Evaluation of Emerging TSV-enabled Main Memories on the PARSEC Benchmark

[1]Rodrigo Cataldo, [1]Guilherme Korol, [1]Ramon Fernandes, [1]Gustavo Sanchez, [2]Debora Matos, [1]César Marcon

[1]Pontifícia Universidade Católica do Rio Grande do Sul (PUCRS), Porto Alegre, Brazil
[2]Universidade Estadual do Rio Grande do Sul (UERGS), Porto Alegre, Brazil
{rodrigo.cataldo, guilherme.korol, gustavo.sanchez, ramon.fernandes}@acad.pucrs.br; debora.motta@gmail.com; cesar.marcon@pucrs.br

Abstract—**This paper evaluates emerging TSV-interconnected memory technologies employed as the main memory of 3D Symmetric MultiProcessing (SMP). As the target architecture, we implemented a typical 3D SMP including L1 and L2 caches together with some tiers of main memory. Besides, we employed DDR3 as a baseline comparison to normalize all results. The experimental results show a tradeoff between energy efficiency and execution time when using six memory technologies executing a set of applications of PARSEC benchmark on Gem5. All evaluated memories reveal significant reductions in energy consumption with some penalization on the execution time. Additionally, HBM shows to be the most promising one, reducing more than five times the energy consumption and giving a small performance boost than DDR3, in general.**

Keywords—Memory technologies; Execution time minimization; Energy Consumption Efficiency; 3D circuits

I. Introduction

Through-Silicon Via (TSV) is at the center of one of the most significant changes to the memory interface and, consequently, to the Memory Wall [1], which is the observation of the increasingly processor/memory performance gap in at least the last 20 years. On top of that, the trend of placing more and more cores on a single chip exacerbates this gap. One way to diminish this is to increase memory bandwidth through wider memory interfaces. However, until now this has been very challenging because wider interfaces imply more off-chip pins and such pins are very expensive [2]. TSV eliminates such limitation by stacking dies and incorporating main memory into the chip. Hence, no extra off-chip pins are necessary.

AMD and Hynix produce High Bandwidth Memory (HBM), which is a technology that exploits an enormous number of signals available with die-stacking to provide very high memory bandwidth. Each HBM stack accommodates eight independent memory channels, and each channel follows the traditional Double Data Rate (DDR) memory interface with power saving methods from Low Power DDR (LPDDR). HBM achieves better power saving using a substantially smaller form factor and lower operating voltages than traditional DDR [3]. Wide I/O is an interface standard that maximizes the memory bandwidth at the lowest possible power dissipation. The key idea is to stack multiple memory channels interconnected through TSVs on top of the system. Recently, JEDEC (a global leader in developing open standards for the microelectronics industry) published the second standard of Wide I/O that presents significant improvements [4]. Wide I/O maintains the same memory bandwidth at half of the LPDDR3 power dissipation. Increasing the memory frequency, Wide I/O effectively provides more than double the baseline LPDDR bandwidth [5]. The first Wide I/O version has four channels of 128-bus width. The second version doubled the number of channels and halved their width. Both versions can operate at 200 and 266 MHz.

Hybrid Memory Cube (HMC) is another memory solution that relies on TSV. While Wide I/O aims at the mobile low-power market, HMC aims at the high-performance server market. HMC achieves up to 15 times the bandwidth and 70% less energy consumption when compared to traditional DDR3 technology [6]. In HMC, four to eight stacks of DRAM are on top of a single logic chip responsible for data access. The HMC Consortium develops the HMC interface specification and promotes integration into a wide variety of systems. HMC utilizes closed-page policy, and its DRAM devices are redesigned to have short rows (256 bytes instead of 8-16 KiB in a typical DDR3 device) for high-performance computing and server workloads that exhibit little spatial and/or temporal locality on the memory address accesses [6].

There have been some studies of the impact of TSV-enabled memory [6][7][8], generally targeting a single memory with multiple configurations. Azarkhish et al. [6] propose an extension to the standard HMC that supports near memory computation at the logic base. Employing trace-based simulation, they showed that this extension could meet the demands of current HMC designs. The authors also note that PARSEC, and others current multi-threaded workloads, cannot easily utilize the vast bandwidth potential provided by HMC, mainly due to the overheads of the cache coherence mechanisms. Rosenfeld [7] analyzes several parameters, performance characteristics, and tradeoffs in the HMC architecture. Using MARSSx86, a full system simulator, the author concludes that recent multiprocessor architectures pose a challenge for a high-throughput HMC since it restricts the memory bandwidth to the cache-coherent interconnect bandwidth. Woo et al. [8] propose to redesign the traditional L2 and DRAM interface for exploiting the advantages of TSV integration. The idea is to leverage the TSV bandwidth to hide latency behind enormous data transfers. To tackle the Memory Wall problem, the authors prefetch entire memory pages (4 KiB) instead of a usual 64-byte cache line. Using SESC simulator and three benchmark suites, the authors proved the advantages of their proposition.

978-1-5090-6113-6/16/$31.00 ©2016 IEEE

Authorized licensed use limited to: Pontificia Universidade Catolica do Rio Grande do Sul (PUC/RS). Downloaded on November 11,2022 at 14:53:19 UTC from IEEE Xplore. Restrictions apply.

Given the inherent difficulty of writing programs to run efficiently on parallel systems, one feature often found is the ability to address the entire physical memory space as a single entity. Thus, the programmer does not need to concern itself with the data placement, because all variables are accessible at any time to any processor. This type of system is named Symmetric MultiProcessing (SMP). When the physical address is a unique entity, the hardware typically implements cache coherence to provide a consistent view of the memory subsystem. However, such consistency diminishes the achievable memory bandwidth due to the inherent requirements of such protocol [7]. Hence, the contribution of this paper is the analysis of how emergent TSV-interconnected memories behave under a diverse set of real-world applications running on an 8-core SMP. Thus, we employed the Gem5 full system simulator, a widely applied framework for architecture exploration [9], and the PARSEC benchmark suite. To the best of the authors' knowledge, there has not been a study of multiple emergent memories on the same system and workload.

The remainder of this work is divided as follows. Section II describes the most important aspects of the target architecture employed here. Section III details the method used for extracting the experimental results. Section IV reports and discusses the experiments. Finally, Section V presents the major findings and implications of this study.

II. Target Architecture

Figure 1 shows the SMP target architecture employed in here that covers 8 homogeneous 1 GHz ARM-v7a processors capable of executing Out-Of-Order (O3) instructions on a seven-stage pipeline. Each processor has access to a 32 KiB instruction cache and a 32 KiB data cache. The eight processors have access to a single shared L2 cache of 1 MiB size. The main memory is implemented through several tiers connected through TSV. According to each experimental results, the main memory is carried out with DDR3, LPDDR3, Wide I/O – v1, Wide I/O – v2, HBM or HMC.

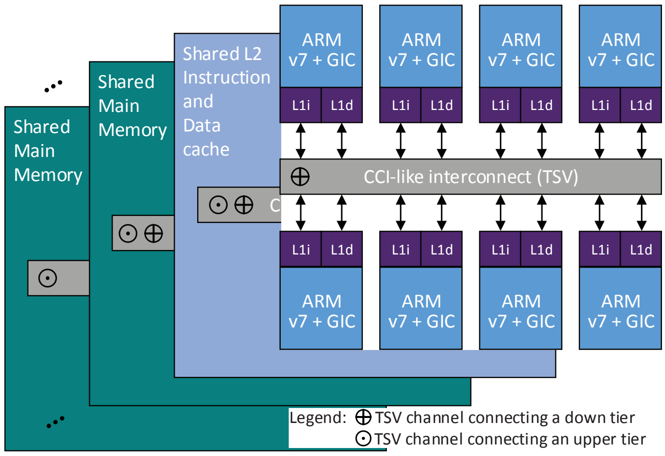
Shared Main Memory
Shared Main Memory
Shared L2 Instruction and Data cache
ARM v7 + GIC
L1i
L1d
CCI-like interconnect (TSV)
Legend: ⊕ TSV channel connecting a down tier
⊙ TSV channel connecting an upper tier

Figure 1. Target architecture comprising eight ARM-v7 processors in an SMP architecture. TSVs are used to connect memory models of other 3D tiers.

Cache coherence is provided via hardware through a Cache Coherent Interconnect (CCI), which is a low-cost and low-power crossbar communication architecture [10]. All read and write data channels are of fixed 64-bit width. The General Interrupt Controller (GIC) is responsible for distributing interrupts for all processors. Since there is no processor distinction for accessing the main memory, this denote a Uniform Memory Access (UMA) architecture.

III. Methodology

All experiments were performed using Gem5 with the most detailed mode of execution – the O3 CPU model – to provide an accurate characterization of the target application. We selected the PARSEC benchmark suite as target application due to its broad range of application domains, parallel techniques and emerging workloads [11]. Table 1 highlights the subset of PARSEC used here.

Table 1. The eight applications of PARSEC benchmark suite employed in this work and its characteristics [11].

Program	Application domain	Parallelization		Working set	Data usage	
		Model	Granularity		Sharing	Exchange
Blackscholes	Financial analysis	Data-parallel	Coarse	Small	Low	Low
Bodytrack	Computer vision	Data-parallel	Medium	Medium	High	Medium
Canneal	Engineering	Unstructured	Fine	Unbounded	High	High
Dedup	Enterprise storage	Pipeline	Medium	Unbounded	High	High
Fluidanimate	Animation	Data-parallel	Fine	Large	Low	Medium
Swaptions	Financial analysis	Data-parallel	Coarse	Medium	Low	Low
Vips	Media processing	Data-parallel	Coarse	Medium	Low	Medium
X264	Media processing	Pipeline	Coarse	Medium	High	High

This subset gives us a diverse application domain and all combinations of parallelization model, working set size and data usage. We simulated the entire application time using the largest input intended for simulators available (*simlarge*). All applications were compiled with the GNU Compiler Collection using the O3 optimization flag. Moreover, every application was executed three times, and their average results were normalized. The Linux Kernel 3.03-rc3 was compiled, and a suitable filesystem was built to execute PARSEC. The effects of mapping techniques are outside of this work. They are complementary. As such, the standard Linux Kernel scheduler, Completely Fair Scheduler, was used.

Table 2 depicts the most important parameters of the six memories analyzed in this work. The values were extracted from datasheet documents provided by industry products and JEDEC standard documents [12].

Table 2. The six memories analyzed in this work and its characteristics.

Memory	Clock	Bus width	Channels	*tRCD*	*tRP*	tCL	tRAS
DDR3	800 MHz	64 bits	2	13.8 ns	13.8 ns	13.7 ns	35 ns
LPDDR3	800 MHz	32 bits	2	18 ns	18 ns	15 ns	42 ns
Wide I/O – v1	200 MHz	128 bits	4	18 ns	18 ns	18 ns	42 ns
Wide I/O – v2	266 MHz	64 bits	8	18 ns	21 ns	18 ns	42 ns
HBM	500 MHz	128 bits	8	15 ns	15 ns	15 ns	33 ns
HMC	1250 MHz	32 bits	16	10.2 ns	7.7 ns	9.9 ns	21.6 ns

Legend: tRCD - Row Address to Column Address Delay; tCL - Column Address Latency; tRAS - Row Active Time; tRP - Row Pre-charge Time.

IV. Experimental Results

Figure 2 displays the experimental results of the execution time for PARSEC benchmark on the six target architectures, which differs only the main memory technology. All results were normalized according to the execution time of the target architecture with main memory implemented with DDR3.

Authorized licensed use limited to: Pontificia Universidade Catolica do Rio Grande do Sul (PUC/RS). Downloaded on November 11,2022 at 14:53:19 UTC from IEEE Xplore. Restrictions apply.

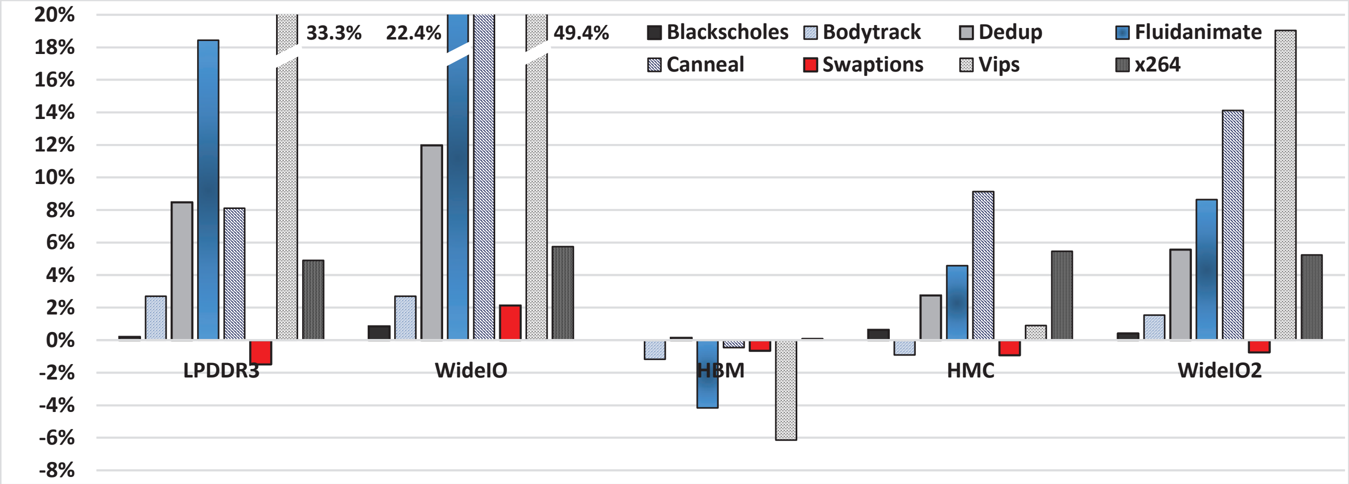

Figure 2. The execution time of eight applications versus six memory technologies. All values are relatively normalized according to the DDR3; i.e., for all applications, the execution time of DDR3 is 0% and the remaining values are perceptual deviations of this reference.

The experiments show that both Wide I/O versions have the least optimized execution time due to its overall low frequency. However, the results show that the execution time is still competitive even when compared to desktop DDR3 – the execution time for five benchmarks were less than 20%. Combining these results with the fact that Wide I/O dissipates nearly half the power of LPDDR3, this is a remarkable memory for MPSoC. Wide I/O achieves this performance by providing double (v1) and quadruple (v2) memory channels available to DDR3 and LPDDR3. Canneal, Dedup, Fluidanimate, and Vips are the PARSEC's applications with significant performance impact (i.e. more than 10%).

LPDDR is the traditional memory technology employed in low-power embedded systems such as mobile cellphones. It uses power saving techniques to reduce its overall power dissipation – for instance, a desktop DDR3-1333 consumes 39 pJ/bit while LPDDR3 consumes 9.2 pJ/bit. Although this work uses the same baseline clock for both, in LPDDR3 the tCL, tRCD, and tRAS timing parameters are approximately 15% slower than its DDR3 counterpart does; therefore, its memory bank requires more operating time. The results display that the effects of low-power for six applications are restrained – the execution time increases less than 10% of the normalized runtime. Once again, Vips and Fluidanimate are the most sensible benchmark due to theirs application nature. Swaptions is the only application that has better execution time than on DDR3.

HMC technology serializes the IO pins of the parallel communication requiring a serial and high-speed transceiver technology. As such, HMC uses only 10% of the pin counts of DDR3 (power and ground are not counted) and consumes 16 pJ/bit. The HMC model on Gem5 is based on the works of Azarkhish et al. [6] and uses a 32-bit bus width per memory channel. The results show that HMC performs worse than DDR3 even considering its higher frequency because of two HMC characteristics: (i) HMC uses closed-page policy to handle memory pages recently accessed. As such, it punishes applications that have good spatial/temporal locality because they will need additional latency to open an already accessed page. For this reason, DDR memories have traditionally employed open-page policies, and applications, such as those present in the PARSEC suite, are developed to exploit this. (ii) The narrow bus width increases the congestion of memory requests from the application. We will show that more than half of the HMC response time is attributed to congestion on the external memory bus for the Dedup application. Azarkish et al. [6] and Rosenfeld [7] also observed this underwhelming performance of HMC.

HBM continues the trend of using parallel communication and doubling the number of data and address pins employing a 1024-bit bus width across eight memory channels. HBM consumes approximately 6 to 7 pJ/bit and presents a counterpoint to the performance of the HMC for the analyzed workload. HBM has the best overall performance for our work set, albeit a restricted improvement from DDR3. For Canneal and Vips, two applications that share a profile of low-speedup and high L2 miss rates, HBM reduced the execution time by 0.5% and 6.1% compared to DDR3, respectively. Furthermore, Fluidanimate also had a decrease in the execution time of 4.2%. Only two applications (Bodytrack and x264) had an increase of execution time, which is insignificant (less than 0.1%).

Figure 3 depicts the average latency of the DRAM controller and L2 miss handler for the Dedup application, which has 6% up to 10% L2 miss rates across the analyzed memory technologies. The average latency of the DRAM controller is the arithmetic mean of all memory channels. This latency is further broken down into the following components: ***queue***, ***bus***, ***bank*** and ***others*** [13]. The ***queue*** latency is the experienced delay for servicing each DRAM burst. As the device bus width determines the DRAM burst, increasing the device bus width in lower queue delay, as there are fewer DRAM bursts to be executed. The ***bus*** latency is the time required to expedite a memory packet, which does not include any outside contention. The ***bank*** latency is the time spent to execute all operations related to DRAM banking. The ***others*** latency describe the time spent outside of the DRAM controller – mainly on-chip congestion. The absence of the

Authorized licensed use limited to: Pontificia Universidade Catolica do Rio Grande do Sul (PUC/RS). Downloaded on November 11,2022 at 14:53:19 UTC from IEEE Xplore. Restrictions apply.

others does not mean that is not present – it means that it did not contribute significantly to the overall average latency. The error bars in Figure 3 represent the lowest and highest average value encountered for L2 miss rates.

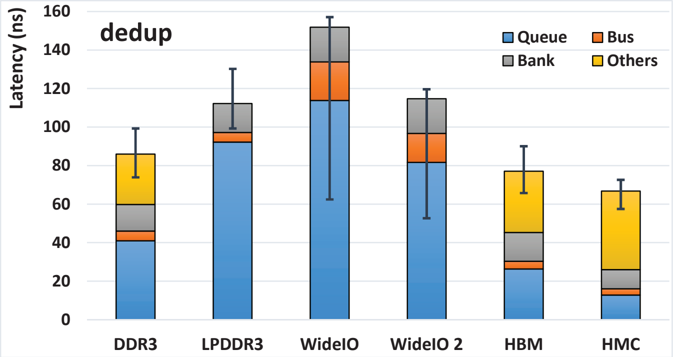

Figure 3. Average of DRAM and L2 latency for the Dedup application. Error bars represent the lowest and highest average values encountered.

From this set of results, we can see that ***queue*** delay is responsible for at least half of the DRAM latency in all cases (excluding external congestion). Three complementary ways to decrease this delay are achieved increasing the (i) operational frequency, (ii) memory channels, and (iii) bus width. The memory technologies analyzed in this work employ a selected number of these ways to achieve a balance between performance and energy consumption.

DDR3 and LPDDR3 show the effect of using 64- and 32-bit bus width, respectively. LPDDR3 has approximately 15% slower bank operations than DDR3. Overall, the LPDDR3 has higher latency penalty in all cases. HBM and HMC show the effects of increasing operational frequency and using a large number of memory channels. Conversely, HMC uses a low-bit bus width, which results in higher on-chip congestion due to longer transmission of data from DRAM to L2. HBM and HMC have lower average latency when compared to DDR3, even though both consume less energy. However, they require 3D chips. Wide I/O version 1 shows the worst average latency and version 2 shows a very similar performance when compared to LPDDR3. We highlight that it is possible to avoid bank operations if a read operation finds the data in the write buffer because Gem5's memory controller buffers the write operations and drains them at a later point.

Finally, Figure 4 shows the average of energy consumption of all memories used on the target architecture when running the PARSEC applications. The energy consumed by each memory is normalized according to the DDR3 consumption.

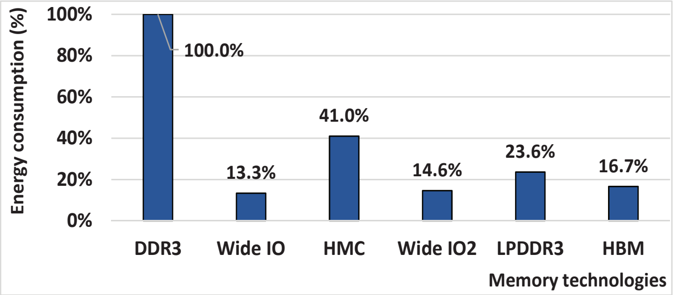

Figure 4. Average energy consumption of all evaluated memories during PARSEC execution.

Here, it is evident the energy consumption gains of emergent memories. Besides, this figure, analyzed jointly with Figure 2, emphasizes the gains of HBM over other solutions.

V. Conclusions

We analyzed the execution of six TSV-enabled memory technologies through a broad range of applications from the PARSEC benchmark. We demonstrated that this set of applications ranged from -6.1% to 49.4% of the performance of a standard desktop DDR3. Moreover, not all memory technologies studied here are intended to give better performance when compared to DDR3. Clearly, Wide I/O is designed for ultralow-power bandwidth as it operates at lower frequencies. Additionally, HBM achieved not only low power results but also an execution time which is competitive with DDR3. It is important to consider that this range of performance variance may be more or less effective depending on the type of applications running on the system. For applications that execute intensively for only a few seconds, this variance can be negligible. However, for real-time applications with strict deadlines or even applications such as media encoding, this variance is critical. Therefore, the system nature defines the impact of adopting these new types of memories.

References

[1] W. Wulf; S. McKee, "Hitting the Memory Wall: Implications of the Obvious". *ACM Special Interest Group on Computer Architecture (SIGARCH)*, vol. 23, no. 1, pp 20-24, 1995.

[2] W. Fu; L. Liu; T. Chen, "Direct distributed memory access for CMPs". *Journal of Parallel and Distributed Computing*, vol. 74, no. 2, pp. 2109-2122, 2014.

[3] S. Wasson (2015, May 19). *AMD's high-bandwidth memory explained - Inside the next generation of graphics memory* [Online]. Available: http://techreport.com/review/28294/amd-high-bandwidth-memory-explained.

[4] JEDEC (2014, September 8). *JEDEC Publishes Wide I/O 2 Mobile DRAM Standard* [Online]. Available: http:///www.jedec.org/news/pressreleases/ jedec-publishes-wide-io-2-mobile-dram-standard.

[5] G. Kimmich, "3D – What's Next". *D43D Workshop*, pp 1-23, 2013.

[6] E. Azarkhish; D. Rossi; I. Loi et al., "A Logic-base Interconnect for Supporting Near Memory Computation in the Hybrid Memory Cube". *Workshop on Near-Data Processing (WoNDP, MICRO)*, pp 1-6. 2014.

[7] P. Rosenfeld, "Performance Exploration of the Hybrid Memory Cube". Ph.D. dissertation, Dept. Elect. Eng., University of Maryland, 2014.

[8] D. Woo; N. Seong; D. Lewis et al., "An Optimized 3D-Stacked Memory Architecture by Exploiting Excessive, High-Density TSV Bandwidth". *International Symposium on High Performance Computer Architecture (HPCA)*, pp 1-12, 2010.

[9] N. Binkert; B. Beckmann; G. Black et al., "The gem5 Simulator". *ACM SIGARCH Computer Architecture News*, vol. 39, no. 2, pp. 1-7, 2011.

[10] U. Wiener, "Modeling and Analysis of a Cache Coherent Interconnect". M.S. Thesis, Dept. Elect. Eng., Eindhoven University of Technology, 2012.

[11] C. Bienia; S. Kumar; J. Singh et al., "The PARSEC Benchmark Suite: Characterization and Architectural Implications". *International Conference on Parallel Architectures and Compilation Techniques (PACT)*, pp. 72-81, 2008.

[12] *JEDEC Standard High Bandwidth Memory (HBM) DRAM, JESD235A*, 2015.

[13] A. Hansson; N. Agarwal; A, Kolli et al., "Simulating DRAM controllers for future system architecture exploration". *International Symposium on Performance Analysis of Systems and Software (ISPASS)*, pp 1-10, 2014.

Authorized licensed use limited to: Pontificia Universidade Catolica do Rio Grande do Sul (PUC/RS). Downloaded on November 11,2022 at 14:53:19 UTC from IEEE Xplore. Restrictions apply.