tracked (specified by *Candi_Tag* and *Candi_Remap* fields), these two fields are updated to point to the new LPDDR4 page being serviced and the counter is reset to zero.

Fig. 6 illustrates the behaviors of the candidate remapping counter. Assume that Page 1B and 5C are being accessed in LPDDR4 as migration candidates and that Page 3B is migrated to the WideIO2 page in the set (originally occupied by Page 3A) as shown in Fig. 5. If requests to Page 5C arrive consecutively for the page group to reach the threshold (Fig. 6(a)), a page migration is invoked to swap Page 5A and 5C. If a request to another LPDDR4 page, say, Page 1B, intervenes before the counter hits the threshold (Fig. 6(b)), the counter is reset to zero and starts to track the new page (1B) as a migration candidate. Finally, if a request to

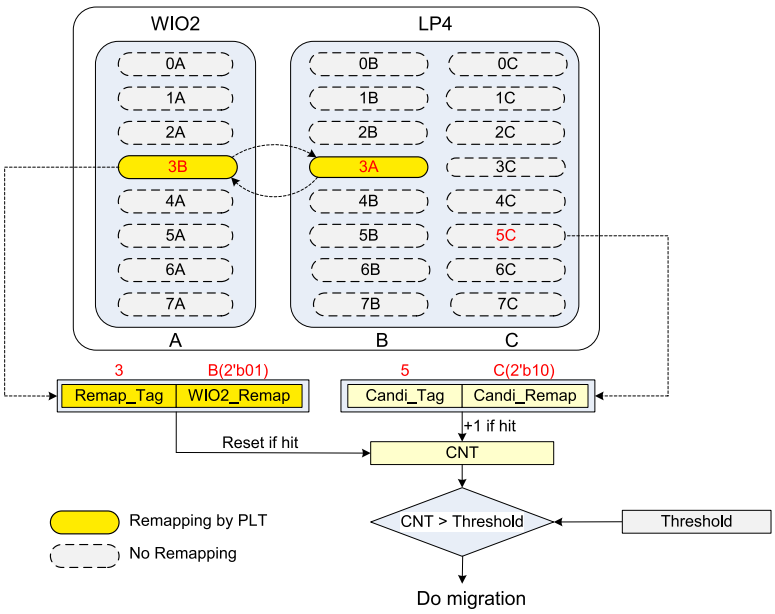

Fig. 5. PLT operations

DOI: 10.1587/elex.14.20171002
Received September 27, 2017
Accepted November 10, 2017
Publicized November 30, 2017
Copyedited December 25, 2017

the WideIO2 page specified by *Remap_Tag* and *WIO2_Remap* (i.e., Page 3B) intervenes (Fig. 6(c)), the counter is again reset to zero but still tracks the existing LPDDR4 page (5C). Note that this migration scheme is conservative in that a page migration is triggered only when a strong hot page is identified by a certain number of consecutive accesses within the page group. Although double migrations can happen if another LPDDR4 page in the same page group currently occupies the WideIO2 page in the set (i.e., *WIO2_Remap* is not `2'b00`), the probability of this is very low as the strong hot page migrated to WideIO2 is likely to reset the counter frequently.

Dynamic Threshold. The migration threshold is the primary knob to control the amount of page migrations. HMMS periodically adjusts it by taking into account the bandwidth usage of LPDDR4 devices. A higher threshold value suppresses page migrations; a lower threshold value yields more migrations. The threshold value is determined as follows (Eq. 2):

$$Threshold\_value = Migration\_overhead \times (1.5 + delta\_threshold) \quad (2)$$

The migration overhead is the cost for data transfers between LPDDR4 and WideIO2 for a page migration. Since the bandwidth of WideIO2 is twice as much as LPDDR4, we count the overhead of migration in WideIO2 as a half of that in LPDDR4. We select the migration overhead value of 96 by default to reflect the cost for transferring one and a half LPDDR4 pages. Since HMMS delivers 64 bytes for a single data transfer, 96 requests amounts to transferring three 2 KB pages. Besides, *delta_threshold* is adjusted like Fig. 7 to have a lower value at an initial phase of migration when memory accesses are mostly directed to LPDDR4. The value is evaluated periodically at every 2048 memory accesses, and determined by the number of LPDDR4 requests (in x axis) during the period. The updated *delta_threshold* is used for the next 2048 memory accesses. Assuming a 2:1 capacity ratio between LPDDR4 and WideIO2, the number of LPDDR4 accesses is twice that of WideIO2 if memory requests are uniformly distributed. As we perform more migrations, the portion of requests to LPDDR4 will diminish rapidly,

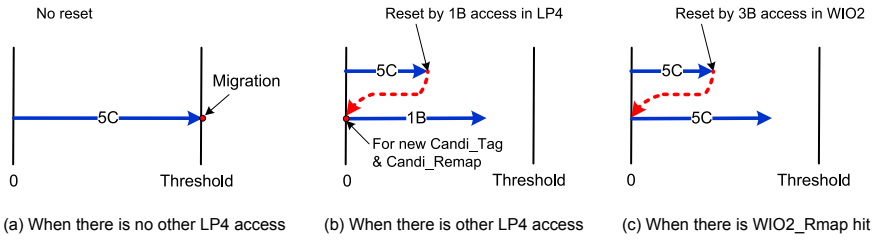

Fig. 6. Behaviors of candidate remapping counter

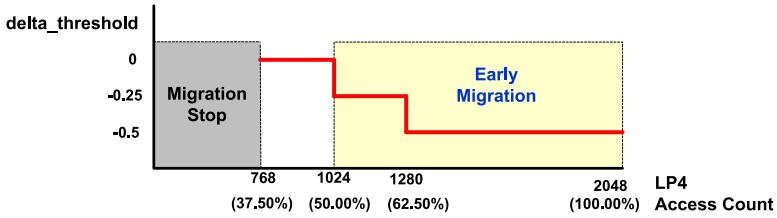

Fig. 7. Dynamic threshold value

DOI: 10.1587/elex.14.20171002
Received September 27, 2017
Accepted November 10, 2017
Publicized November 30, 2017
Copyedited December 25, 2017

resulting in a high threshold value. In this way HMMS effectively reduces the number of migrations, and hence bandwidth and energy wastes, as most memory requests already go to WideIO2 devices. With varying *delta_threshold*, the ratio of maximum and minimum threshold values in Eq. (2) is 1.5.

Migration Stop. In fast and slow heterogeneous memories, more migrations are generally desirable to maximize the utilization of the fast memory. However, too many migrations can be harmful for an LPDDR4-WideIO2 heterogeneous memory, where the performance gap between the two devices are relatively small. Thus, we adopt a migration stop mechanism when LPDDR4 was accessed fewer than 68 times during the previous 2048 memory accesses like Fig. 7 (equivalently, WideIO2 is accessed more than 1280 times during the same period). These numbers include about 4% margin from 33% and 66% points, respectively, when the bandwidth ratio of LPDDR4 and WideIO2 is 1:2. This prevents excessive bandwidth pollution caused by excessive migrations during a particular period.

4 Evaluation

We evaluate the performance and energy efficiency of HMMS using the McSimA+ simulator [13]. Table I shows architectural parameters. The modeled system has eight out-of-order cores with 2 GB WideIO2 DRAM which is directly connected to the CPU die with TSV channels, and 4 GB LPDDR4 DRAMs. The total bandwidth of WideIO2 is twice than LPDDR4 and the capacity ratio of LPDDR4 and WideIO2 is 2:1. This eight-core machine runs eight multi-programmed memory-intensive workloads composed of SPEC CPU 2006 programs as shown in Table II, and 1.2 billion instructions are simulated.

We compare the following 8 designs:

- No Swap with 4 GB LPDDR4 and 2 GB WideIO2 (baseline, No_Swap(2vs1))
- No Swap with 2 GB LPDDR4 and 4 GB WideIO2 (No_Swap(1vs2))
- POM [7] with 32/64/128 K remap cache, 4 GB LPDDR4 and 2 GB WideIO2
- HMMS with 32/64/128 K PLT, 4 GB LPDDR4 and 2 GB WideIO2

Table I. Architecture parameters

Component	Parameters
CPU	Out-of-order, 8 cores, 3.2 GHz
L1 Cache	4-way, 32 KB I-cache, 32 KB D-cache, 64 B line, 2 cycles
L2 Cache	16-way, 2 MB shared cache per core, 64 B line, 6 cycles
PLT entries	32 K/64 K/128 K, 2 cycles
WideIO2 DRAM (2 GB)	
Bus frequency	400 MHz (DDR 800 MHz)
Channel/Bank	8 channels/4 banks per channel
Bus width	64 bits per channel
LPDDR4 DRAM (4 GB)	
Bus frequency	1.6 GHz (DDR 3.2 GHz)
Channel/Bank	4 channels (2 dies)/8 banks per rank
Bus width	16 bits per channel

DOI: 10.1587/elex.14.20171002
Received September 27, 2017
Accepted November 10, 2017
Publicized November 30, 2017
Copyedited December 25, 2017

Table II. Workload groupings

Case0	sphinx3-leslie3d-milc-omnetpp-soplex-GemsFDTD-lbm-mcf
Case1	zeusmp-cactusADM-milc-omnetpp-soplex-GemsFDTD-lbm-mcf
Case2	zeusmp-cactusADM-wrf-sphinx3-soplex-GemsFDTD-lbm-mcf
Case3	zeusmp-wrf-sphinx3-leslie3d-milc-GemsFDTD-lbm-mcf
Case4	cactusADM-wrf-sphinx3-leslie3d-milc-omnetpp-soplex-mcf
Case5	zeump-wrf-leslie3d-omnetpp-soplex-GemsFDTD-lbm-mcf
Case6	cactusADM-wrf-sphinx3-leslie3d-milc-soplex-lbm-mcf
Case7	zeusmp-cactusADM-sphinx3-leslie3d-omnetpp-soplex-lbm-mcf
Case8	cactusADM-wrf-sphinx3-leslie3d-omnetpp-soplex-GemsFDTD-mcf
Case9	wrf-sphinx3-leslie3d-milc-soplex-GemsFDTD-lbm-mcf

Fig. 8 shows the normalized IPC of our proposed scheme, static mapping scheme without migration and PoM [7] for comparisons. Static mapping is a baseline and POM is a migration scheme using internal remapping cache and external remapping table in fast and slow memories. HMMS shows about 13% of average IPC improvement compared to baseline and about 8% compared to PoM. The IPCs of POM are influenced by remapping cache size that can make memory latency when there are misses at the internal remapping cache. But HMMS has similar IPCs regardless of PLT size because the number of migration is smaller than PLT size with 32 K entries. Case0 has the most memory-intensive workloads and shows better performance than any other cases. This means HMMS can operate in memory-intensive applications well.

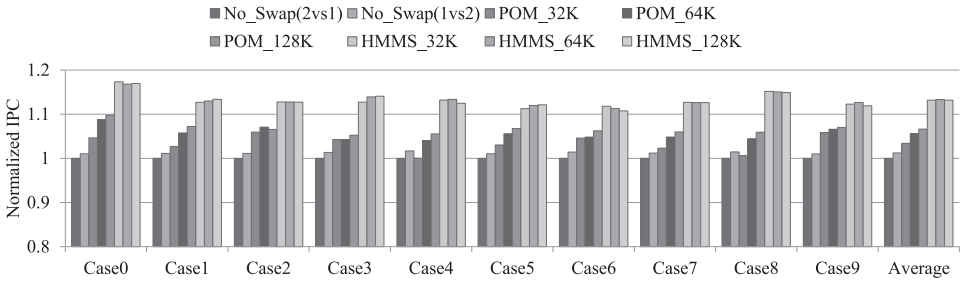

Fig. 8. Normalized IPC

Fig. 9 shows the normalized bandwidth to the baseline in Case0. BW_FULL means full bandwidth including data and migration, and BW_DATA means only data bandwidth for data transfer without migration. In comparison with POM cases,

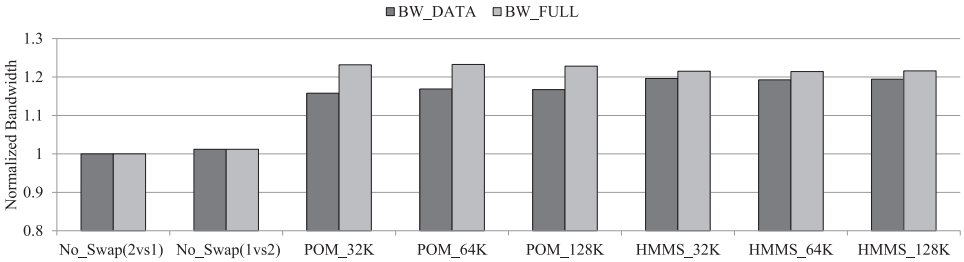

Fig. 9. Normalized bandwidth in Case0

DOI: 10.1587/elex.14.20171002
Received September 27, 2017
Accepted November 10, 2017
Publicized November 30, 2017
Copyedited December 25, 2017

HMMS cases have higher data bandwidth and small gap between BW_DATA and BW_FULL. HMMS selects more correct hot page and swaps the smaller number of pages between LPDDR4 and WideIO2 thanks to conservative migration scheme. This reduces migration overhead that comes from excessive migrations and external remapping table access. Particularly No_Swap(1vs2) in Fig. 8 does not have better performance than POM and HMMS because the bandwidth is similar to baseline. This means the higher data bandwidth the system has the higher performance it has.

We analyze the Energy-Delay Product (EDP) for different designs like Fig. 10. We assume that CPU core consumes 80% of the power and the rest is for memories. And the power ratio between LPDDR4 and WideIO2 is 2:1 and its value is referenced from [3]. HMMS shows about 21% of average EDP improvement compared to baseline and up to 17% compared to PoM_32K.

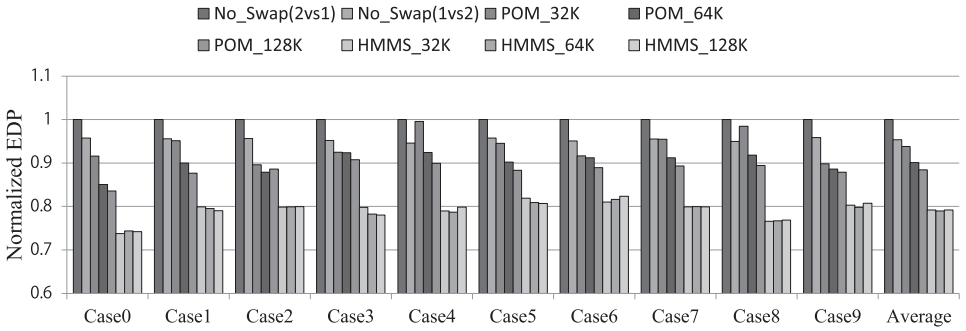

Fig. 10. Normalized EDP

5 Related work

Many existing heterogeneous memory systems [7, 8, 9, 10, 11] focus on the combination of fast and slow memories. Fast memory has short latency and much higher bandwidth, but slow memory has long latency and smaller bandwidth. To get more performance, fast memory can be used either as large cache [9, 10] or part of physical memory [7, 8, 10, 11]. These existing techniques require a big table for either tags or remapping entries in DRAM, which takes a long latency to access. However, they become suboptimal for a LPDDR4-WideIO2 heterogeneous memory system, where the bandwidth and latency gap between the two memory devices is relatively small. In contrast, HMMS is optimized for the heterogeneous mobile memory system in particular, and achieves more robust performance than a state-of-the-art technique [7].

Like WideIO2, High Bandwidth Memory (HBM) and Hybrid Memory Cube (HMC) also provide high bandwidth with a small form factor leveraging 3D die-stacking and silicon interposer technologies [14, 15, 16]. However, these devices are not suitable for power-constrained mobile platforms. HBM primarily targets high-end GPUs, and HMC networking and high-performance computing platforms. WideIO2 trades bandwidth and bank count for higher power efficiency. Thus, we use WideIO2 to architect an HMS for mobile platforms.

DOI: 10.1587/elex.14.20171002
Received September 27, 2017
Accepted November 10, 2017
Publicized November 30, 2017
Copyedited December 25, 2017

6 Conclusion

The bandwidth demands from mobile APs has been growing continuously, and it is becoming more and more difficult to satisfy this with DRAM I/O frequency scaling. To address this, we propose HMMS, an LPDDR4-WideIO2 hybrid memory system for mobile devices, which reduces the bandwidth waste for excessive migrations and eliminate long-latency accesses for the remapping table in DRAM. Overall, HMMS improves the performance and energy-delay product (EDP) by 13% and 21%, respectively, over the baseline heterogeneous memory system with no migrations.

Acknowledgments

This work was supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) (B0101-17-0644, Research Project on High Performance and Scalable Manycore Operating System). Jae W. Lee is the corresponding author.

DOI: 10.1587/elex.14.20171002
Received September 27, 2017
Accepted November 10, 2017
Publicized November 30, 2017
Copyedited December 25, 2017