Theme Article: Monolithic 3-D Integration

Analyzing the Monolithic Integration of a ReRAM-Based Main Memory Into a CPU's Die

Meenatchi Jagasivamani
University of Maryland

Candace Walden
University of Maryland

Devesh Singh
University of Maryland

Luyi Kang
University of Maryland

Shang Li
University of Maryland

Mehdi Asnaashari
Crossbar Incorporated

Sylvain Dubois
Crossbar Incorporated

Bruce Jacob
University of Maryland

Donald Yeung
University of Maryland

***Abstract*—Nonvolatile memory, such as resistive RAM (ReRAM), is compatible with standard CMOS logic processes, allowing a sizable main memory system to be integrated into a CPU's die. ReRAM bitcells are fabricated within *crosspoint subarrays* that leave the bulk of transistors underneath the subarrays vacant. This permits placing the memory system over the CPU, improving area, parallelism, and power. Our work quantifies the impact of integrating ReRAM into a CPU's die. When integrating ReRAM over CPU logic, the best area efficiency occurs when 48% of the die is covered with ReRAM. The CPU's area increases by 18.8%, but we can recoup 35.5% of the die area by utilizing the free transistors underneath the crosspoint subarrays. When integrating ReRAM over CPU cache, up to 85.3% of the cache can be covered with ReRAM. Our work also shows that**

Digital Object Identifier 10.1109/MM.2019.2944335
Date of publication 27 September 2019; date of current version 8 November 2019.

Published by the IEEE Computer Society

on-die ReRAM can support very high bandwidth through massively parallel memory access. At 28 nm, 4–16k independent ReRAM banks could be integrated onto the CPU die, providing 512–1024–GB/s peak bandwidth. At more advanced technology nodes, 5–10 TB/s may be possible.

In the post-Moore era, computer architects will no longer be able to rely on feature size scaling to continue fueling performance and power efficiency gains. Instead, new technological innovations will be needed to replace Moore's Law scaling. A promising direction is to use emerging nonvolatile memories to supplement or even replace DRAM. Several technologies are vying to do so, including resistive RAM (ReRAM), spin-transfer torque magnetic RAM (STT-MRAM), and phase change memory (PCM).

Many researchers have proposed memory systems containing such nonvolatile memories.[1–3] These new memory systems provide higher capacity because the nonvolatile memories are denser than DRAM. There is also a significant power efficiency benefit due to the fact that nonvolatility eliminates the need for refresh, yielding power savings as memory capacity scales. The drawback, though, is that nonvolatile memories have higher latency than DRAM and suffer wearout. A popular approach is to retain a small amount of DRAM to buffer frequently accessed data.

New technological innovations will be needed to replace Moore's Law scaling. A promising direction is to use emerging nonvolatile memories to supplement or even replace DRAM.

All of this prior research, however, has ignored what we believe to be a significant opportunity of emerging nonvolatile memory: its compatibility with CMOS. Whereas DRAM requires special VLSI processes tuned for implementing the DRAM's memory cells, fabricating nonvolatile memory can be done in a standard CMOS logic process. This implies we can integrate nonvolatile memory into a CPU's die, creating a monolithically integrated CPU–main memory chip. Putting the CPU and its memory system on the same die will significantly reduce the energy to access memory, improving power efficiency. It will also allow for an extremely wide connection between the cores and the memory system, which will benefit highly parallel architectures, such as *tiled CPUs*. This will provide much higher throughput and performance for data-intensive computations compared to DRAM, despite the nonvolatile memory's higher latency.

In such monolithically integrated CPU–main memory chips, the entire processor and memory system must fit into a single die. If the two are integrated in a 2-D planar fashion, then the available area can become a limiting factor. Recent nonvolatile memories allow for 3-D stacking of the memory cells to improve density. Examples include Intel's 3D XPoint[4] and Crossbar's 3D ReRAM.[5] In these 3D memory architectures, the memory bitcells are sandwiched in between metal wires—i.e., at the intersection of wires laid out perpendicularly in adjacent VLSI layers—giving rise to a "crosspoint architecture." Rather than isolating individual bitcells using access transistors, isolation in crosspoint subarrays is provided via per-cell "selector devices."

The use of selector devices enables extremely small bitcells that can be stacked vertically across multiple metal layers. But, it also means that the transistors underneath the crosspoint subarrays are free for implementing nonmemory circuits. Some logic is still needed for access circuitry, but the bulk of the transistors are unused. This implies the crosspoint memory can be fabricated over the CPU, occupying the top-level metal layers of the CPU's die. Meanwhile, the CPU's logic can be implemented in the die's logic transistors, minus those needed for the memory access circuits. This can yield high area efficiency.

In this form of monolithic 3-D integration, the logic transistors still exhibit a planar layout. Only the memory bitcells are integrated in a 3-D fashion. In comparison, more aggressive forms of monolithic 3-D integration allow multiple layers of both logic and memory to be stacked vertically within a single die.[6] The benefit of our

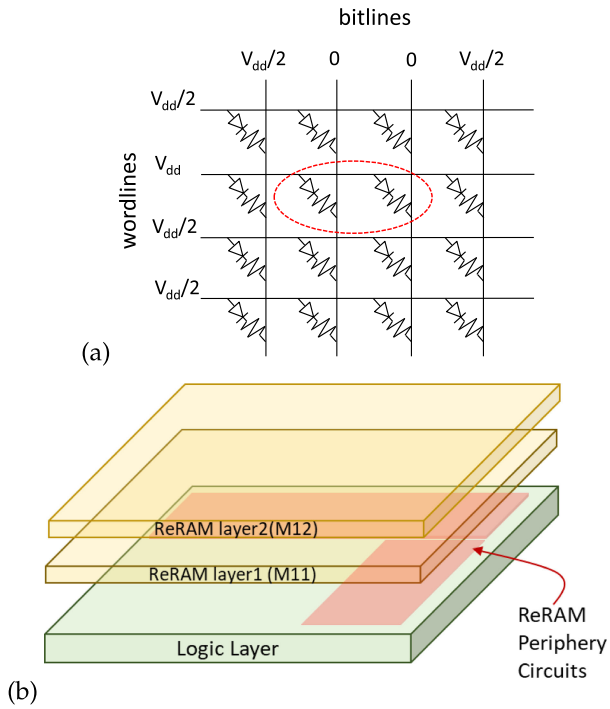

Figure 1. (a). Bitcell activation in crosspoint subarrays. (b). Majority of transistors under the subarray are free for nonmemory circuits.

approach, however, is that it does not require advanced 3-D processing techniques, which to the best of our knowledge, only exist in research fabs. Instead, monolithically integrated CPU–main memory chips that exploit 3-D crosspoint subarrays can be fabricated today in commercial CMOS fabs at the same technology node as the underlying logic.[5]

This article makes the following contributions. We first quantify the impact of integrating crosspoint subarrays into a CPU's die. Our work shows how densely we can populate the CPU die with nonvolatile memory when integrating ReRAM over both CPU logic and cache. Next, we analyze the potential performance benefits of the on-die main memory. In particular, we show the parallelism and latency that monolithically integrated ReRAM could support, and derive the range of memory bandwidths that could be achieved. We also discuss the increased power efficiency that monolithic integration can provide through data movement reduction.

First, the "CPU–ReRAM Integration" section presents our CPU-ReRAM integration study. Next, the "Potential Capabilities" section discusses performance and power efficiency benefits. Last, the "Conclusion" section concludes this article.

CPU–ReRAM INTEGRATION

Crosspoint subarrays employ diode-like selector devices instead of per-cell access transistors. As shown in Figure 1(a), a bitcell is activated when a V_{dd} drop appears across the bitcell and its selector. By setting other wordlines and bitlines to $V_{dd}/2$, selectors for nonactivated bitcells are turned off because of an insufficient voltage drop. Crosspoint subarrays are inherently fined grained; only a small number of cells can be sensed per access due to the limited current that can be sourced down a wordline. This limits the activation to eight bitcells per layer. The bitcell/selector devices can be fabricated in the top metal layers of the CPU die, during back-end-of-line processing steps.[5] As shown in Figure 1(b), multiple layers of bitcells/selectors can be stacked vertically with eight-layer memory possible.[5]

The ReRAM bitcells connect to peripheral access transistors in the lower layers, thus taking up part of the silicon area. Wordline decoders reside on the side of the array, while column-mux circuits reside on the bottom, forming an "L-shape," as shown in Figure 1(b).

The amount of the remaining transistor area depends on the size of the subarray, with larger subarrays yielding more unused area. For subarrays with 2048×2048 bitcells (per layer), we estimate that 26% of the area is utilized by the peripheral access circuits, leaving 74% of the area free for nonmemory circuits. Instead of leaving this area vacant, one possibility is to utilize these free transistors to implement parts of the CPU and achieve higher area efficiency. In the following, we consider integrating the ReRAM over two different structures: CPU random logic and CPU cache.

ReRAM Over Logic

We first integrate the crosspoint subarrays over a simple CPU core. (Although we envision having ReRAM over a parallel architecture, like a tiled CPU, such architectures contain many identical cores distributed across compute tiles. So, the results in this section could be applied across all of the compute tiles within a larger CPU.) The core we use is an in-order VSCALE

processor from the Berkeley RISC-V architecture.[7] (The datapath was increased from 32 to 256 bits in order to provide a more substantial design for our study.) In this section, we omit the processor's caches (see the "ReRAM Over Cache" section). For the ReRAM, we assume that the unit of replication is a tile consisting of 2 × 2 crosspoint subarrays with rotations of 0°, 90°, 180°, and 270°. All four of the L-shaped peripheral circuits are within a tile abut and form a "cross."

We generated a physical layout of this integrated CPU–main memory chip for the 45 nm node. We used Synopsys Design Compiler for the synthesis and Cadence Encounter for the Automatic Place and Route (APR).[8] To mimic the integration constraints presented by the crosspoint subarrays, we indicated two types of blockage layers for the APR step. The first is placement blockage to prevent placing standard cells from the CPU in the ReRAM peripheral circuit areas. And, the second is routing blockage. We blocked metal layers 1–8 underneath the ReRAM bitcells, which we assumed are reserved for the access circuits, but allowed routing through the blocked region using metals 9 and 10. [The ReRAM is assumed to be in metal layers above, as indicated in Figure 1(b).]

Figure 2(a) shows the best integrated CPU–main memory layout from our study. For our small VSCALE processor, we were only able to integrate four tiles into the core. In 45-nm technology, each ReRAM subarray occupies 109 μm × 109 μm. There is 2 MB of ReRAM per layer across the 16 subarrays in Figure 2(a).

Our goal was to find the best intercluster spacing. We iteratively performed the digital implementation to find the smallest floorplan area at different spacings. Figure 3(a) shows the results. At 45 nm, the best intercluster spacing for total area occurs between 100 and 150 μm and results in the ReRAM tiles covering 48% of the entire floorplan. Larger spacing leads to inefficiency from unused synthesized areas while smaller spacing leads to inefficiency from routing congestion.

The other goal of our experiment is to quantify the impact of the integrated ReRAM clusters on the CPU's area. The results for this are shown in Figure 3(b). At the best intercluster spacing between 100 and 150 μm, the area penalty to the CPU layout is at a minimum as well: 18.8%. This CPU area penalty is offset by the area we recoup by utilizing the unused transistors underneath the crosspoint subarrays. In particular, we recover 74% of the area underneath the subarrays which occupy 48% of the entire floorplan—i.e., we recover 35.5% of the total area. So, after paying the 18.8% area penalty, there is a net gain of 16.7%. This gain reflects the improvement that 3-D integration of the ReRAM above the CPU logic provides over an approach, which leaves the free transistors underneath the ReRAM subarrays unused. ReRAM subarrays with smaller

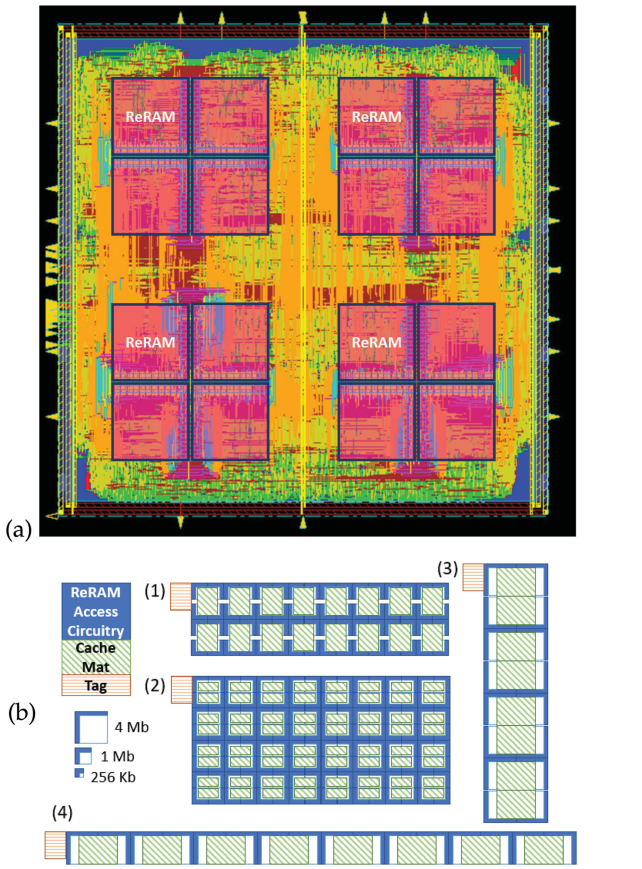

Figure 2. (a) Multiple 2 × 2 ReRAM tiles over a VSCALE processor. Each 2 × 2 ReRAM tile, labeled "ReRAM," is clearly visible along with its access circuitry (the cross within each tile). (b) Different 128-KB cache-ReRAM configurations: (1) two subbanks, eight mats/subbank, four 1-Mb ReRAM subarrays/mat; (2) eight subbanks, eight mats/subbank, eight 1-Mb ReRAM subarrays/mat; (3) eight subbanks, one mat/subbank, two 4-Mb ReRAM subarrays/mat; (4) over cache organized as one subbank, eight mats/subbank, two 4-Mb ReRAM subarrays/mat.

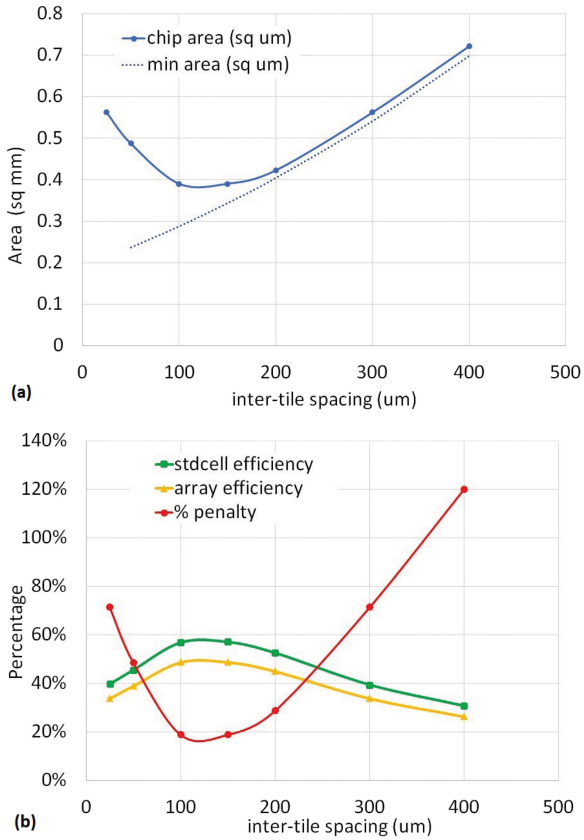

Figure 3. Impact of intercluster spacing on (a) overall area and (b) efficiency.

capacities can significantly reduce the amount of area recovered by this technique as the access circuitry occupies a larger fraction of the area beneath the subarray.

Extrapolating a 48% ReRAM coverage to a large 4-cm^2 CPU die at a 28-nm technology node (the feature size of some recent Crossbar ReRAMs), we could expect to have 40 GB of memory. At more advanced technology nodes, hundreds of gigabytes would be possible.

ReRAM Over Cache

We next consider integrating crosspoint subarrays over CPU cache. Compared to datapath and control logic, caches have a very regular layout and minimal connections between internal cache mats. Consequently, crosspoint subarrays can be integrated over cache mats without incurring a routing-induced area penalty. Rather than just a 48% coverage for CPU logic, in the case of CPU cache, we can pack the crosspoint subarrays much more densely. While previous research has suggested integrating 3-D memory over SRAMs,[9] or integrated them across die stacks,[10] to the best of our knowledge, our work is the first to perform a detailed quantitative study of monolithically integrating crosspoint memory over SRAMs.

Using Cacti 6.5 at the 45-nm technology node, we explored different cache configurations' ability to integrate with crosspoint subarrays. Cacti already splits bitlines to vary the number of subbanks and wordlines to vary the number of mats per subbank. We exploit this inherent configurability to tailor cache mat dimensions so that they just fit under two, or four crosspoint subarrays while needing only minimal routing through the subarray blockages. Since Cacti routes through the center of each mat, cache mats can be placed such that their wiring runs along the subarray boundaries; the intercluster spacing can be set just large enough for the address and data paths to pass through.

The placement of the cache mats under crosspoint subarrays is illustrated in Figure 2(b). Two 2048 $\times$ 2048 ReRAM subarrays can have a 16-KB cache mat integrated underneath them, two 1024 $\times$ 1204 ReRAM subarrays can accommodate a 2-KB cache mat, and an 8-KB cache mat fits beneath four 1024 $\times$ 1204 ReRAM subarrays.

Although integrating ReRAM over cache does not cause routing-induced area penalties, there are nevertheless penalties to the cache as the cache mat size required to fit beneath crosspoint subarrays is smaller than the optimal mat size. The penalties in delay, power, and area for having these nonoptimal cache mat sizes is presented in Table 1.

As Table 1 shows, dividing the cache array only into mats incurs a greater dynamic energy penalty than subbanks (due to needing to activate multiple larger mats to access the entire wordline); dividing only into subbanks results in a lower dynamic read energy. The more divisions present, the higher the standby leakage. To maintain the correct aspect ratio, the ways in a set are split onto multiple wordlines for all but the one subbank case. This does not decrease performance if the tag and data arrays are accessed sequentially, but if not, multiple lines may need to be read to access the way in which the data is stored. The timing penalty will be

Table 1. Cacti design output for a 128-KB cache at 45 nm.

Figure	Configuration	Access time (ns)	Dynamic read energy (nJ)	Dynamic write energy (nJ)	Standby leakage (mW)	Area (mm^2)	ReRAM coverage %
	Ideal cache	0.730	0.041	0.039	149.8	0.488	–
2(b)(1)	2 subbanks, 8 mats/subbank	1.458 +99.7%	0.039 −3.0%	0.052 +33.4%	174.7 +16.6%	0.658 +35.0%	77.2%
2(b)(2)	8 subbanks, 8 mats/subbank	1.674 +129.2%	0.041 +0.2%	0.052 +31.7%	181.8 +21.3%	0.766 +56.9%	85.3%
2(b)(3)	8 subbanks, 1 mat/subbank	1.145 +56.8%	0.031 −22.8%	0.041 +3.5%	165.6 +10.5%	0.703 +44.2%	69.7%
2(b)(4)	1 subbank, 8 mats/ subbank	1.657 +126.9%	0.054 +32.7%	0.063 +60.6 %	165.6 +10.5%	0.547 +12.2%	78.9%

determined by the ReRAM subarray layout more so than the division of the cache arrays as it is dominated by wire delay.

The last column in Table 1 reports the ReRAM coverage. If one is willing to tolerate the higher penalties of eight mat divisions, then 78.9% of the cache area can be covered by ReRAM with 2 MB of cache/GB of ReRAM. Configuration 2 only incorporates 1 MB of cache/GB of ReRAM, affording it 85.3% coverage at the expense of cache capacity. At 28 nm, assuming half a 4-cm^2 CPU die is dedicated to ReRAM and cache, 36 GB of ReRAM can be integrated with 36 MB of cache. At more advanced technology nodes, hundreds of gigabytes of ReRAM over hundreds of megabytes of cache would be possible.

POTENTIAL CAPABILITIES

Monolithic integration of the CPU and main memory affords two major benefits: parallelism and locality. These can be leveraged for increased memory bandwidth and power efficiency.

Memory Bandwidth

Monolithic integration of main memory eliminates the need for memory requests to traverse narrow off-chip channels. Instead, high wire density can be supported from the cores all the way to the memory banks. This means memory bandwidth is not limited by the transfer of data between the memory system and cores. Rather, it is determined by the amount of bank-level parallelism supported in the memory system and the memory's access latency. Given b banks, a fetch width of f, and a latency of l, the peak bandwidth for the memory system is given by

$$\mathrm{BW}_{\mathrm{peak}} = \frac{b \times f}{l}. \quad (1)$$

The parameters b and f are determined by the logical organization of the crosspoint subarrays, which presents a design tradeoff. Multiple subarrays can be logically combined into memory banks to yield larger fetch widths, but this will result in fewer total banks, thereby reducing access parallelism. Independent of how the subarrays are organized, however, the product $b \times f$ is fixed given some total number of on-die subarrays and the amount of data sourced from each subarray, as expressed in

$$b \times f = \frac{A \times c}{a} \times p \times 8. \quad (2)$$

In particular, the former is determined by the total area A, the fraction of that area covered by ReRAM c, and the area for each ReRAM subarray a. The latter is determined by how many ReRAM layers can be accessed in parallel p, and the number of bitcells that can be accessed per layer, which we assume is 8 bits.

Assuming eight layers of ReRAM with $p = 4$, up to 4 B could be accessed from a single eight-stack crosspoint subarray. Based on the study in the "CPU–ReRAM Integration" section, at the 28-nm technology node on a 4-cm^2 CPU die, there could be 8192 subarrays ($b \times f =$ 32 KB) when using 2048 $\times$ 2048 subarrays. For 1024 $\times$ 1204 subarrays, there could be 32 768 subarrays ($b \times f =$ 128 KB). Technology scaling, larger

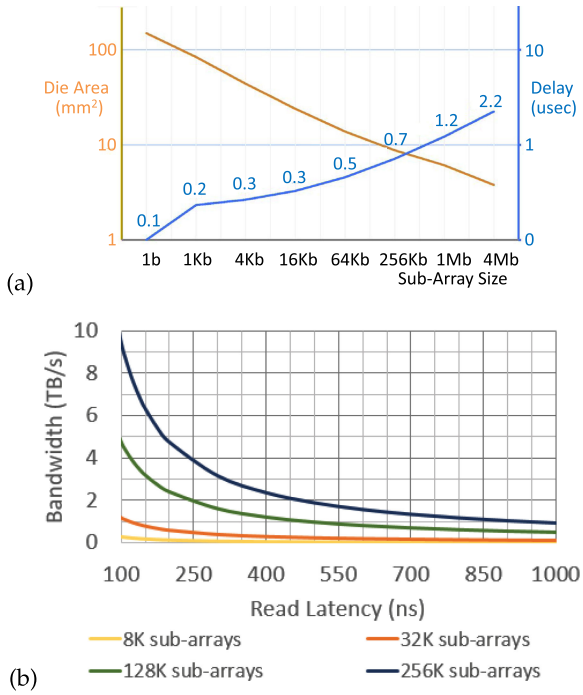

Figure 4. (a) Latency and area for Crossbar 3D ReRAM at 28 nm as a function of the subarray size. (b) Maximum read bandwidth depending on the number of ReRAM banks and the read latency.

dies, or smaller arrays could reasonably increase the total chip-wide access width beyond 262 144 subarrays ($b \times f = 1$ MB).

Besides b and f, (1) also depends on l. ReRAM is still a relatively immature technology. One consequence of this is that current read latencies can exceed 1 μs for large subarrays. (Write latencies are typically double read latencies, but because write latency is easier to tolerate, we focus on reads.) As with other memory technologies, ReRAM delay is wire limited, so latency can be reduced by using smaller subarrays. In fact, latencies near 100 ns are possible, but these subarrays are less dense and have less room to accommodate logic beneath them. Figure 4(a) shows this tradeoff for a 28-nm Crossbar ReRAM. However, these ReRAM memories are currently designed for storage applications; we expect that a redesign of the bitcells and subarrays to optimize them for main memory will permit latencies below 200 ns, even for larger subarrays.

Figure 4(b) shows the memory bandwidth achievable according to (1) depending on the number of subarrays (which can access 4 B) and read latency. As the amount of parallelism increases, the peak bandwidth reaches 1 TB/s at the higher latencies. But for lower latencies, the peak bandwidth can be as high as 5–10 TB/s. Notice that these bandwidths can be achieved at a *fine access granularity* (8 B). Rather than fetching wide sequential blocks, the high memory bandwidth is maintained by servicing lots of narrow memory requests simultaneously. This can deliver high throughput not only for regular computations, but also for computations with irregular memory access patterns.

Energy

Currently, the largest portion of access energy for the memory system is data movement. For example, in HBM2, data movement accounts for approximately 70% of the access energy.[11] A major benefit of monolithically integrated memory is the ability to place the memory system physically close to the cores, minimizing data movement.

In a tiled CPU, some of the ReRAM banks could be co-located with each core, either integrated directly over the core (see the "ReRAM Over Logic" section) or over the core's slice of the LLC (see the "ReRAM Over Cache" section). This would virtually eliminate data movement for memory accesses to those local ReRAM banks. Even if a core accesses data in a remote ReRAM bank on a different compute tile, the memory request would only need to traverse the on-die interconnect. This is still better than discrete memory systems where a memory request must go off-chip and traverse the silicon interposer (e.g., for HBM2) or the system motherboard (e.g., for DDR4). This reduction in data movement can provide very significant gains in power efficiency.

Write Endurance

Wearout is a concern for any nonvolatile memory system, including monolithically integrated ones. ReRAM endurance as high as 10^{12} has been reported.[12] As ReRAM matures, even higher endurance may be possible, but until then, incorporating wear leveling and/or hybrid memory is warranted. For the latter, in-package DRAM such as HBM2 could be an option. We expect on-die ReRAM to provide higher

bandwidth, but frequently written data could be moved to HBM2 DRAM.

Thermal Dissipation

Compared to conventional 3-D integration via die stacking, we do not expect heat dissipation to be as problematic for our approach since we only add metal layers on top of a single CPU die. Most of the heat will be produced by the logic in the die's substrate which is already adjacent to a heat sink. Moreover, the silicon-based switching material used in Crossbar's ReRAM is very stable across a wide temperature range,[5] so the nonvolatile memory does not impose additional heat dissipation constraints.

CONCLUSION

Emerging nonvolatile memories, such as ReRAM, are compatible with standard CMOS logic processes, allowing a sizable main memory system to be integrated into a CPU's die. Moreover, ReRAM bitcells are fabricated within crosspoint subarrays that leave the bulk of logic transistors underneath the subarrays vacant. This permits placing the memory system over the CPU's logic and/or caches.

Emerging nonvolatile memories, such as ReRAM, are compatible with standard CMOS logic processes, allowing a sizable main memory system to be integrated into a CPU's die. More- over, ReRAM bitcells are fabricated within cross-point subarrays that leave the bulk of logic transistors underneath the subarrays vacant. This permits placing the memory system over the CPU's logic and/or caches.

For ReRAM over CPU logic, the best area efficiency occurs when 48% of the die is covered with ReRAM. In this case, the CPU's area increases by 18.8%, but we can recoup 35.5% of the unused area underneath the crosspoint subarrays. When integrating over cache, the ReRAM subarrays can be packed much more densely, but with some area, delay, and energy penalty to the cache.

Monolithically integrated CPU–main memory systems will provide extremely high memory bandwidth and power efficiency. In particular, high bandwidth can be supported due to the large number of parallel banks that can be formed. At 28 nm, 4096-16384 ReRAM banks could be integrated on the CPU die, which could support between 512 and 1024 GB/s. At more advanced technology nodes, upwards of 5–10 TB/s could be possible.

ACKNOWLEDGMENT

This work was supported by the Department of Defense.

REFERENCES

1. S. R. Dulloor *et al.*, "Data tiering in heterogeneous memory systems," in *Proc. 11th Eur. Conf. Comput. Syst.*, 2016, pp. 15:1–15:16.
2. S. Lee, H. Bahn, and S. H. Noh, "CLOCK-DWF: A write-history-aware page replacement algorithm for hybrid PCM and DRAM memory architectures," *IEEE Trans. Comput.*, vol. 63, no. 9, pp. 2187–2200, Sep. 2014.
3. M. K. Qureshi *et al.*, "Scalable high performance main memory system using phase-change memory technology," in *Proc. Int. Symp. Comput. Archit.*, 2009, pp. 24–33.
4. Intel, "Intel Optane Technology," 2017. [Online]. Available: http://www.intel.com/content/www/us/en/architecture-and-technology/intel-optane-technology.html
5. Crossbar, "ReRAM Memory, Crossbar," 2017. [Online]. Available: https://www.crossbar-inc.com/assets/resources/white-papers/Crossbar-ReRAM-Technology.pdf
6. M. M. S. Aly *et al.*, "Energy-efficient abundant-data computing: The N3XT 1,000x," *Computer*, vol. 48, no. 12, pp. 24–33, Dec. 2015.
7. A. Waterman *et al.*, "The RISC-V instruction set manual, Volume I: User-level ISA, Version 2.0," Dept. Elect. Eng. Comput. Sci., Univ. California, Berkeley, Berkeley, CA, USA, Tech. Rep. UCB/EECS-2014-54, 2014. [Online]. Available: http://www2.eecs.berkeley.edu/Pubs/TechRpts/2014/EECS-2014-54.html
8. M. Jagasivamani *et al.*, "Memory-systems challenges in realizing monolithic computers," in *Proc. Int. Symp. Memory Syst.*, 2018, pp. 98–104.
9. S. Mittal *et al.*, "Exploring design space of 3-D NVM and eDRAM caches using DESTINY tool," Oak Ridge Nat. Lab., Oak Ridge, TN, USA, Tech. Rep. ORNL/TM-2014/636, 2015.

10. G. Sun, X. Dong, Y. Xie, J. Li, and Y. Chen, "A novel architecture of the 3-D stacked MRAM L2 cache for CMPs," in *Proc. IEEE 15th Int. Symp. High Perform. Comput. Archit.*, 2009, pp. 239–249.
11. M. O'Connor *et al.*, "Fine-grained DRAM: Energy-efficient DRAM for Extreme Bandwidth Systems," in *Proc. 50th Annu. IEEE/ACM Int. Symp. Microarchit.*, 2017, pp. 41–54.
12. M.-J. Lee *et al.*, "A Fast, High-Endurance and Scalable Non-Volatile Memory Device Made from Asymmetric Ta2O5-x/TaO2-x Bilayer Structures," *Nature Mater.*, vol. 10, no. 8, pp. 625–630, 2011.

Meenatchi Jagasivamani is currently working toward the PhD degree in the Department of Electrical and Computer Engineering, University of Maryland (UMD). Prior to UMD, she worked on a number of emerging technologies at Intel, Micron, and on government research projects with DARPA and IARPA. She has a Master's degree in Electrical Engineering from Virginia Tech. Contact her at: meenatchi@gmail.com.

Candace Walden is currently working toward the PhD degree in the Department of Electrical and Computer Engineering, University of Maryland. Her research focuses on monolithic integration of nonvolatile memories and miss tolerant caches. Contact her at: cbwalden@umd.edu.

Devesh Singh is currently working toward the PhD degree in the Department of Electrical and Computer Engineering, University of Maryland. His research domain is monolithic integration using nonvolatile memory technology. Contact him at: dsingh19@terpmail.umd.edu.

Luyi Kang is currently working toward the PhD degree in the Department of Electrical and Computer Engineering, University of Maryland. His research focuses on nonvolatile memory systems and key-value storage. Contact him at: luyikang@terpmail.umd.edu.

Shang Li is a researcher at Cadence Design Systems. He recently received his PhD degree from the Department of Electrical and Computer Engineering, University of Maryland. Contact him at: shangli@terpmail.umd.edu.

Mehdi Asnaashari has been Chief System Architect at Crossbar Incorporated since 2014. Prior to joining Crossbar, he was with Avalanche Technology, and from 1994 to 2010, he held different positions at Lexar/Micron. Contact him at: mehdi.asnaashari@crossbar-inc.com.

Sylvain Dubois has been the Vice President of Strategic Marketing and Business Development, Crossbar Incorporated, since 2013. Prior to joining Crossbar, he held positions at Spansion and Texas Instruments. Contact him at: sylvain.dubois@crossbar-inc.com.

Bruce Jacob is a full Professor in the Department of Electrical and Computer Engineering, University of Maryland. He has helped Micron design their new Hybrid Memory Cube DRAM architecture; he redesigned Cray's memory controller for their Black Widow memory system; he helped Northrop Grumman design a system interconnect for their experimental ultralow-power datacenter; he designed a high-performance memory system for the 1024-core Teraflux chip funded by the European Commission; and he currently collaborates with researchers at the Department of Energy on the design of their next-generation supercomputers. He is the founder of the annual International Symposium on Memory Systems. Contact him at: blj@umd.edu.

Donald Yeung received the BS degree from Stanford University, and the SM and PhD degrees from the Massachusetts Institute of Technology. He is currently a professor in the Department of Electrical and Computer Engineering, University of Maryland. His research is in the area of parallel computer architecture, focusing on the interaction of parallel architectures with operating systems, compilers, and applications. He has made contributions in the areas of shared memory coherence protocols, helper threads, approximate computing, and reuse distance models for multicore processors. Contact him at: yeung@umd.edu.