POSTER: SILC-FM: Subblocked InterLeaved Cache-Like Flat Memory Organization

Jee Ho Ryoo
Department of Electrical and
Computer Engineering
The University of Texas at
Austin
Austin, TX
jr45842@utexas.edu

Mitesh R. Meswani
AMD Research
Austin, TX
mitesh.meswani@amd.com

Reena Panda,
Lizy K. John
Department of Electrical and
Computer Engineering
The University of Texas at
Austin
Austin, TX
reena.panda@utexas.edu,
ljohn@ece.utexas.edu

ABSTRACT

In this paper, we present a flat address space organization called SILC-FM that allows subblocks from two pages to co-exist in an interleaved fashion in die-stacked DRAM. Data movement at subblocked granularity consumes less bandwidth compared to migrating the entire large block and prevents fetching useless subblocks that may never get accessed. SILC-FM can get more spatial locality hits than CAMEO and PoM due to page-level operation and interleaving blocks respectively. The interleaved subblock placement improves performance by 55% on average over a static placement scheme without data migration. We also selectively lock hot blocks to prevent them from being involved in the hardware swapping operations. Additional features such as locking, associativity and bandwidth balancing improve performance by 11%, 8%, and 8% respectively, resulting in a total of 82% performance improvement over no migration static placement scheme. Compared to the best state-of-the-art scheme, SILC-FM gets performance improvement of 36%.

1. INTRODUCTION

Die-stacked DRAM is an emerging technology that offers much higher bandwidth than the conventional DDR technology [8, 11, 12, 21]. Over the years, the capacity of available die-stacked DRAM modules has increased steadily from a few hundred megabytes to a few gigabytes, so exposing the die-stacked DRAM capacity for application usage can make an impact for capacity constrained applications [3]. Researchers focus on ways to manage tags (metadata) efficiently by managing data at different cacheline sizes to reduce the metadata storage overheads, while at the same time, they try to perform selective cacheline fetching from

the off-chip DRAM to reduce off-chip bandwidth usage [4, 5, 6, 7, 13, 14, 15, 17, 19, 22, 24, 25].

In this work, we present SILC-FM, Subblocked InterLeaved Cache-Like Flat Memory Organization, which fully utilizes the die-stacked DRAM capacity while intelligently placing hot pages in die-stacked DRAM. The subblock level data movement reduces the bandwidth usage compared to PoM [23] and prevents fetching data that may not get utilized much. However, SILC-FM fetches a higher number of small size blocks at a time, resulting in more spatial locality hits compared to CAMEO [3]. In effect, SILC-FM gets benefits of both CAMEO and PoM. Furthermore, SILC-FM allows interleaving between die-stacked and off-chip DRAM hot blocks, so subblocks from both die-stacked and off-chip DRAM can coexist together in a die-stacked DRAM page. Our scheme is a hardware data management scheme that is robust to common problems associated with hardware data management such as thrashing and conflicts since SILC-FM can exclude and lock hot pages from being involved in data migration operations. Our associative structure also protects those pages that are not locked and are actively participating in hardware data migrations from being frequently swapped out of die-stacked DRAM. In addition, we maximize the memory bandwidth by utilizing both die-stacked memory bandwidth and off-chip memory bandwidth. We achieve this by moving only a certain subset of memory traffic to off-chip memory rather than servicing all memory requests from die-stacked DRAM. Thus, we further improve performance by utilizing the sum of both die-stacked and off-chip DRAM bandwidth.

2. SILC-FM MEMORY ARCHITECTURE

In this section, we will explain details of the SILC-FM architecture. The SILC-FM scheme uses Near Memory (NM) as OS visible space while internally operating with subblock based mechanisms. Each block contains four subblocks and the mapping is direct-mapped. This implies that multiple subblocks from only one large block (from the same set) in Far Memory (FM) can swap into corresponding subblocks in NM at any one point of time. The migration between two pages within the same set occurs at subblock granularity. This is bandwidth efficient as only 64B worth of data is migrated, yet managing the remap table at a large block granularity reduces the remap table overheads. The subblock tracking is done using a bit vector per NM block where indi-

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

PACT '16 September 11-15, 2016, Haifa, Israel

© 2016 Copyright held by the owner/author(s).

ACM ISBN 978-1-4503-4121-9/16/09.

DOI: <http://dx.doi.org/10.1145/2967938.2974060>

Processor	Values
Number of Cores (Freq)	8 (3.2GHz)
Width	4 wide out-of-order
Caches	Values
L1 I-Cache (private)	64 KB, 2 way, 4 cycles
L1 D-Cache (private)	16 KB, 4 way, 4 cycles
L2 Cache (shared)	8 MB, 16 way, 11 cycles
HBM	Values
Bus Frequency	800 MHz (DDR 1.6 GHz)
Bus Width	128 bits
Channels	8
Ranks	1 Rank per Channel
Banks	8 Banks per Rank
Row Buffer Size	8KB (open-page policy)
tCAS-tRCD-tRP-tRAS	7-7-7-28 (memory cycles)
DDR3	Values
Bus Frequency	800 MHz (DDR 1.6 GHz)
Bus Width	64 bits
Channels	4
Ranks	1 Rank per Channel
Banks	8 Banks per Rank
Row Buffer Size	8KB (open-page policy)
tCAS-tRCD-tRP-tRAS	11-11-11-44 (memory cycles)

Table 1: Experimental Parameters

vidual bits validate the corresponding subblock’s residency in NM. When a block is swapped out of NM, the history of bit vectors is stored in a small SRAM structure called a bit vector table. Multiple subblocks are fetched using this bit vector when a block is swapped in again. This exploits spatial locality as previously used subblocks are swapped at the same time, so any subsequent request to either of the subblocks results in a subblock serviced from NM. Since this scheme does not swap any other undesirable subblocks, it is more bandwidth efficient than large block based schemes, which have to swap every subblock within a large block.

An index to access a page is calculated using the modulo operator in order to access the remap entry and data in NM. Also along with LLC access, the PC and the request address are used to access the predictor. The request is sent to NM using the calculated index and predicted way. If the lock bit and the remap field is set and matched, the NM data is fetched. In case of a remap mismatch, the request address is checked whether it falls under the NM address space. If so, the bit vector is checked to determine the location of the requested subblock (If resident in NM, the bit has to be 0). The prediction can skip the previously mentioned metadata fetching steps. If the block address does not fall under NM space, then the remap entry update and subblock swap from FM are initiated. The subblock is swapped to available ways within the set. A similar operation occurs for a remap mismatch. If the request was made to one of locked blocks, the remap entry is checked. If it matches, then the corresponding subblock is fetched from NM. If not, then the subblock is swapped from FM to NM blocks other than this locked block. For every swap from FM operation, the access rate is checked. If it is beyond a threshold, then the swap from FM bypasses NM and becomes a fetch from FM without any metadata update. In the case of the way prediction misspeculation, the remap entry check takes longer as four remap entries are checked in serial.

3. EXPERIMENTAL SETUP

To evaluate the SILC-FM scheme, we use a Pin-based Sniper simulator [2]. We also use a detailed memory simulator, Ramulator [16]. For NM memory, we use HBM generation 2 technology and derived timing parameters from 235A [11, 12] along with other publicly available sources [1,

Figure 1: Performance Comparison with Other Schemes [8, 9, 21]. DDR3 technology is used as FM memory with latency parameters derived from JEDEC and vendor datasheet [10, 20]. Simulation parameters are shown in Table 1. We compare our scheme against other three other die-stacked DRAM designs: Random Static Placement (Rand), HMA [18], CAMEO (CAM). Random uses the entire NM and FM as OS visible address space and maps pages randomly. Thus, this scheme does not consider different bandwidth/latency characteristics of NM and FM, and rather, treats them the same.

4. EVALUATION

Figure 1 shows the performance improvement of our proposed scheme against other schemes. Unless otherwise specified, we use the baseline as the system without NM. First, the Random scheme does not see much significant performance improvement. The placement is done randomly without considering NM and FM characteristics, so pages are statically allocated. The SILC-FM scheme effectively removes conflicts by offering associativity and locked blocks. Unlike HMA, the blocks are locked in NM as soon as the access count reaches the threshold. This makes the hot block capturing ability of our scheme to respond much quickly to changes in the hot working set. The associativity reduces conflicts among those pages that are not locked yet. This additional performance gain makes SILC-FM achieve higher performance than the state-of-the-art scheme, CAMEO. Furthermore, the bypassing feature makes certain workloads such as milc extract performance opportunities by using FM bandwidth, which would have been idle in other schemes due to its high access rate.

5. CONCLUSION

Prior approaches focused on using block or epoch based schemes, but adopting either one will benefit only a subset of different workloads. In this paper, we have presented an associative locking memory architecture called SILC-FM that locks hot pages in NM and intelligently remaps FM subblocks into NM. In the end, SILC-FM is able to achieve on average 36% performance improvement over state-of-the-art die-stacked DRAM architecture. In conclusion, SILC-FM is a novel memory architecture that takes the advantage of the large NM capacity by holding as many hot data as possible in NM.

6. ACKNOWLEDGEMENT

This work was supported in part by National Science Foundation grants 1337393 and 1117895 and AMD. Any opinions, findings, conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of sponsors.

7. REFERENCES

- [1] AMD Working With Hynix For Development of High-Bandwidth 3D Stacked Memory. 2013.
- [2] T. E. Carlson, W. Heirman, and L. Eeckhout. Sniper: Exploring the Level of Abstraction for Scalable and Accurate Parallel Multi-core Simulation. In *Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis, SC '11*, 2011.
- [3] C. Chou, A. Jaleel, and M. K. Qureshi. CAMEO: A Two-Level Memory Organization with Capacity of Main Memory and Flexibility of Hardware-Managed Cache. In *Proceedings of the 47th Annual IEEE/ACM International Symposium on Microarchitecture, MICRO-47*, 2014.
- [4] M. El-Nacouzi, I. Atta, M. Papadopoulou, J. Zebchuk, N. E. Jerger, and A. Moshovos. A dual grain hit-miss detector for large Die-Stacked DRAM caches. In *Design, Automation Test in Europe Conference Exhibition (DATE)*, 2013, pages 89–92, 2013.
- [5] S. Franey and M. Lipasti. Tag tables. In *High Performance Computer Architecture (HPCA)*, 2015 *IEEE 21st International Symposium on*, 2015.
- [6] N. Gulur, M. Mehendale, R. Manikantan, and R. Govindarajan. Bi-Modal DRAM Cache: Improving Hit Rate, Hit Latency and Bandwidth. In *Microarchitecture (MICRO)*, 2014 *47th Annual IEEE/ACM International Symposium on*, 2014.
- [7] F. Hameed, L. Bauer, and J. Henkel. Simultaneously Optimizing DRAM Cache Hit Latency and Miss Rate via Novel Set Mapping Policies. In *Proceedings of the 2013 International Conference on Compilers, Architectures and Synthesis for Embedded Systems, CASES '13*, 2013.
- [8] HMC Consortium. HMC Specification 1.0. 2012.
- [9] M. T. Inc. HMC Generation 2. 2013.
- [10] JEDEC. DDR SDRAM (JESD79-3C). 2008.
- [11] JEDEC. High Bandwidth Memory (HBM) DRAM (JESD235), 2013.
- [12] JEDEC. High Bandwidth Memory (HBM) DRAM Gen 2 (JESD235A). 2016.
- [13] D. Jevdjic, G. Loh, C. Kaynak, and B. Falsafi. Unison Cache: A Scalable and Effective Die-Stacked DRAM Cache. In *Microarchitecture (MICRO)*, 2014 *47th Annual IEEE/ACM International Symposium on*, 2014.
- [14] D. Jevdjic, S. Volos, and B. Falsafi. Die-stacked DRAM Caches for Servers: Hit Ratio, Latency, or Bandwidth? Have It All with Footprint Cache. In *Proceedings of the 40th Annual International Symposium on Computer Architecture, ISCA '13*, 2013.
- [15] X. Jiang, N. Madan, L. Zhao, M. Upton, R. Iyer, S. Makineni, D. Newell, Y. Solihin, and R. Balasubramonian. CHOP: Integrating DRAM Caches for CMP Server Platforms. *IEEE Micro*, 31(1), 2011.
- [16] Y. Kim, W. Yang, and O. Mutlu. Ramulator: A Fast and Extensible DRAM Simulator. *Computer Architecture Letters*, PP(99):1–1, 2015.
- [17] G. Loh and M. D. Hill. Supporting Very Large DRAM Caches with Compound-Access Scheduling and MissMap. *IEEE Micro*, 32(3), May 2012.
- [18] M. Meswani, S. Blagodurov, D. Roberts, J. Slice, M. Ignatowski, and G. Loh. Heterogeneous memory architectures: A HW/SW approach for mixing die-stacked and off-package memories. In *High Performance Computer Architecture (HPCA)*, 2015 *IEEE 21st International Symposium on*, 2015.
- [19] J. Meza, J. Chang, H. Yoon, O. Mutlu, and P. Ranganathan. Enabling Efficient and Scalable Hybrid Memories Using Fine-Granularity DRAM Cache Management. *Computer Architecture Letters*, 11(2):61–64, July 2012.
- [20] Micron Technology Inc. TN-46-03 Calculating Memory System Power for DDR. 2001.
- [21] J. T. Pawlowski. Hybrid Memory Cube (HMC). In *Proceedings of 23rd Hot Chips*, 2011.
- [22] M. K. Qureshi and G. H. Loh. Fundamental Latency Trade-off in Architecting DRAM Caches: Outperforming Impractical SRAM-Tags with a Simple and Practical Design. In *Proceedings of the 2012 45th Annual IEEE/ACM International Symposium on Microarchitecture*, MICRO-45, 2012.
- [23] J. Sim, A. R. Alameldeen, Z. Chishti, C. Wilkerson, and H. Kim. Transparent Hardware Management of Stacked DRAM as Part of Memory. In *Proceedings of the 2014 47th Annual IEEE/ACM International Symposium on Microarchitecture*, MICRO-47, 2014.
- [24] J. Sim, G. H. Loh, H. Kim, M. O'Connor, and M. Thottethodi. A Mostly-Clean DRAM Cache for Effective Hit Speculation and Self-Balancing Dispatch. In *Proceedings of the 2012 45th Annual IEEE/ACM International Symposium on Microarchitecture*, MICRO-45, 2012.
- [25] L. Zhao, R. Iyer, R. Illikkal, and D. Newell. Exploring DRAM cache architectures for CMP server platforms. In *Computer Design, 2007. ICCD 2007. 25th International Conference on*, pages 55–62, 2007.