A Shared Polyhedral Cache for 3D Wide-I/O Multi-core Computing Platforms

Mihai Lefter George Razvan Voicu Sorin Dan Cotofana
Faculty of EE, Mathematics and CS, Computer Engineering Laboratory
Delft University of Technology, Delft, The Netherlands
{m.lefter, g.r.voicu, s.d.cotofana}@tudelft.nl

***Abstract*—3D-Stacked IC (3D-SIC) based on Through-Silicon-Vias (TSV) is an emerging technology that enables heterogeneous integration and high bandwidth low latency interconnection. In this paper we propose a 3D novel cache architecture that leverages a wide TSV-based data link distributed on the entire memory array to support two orthogonal interfaces: (i) a vertical one, with a large data width, and, (ii) a side one, with a lower data width, but with more bank-type access ports, which reduces the bank conflict probability. Our simulations indicate that our proposal substantially outperforms planar counterparts in terms of access time, energy, and footprint while providing high bandwidth, low bank conflict rate, and an enriched access mechanism set.**

I. Introduction

Computing systems are generally utilizing a hierarchical memory organization (depicted in the middle of Fig. 1), which strives to simultaneously approach the performance of the fastest component, the cost per bit of the cheapest component, and the energy consumption of the most energy-efficient component [1]. Faster, but small memories implemented using a costly technology are present at the top of the hierarchy, closer to the processing units, while slower, but larger inexpensive memories are present at the lower levels.

The interconnection data width between two adjacent levels also determines various trade-offs between performance, energy consumption, and cost. Traditionally, the Last Level Cache (LLC) is linked with a larger width to the upper level, and with a narrower width to the lower level, i.e., main memory. The narrower memory side interconnection widths are due to physical constraints, e.g., limited package I/O pins. With the advent of 3D-Stacked IC (3D-SIC) based on Through-Silicon-Vias (TSV) technology, stacking various memory levels on top of each other enables wider interconnection (wide-I/O) links [2], [3] (see Fig. 1 left).

To better exploit the thread-level parallelism in multi-core architectures, the lower level shared caches are usually internally organized in banks, that can be accessed in parallel [4]. Thus, it is essentially a multi-port structure which can simultaneously serve multiple requests as long as the addresses point to different banks, i.e., there are no bank conflicts. For a constant number of access ports, the more banks there are the lower the conflict probability is [5]. However, each bank requires its own set of access lines, which negatively affects the area, and subsequently the access time and energy consumption [6]. It has been proven though in [7] that 3D memories can in certain limits alleviate some of this issues due to their reduced footprint (see Fig. 1 right).

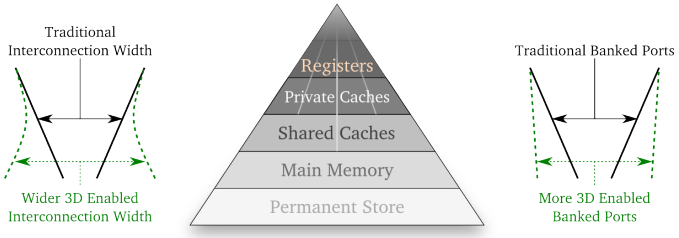

Fig. 1: Memory Hierarchy.

In this paper, we propose a novel shared cache design for TSV-based wide-I/O multi-core systems with reduced conflict probability. The proposed cache consists of multiple identical banks, stacked on top of each other, forming a polyhedral structure. The interface with the upper level caches (cpu-side) is realized horizontally through one access port on every die, while the wide-I/O interface with the lower levels (memory-side) is realized vertically through TSV bundles that traverse the entire IC stack. The TSV bundles are distributed over the entire cache footprint and selectively connected at run-time to the internal data lines in every sub-bank. In addition to supporting the wide-I/O interface, the TSV bundles facilitate horizontal port access to/from a non-local bank (located on a different die) and wide internal cache data transfers. The identical layout structure of all cache dies reduces the manufacturing cost, while the wide-I/O interface increases the communication bandwidth with the lower levels cache or the main memory and provides the means to effectively transfer large amount of data between different banks. In addition, our polyhedral cache has a reduced rate of bank conflicts, due to the creation of a number of virtual banked ports, greater than the number of physical banks in the cache.

To assess the practical implications of our proposal, we compare our novel 3D polyhedral cache against traditional banked multi-port 2D implementations, considering relevant metrics: (i) access time, (ii) footprint, and (iii) leakage and dynamic energy consumption per access cycle. Our results indicate that the 3D polyhedral cache access time is in general smaller than the one of their planar counterparts, with up to 50% reduction in the best case. The inherent footprint reduction directly influences the active energy consumption which in same cases almost reaches the theoretical maximum of, e.g., $8\times$, for an 8-die 8 MB cache. Trade-offs between the wide-I/O interface width and all the other metrics exist: Large widths provide high bandwidth and conflict probability reduction, but increased sub-bank numbers translates into more

978-1-4799-8391-9/15/$31.00 ©2015 IEEE

TSVs and extra decoding logic and wires, which in turn may negatively impact the access time, footprint, and energy cost.

The remainder of the paper is organized as follows. We detail in Section II the proposed polyhedral cache internal structure and access scenarios. In Section III we present an experimental design space exploration for our proposed cache, considering access time, footprint, and energy consumption as relevant metrics. Finally, Section IV concludes this paper.

II. Wide-I/O Shared Polyhedral Cache

In this section we introduce the wide-I/O shared polyhedral cache architecture. However, before detailing our proposal, we review for the sake of clarity the required terminology and traditional memory organization layout.

A. Traditional Cache Design

We consider as a discussion vehicle the banked multiport cache design presented in Fig. 2, consisting of two separate memory arrays, i.e., tag array and data array, a cache controller, and the required data/addresses crossbar switches. The tag array stores the memory address of the cached blocks, while the data array stores the actual data blocks. Considering a number of input ports and a different/same number of banks, the controller detects the desired bank accesses, arbitrates eventual bank conflicts, and generates the required crossbar switches selection signals.

For both arrays we consider the representative layout employed by the Cacti [8] cache simulator, with a zoom-in into the design abstractions presented in the figure. At the highest level the address space is split across identical banks, four in this example, with each bank having its own address and data bus, thus allowing for concurrent bank accesses. Each bank is composed of identical sub-banks, again four in this example, with only one being active per access. Further, each sub-bank is partitioned into multiple mats that simultaneously provide parts of the required data (cache block in a cache data array). Finally, each mat is composed of four identical sub-arrays that share predecoding/decoding logic and peripheral circuitry, and which again deliver together the requested data. An H-tree routing distribution network is used to drive addresses and data to/from the banks, and also to/from every mat inside a bank.

B. Polyhedral Cache Design

In the following we detail the organization of the proposed 3D polyhedral cache. Even though in this work we focus mainly on the cache memory arrays, it is worth noting that the crossbar switches and the controller can also be implemented in 3D technology (either centralized on a separate die, or distributed across several dies).

Our novel cache structure relies on the 3D-stacked memory design presented in Fig. 3 for both tag and data arrays, where every silicon die is an individually addressable memory bank. Therefore, all addresses, as well as cpu-side data, are routed locally on each die (bank). The memory-side data interface consists of TSV bundles, which traverse all vertically stacked

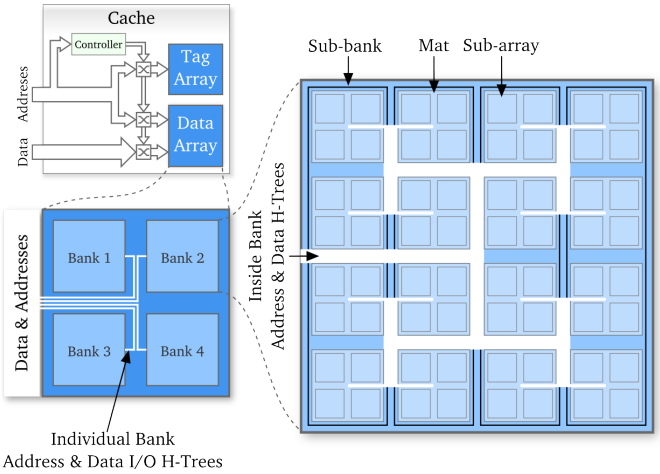

Fig. 2: Traditional Cache and Memory Array Layout.

dies through each mat center, connected to the I/O data lines of each sub-array. This creates a vertical memory-side wide-I/O interface, having a width equal with the cache block size multiplied by the number of sub-banks. We note that horizontal and vertical data flows can coexist within the polyhedral memory array as long as they do not conflict on TSV bundles and/or sub-banks.

Furthermore, by modifying the sub-array input/output data bit routing logic according to Fig. 4, we enhance the versatility of the memory array access mechanism such that it allows the following operations:

1) **Local** cpu-side data access when the issue and the storage dies are identical.
2) **Remote** cpu-side data access when the issue die is different from the storage die. In this case both dies must be accessed with the same address, and the TSVs must be connected to the sub-arrays in a complimentary manner: on one die to the sub-arrays inputs, while on the other die one to the sub-arrays outputs.
3) **Wide-I/O** memory-side access performed by simply selecting on which bank, i.e., on which die, to broadcast the address and to link the sub-arrays data I/O to the TSVs.
4) **Inter-die wide transfers** when large data blocks need to be copied from one die (read die) to another die (write die), accomplished by broadcasting the required addresses on both dies. For sub-banks stacked directly on top of each other this is similar to a remote access, with the main difference that data H-tree is not utilized at all, and can be utilized for other non-Wide-I/O accesses. Transfers between different sub-banks however require the data H-tree to be utilized.

A maximum of five extra wires are required in the H-tree as control signals to support the above mentioned access policies, as indicated by Table I. In order to maintain a low area overhead, we repurpose the already present sub-array output driver as a signal buffer before or after the TSV to diminish its high capacitive load parasitic effect.

In contrast with a traditional banked memory array layout

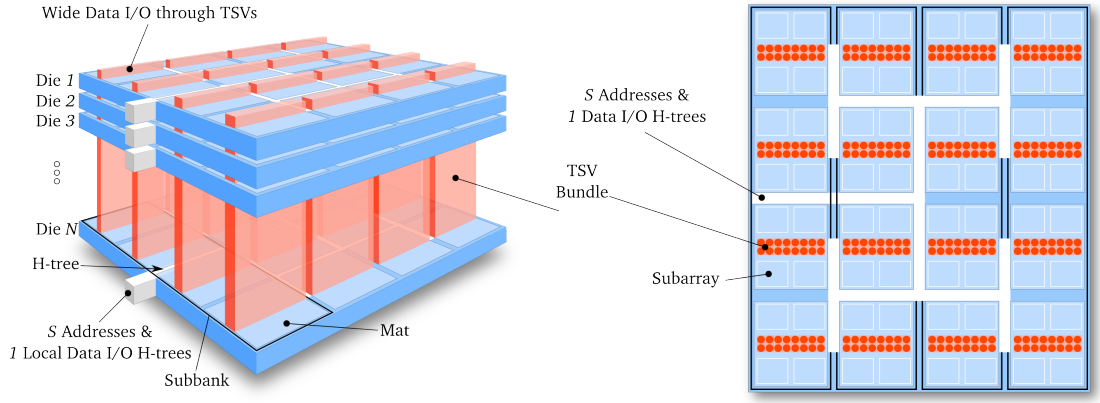

Fig. 3: Polyhedral Memory Array.

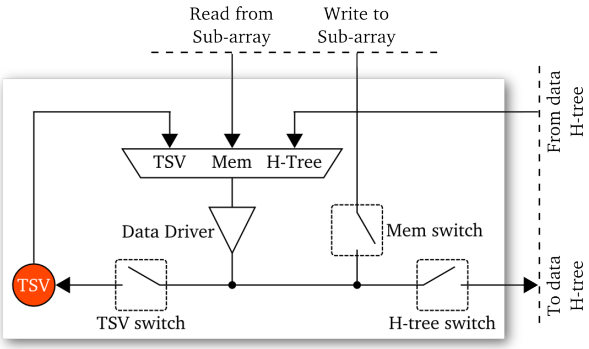

Fig. 4: Required TSV Access Logic.

TABLE I: TSV Access Control Signals.

Access type		Die	MUX selection	Switches TSV	Mem	H-tree
Local	Read		Mem	OFF	OFF	ON
	Write		H-tree	OFF	ON	OFF
Remote	Read	Issue	TSV	OFF	OFF	ON
		Storage	Mem	ON	OFF	OFF
	Write	Issue	H-tree	ON	OFF	OFF
		Storage	TSV	OFF	ON	OFF
Wide-I/O	Read		Mem	ON	OFF	OFF
	Write		TSV	OFF	ON	OFF
Inter-die		Read	Mem	ON	OFF	OFF
		Write	TSV	OFF	ON	OFF

we choose to extend the number of address H-trees inside every bank. In this manner more remote accesses are possible in parallel on the same bank, which has a direct impact on the reduction of memory conflict probability. Essentially, bank conflicts are reduced to two specific types of sub-bank conflicts: (i) local sub-bank conflicts on the same die, and, (ii) TSV conflicts when sub-banks on different dies are conflicting on the TSV bundles. Thus, the memory conflict probability can be significantly reduced by the proposed cache organization as for an N-die memory with S-address ports per die the maximum success probability is the same as for a memory with $N \times S$ banks, i.e., we can consider that we have a memory with $N \times S$ virtual banks.

III. Experimental Evaluation

We modified the Cacti 6 [8] simulation model to take into account the extra area of the TSV and control logic and to automatically size the subarray data drivers to support the capacitive load of a TSV crossing all dies, computed with an electrical equivalent RC model based on the resistance and capacitance equations from [9]. We consider various TSV geometries for which the model was validated against measurements [10]. We measure all relevant metrics, i.e., access time, footprint, dynamic energy, and leakage power of the best 2, 4, and 8 MB caches, considering equal optimization weights for the considered metrics. All instances are 4-way set-associative, with 64B cache block size, and implemented in a high-performance $22nm$ CMOS technology.

We present in Fig. 5 an access time comparison between banked multi-ported planar and 3D polyhedral caches with 2, 4, and 8 parallel access ports, for different vertical wide-I/O widths (i.e., number of sub-banks). Our experiments indicate that as the TSV is reduced in size all metrics proportionally improve, thus we only plot in Fig. 5 results for the largest (L) and smallest (S) TSV geometries, with the exact values found in the figure's legend.

We can observe that in general the access time of our proposed 3D caches is better than the one of their planar counterparts, with a 50% reduction in the best case. This is expected, since due to the 3D folding of the cache the footprint is reduced proportionally to the number of dies, as seen in Fig. 6, and the wires are shorter. The exceptions are explained by the longer wires present in disproportionate sub-banks layout due to the TSVs. Thus, it is only beneficial to have many dies for large caches. The reduction in footprint also leads to a considerable reduction in dynamic energy, with the maximum being close to the theoretical maximum of $8\times$, for an 8-die 8 MB cache. We plot only the read energy in Fig. 7 since the write energy exhibits similar values. Although the static power is also initially decreasing in Fig. 8 when more dies are available (because less H-tree signal repetears are needed), the gain is rapidly cancelled out by the extra logic when a lot of TSVs are added.

An important thing to notice from Figs. 6 to 8 is that trade-offs between the width of the wide-I/O interface and all the metrics exist. Larger widths offer a higher bandwidth and a conflict probability reduction, but the increase in sub-banks translates into more TSVs and extra decoding logic and wires,

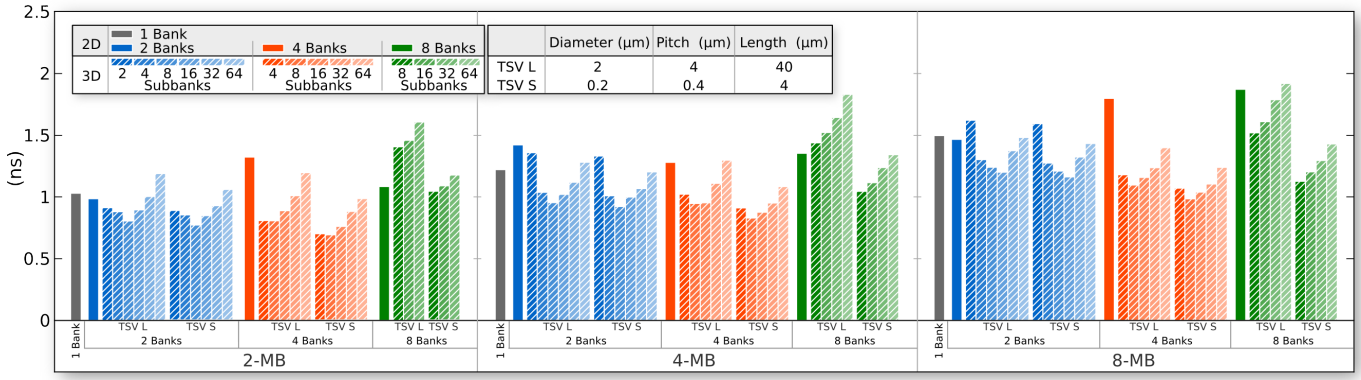

Fig. 5: Access Time Comparison.

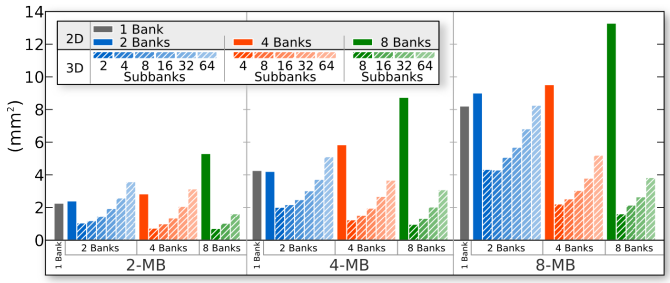

Fig. 6: Footprint Comparison.

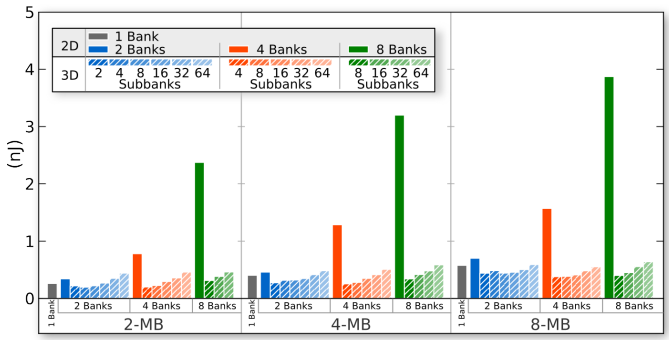

Fig. 7: Read Energy Comparison.

which in turn negatively impacts the access time, footprint and energy cost. Even though there is an apparent optimum from the access point point of view, the vertical interface width design decision should consider the memory access patterns of the running application, since they influence: (i) the total number of real address conflicts, (ii) the leakage contribution to the total energy consumption, and (iii) the amount of cache pollution [11].

IV. Conclusions

In this paper, we proposed a novel shared cache design for TSV-based wide-I/O multi-core systems, which consists of multiple identical banks, stacked on top of each other, forming a polyhedral structure. The cpu-side interface is realized horizontally through one access port on every die, while the wide-I/O memory-side interface is realized vertically through TSV bundles that traverse the entire stack. In addition to supporting the wide-I/O interface, the TSV bundles facilitate inter-die

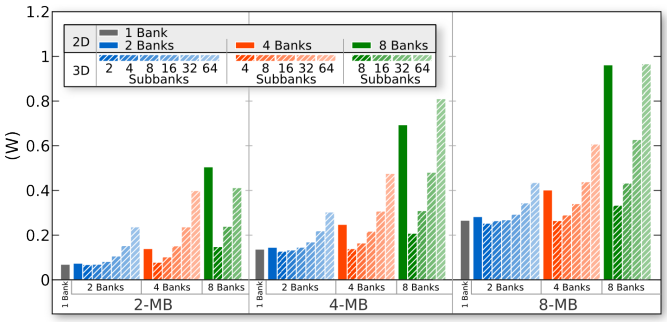

Fig. 8: Leakage Comparison.

wide cache data transfers and the creation of a number of virtual banked ports. Our simulations indicate that the polyhedral cache outperforms planar counterparts in terms of access time, energy, and footprint while providing higher bandwidth, lower bank conflict rate, and enriched functionality.

References

[1] B. Jacob *et al.*, *Memory systems: cache, DRAM, disk.* Morgan Kaufmann Pub, 2007.

[2] C. Liu *et al.*, "Bridging the processor-memory performance gap with 3D IC technology," *IEEE Design Test of Computers*, vol. 22, no. 6, pp. 556–564, Nov 2005.

[3] G. H. Loh, "3D-stacked memory architectures for multi-core processors," in *ISCA 2008*, Washington, DC, USA, pp. 453–464.

[4] R. Balasubramonian and N. Jouppi, *Multi-Core Cache Hierarchies*. Morgan & Claypool Publishers, 2011.

[5] T. Juan *et al.*, "Data caches for superscalar processors," in *International Conference on Supercomputing, ICS '97*, NY, USA, pp. 60–67.

[6] N. Muralimanohar *et al.*, "Optimizing NUCA organizations and wiring alternatives for large caches with CACTI 6.0," in *MICRO 2007*, Dec 2007, pp. 3–14.

[7] K. Puttaswamy and G. Loh, "3D-integrated SRAM components for high-performance microprocessors," *IEEE Transactions on Computers*, vol. 58, no. 10, pp. 1369–1381, Oct 2009.

[8] N. Muralimanohar *et al.*, "CACTI 6.0: A tool to model large caches," HP Laboratories, Tech. Rep., 2009.

[9] I. Savidis and E. Friedman, "Closed-form expressions of 3-D via resistance, inductance, and capacitance," *IEEE Transactions on Electron Devices*, vol. 56, no. 9, pp. 1873–1881, Sept 2009.

[10] K. Chen *et al.*, "CACTI-3DD: architecture-level modeling for 3D die-stacked DRAM main memory," in *DATE 2012*, March 2012, pp. 33–38.

[11] D. H. Woo *et al.*, "An optimized 3D-stacked memory architecture by exploiting excessive, high-density TSV bandwidth," in *HPCA 2010*, Jan 2010, pp. 1–12.