Implementing Caches in a 3D Technology for High Performance Processors

Kiran Puttaswamy[†] Gabriel H. Loh[‡]

Georgia Institute of Technology
School of Electrical and Computer Engineering[†]
College of Computing[‡]
kiranp@ece.gatech.edu loh@cc.gatech.edu

Abstract

3D integration is an emergent technology that has the potential to greatly increase device density while simultaneously providing faster on-chip communication. 3D fabrication involves stacking two or more die connected with a very high-density and low-latency interface. The die-to-die vias that comprise this interface can be treated like regular on-chip metal due to their small size (on the order of 1μm) and high speed (sub-FO4 die-to-die communication delay). The increased device density and the ability to place and route in the third dimension provide new opportunities for microarchitecture design.

In this paper, we first present a brief overview of 3D integration technology. We then focus on the design of on-chip caches using 3D integration. In particular, we show that the dense die-to-die vias enable caches that are 3D-partitioned at the level of individual wordlines or bitlines. This results in a wire length reduction within SRAM arrays, and a reduction in the footprint of individual SRAM banks, which reduces the global routing from the edge of the cache to the banks and back. The wire length reduction provides both power and performance benefits, e.g., 21.5% latency reduction and 30.9% energy reduction for a 512KB cache. We also report that implementing only the caches in 3D, without accounting for possible benefits from implementing other components of the processor in 3D, results in a 12% IPC gain. These results demonstrate some of the potential of this new technology, and motivate further research in 3D microarchitectures.

1. Introduction

Faster gate delays due to technology scaling combined with the increasing relative wire RC delays have made wires a critical performance bottleneck [1, 4, 13, 22]. In current and future generations, the wires heavily influence both the latency and the power consumed by the circuitry. Three-dimensional integrated circuits (3D ICs) is an emergent technology that vertically stacks multiple die with a high-density die-to-die interconnect [16]. Figure 1 shows the overall 3D structure of a two-die stack. The ability to route signals in the vertical dimension enables previously distant blocks to be placed on top of each other. This results in a decrease in the overall wire length, providing a reduction in wire delay and power. In this paper, we will first present an overview of 3D integration technology. We will then discuss several approaches for applying 3D integration to the design of on-chip caches. In particular, we will show how the 3D die-to-die interconnect enables 3D caches that are finely partitioned at the wordline or bitline level for performance and power benefit.

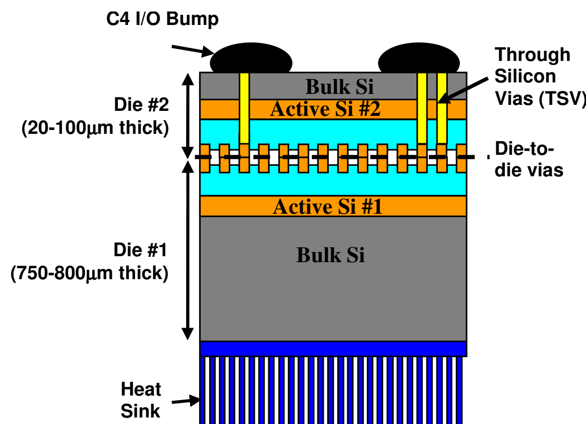

Figure 1. A two-die face-to-face 3D structure (not drawn to scale) [3].

There is currently a large interest in 3D integration. Researchers in the semiconductor industry are evaluating the technology for feasibility and applicability [3, 12, 17, 20]. In the embedded space, there are already companies shipping products that use 3D integration for 3D-stacked SRAM, DRAM and combined SRAM and micro-controller [26] as well as eight-die memory stacks [23]. On the academic side, much of the research effort has been focused on the process technology, physical design and automated design tools [5, 6, 8, 11, 15, 19, 25], but we are not aware of any 3D microarchitecture studies for high-performance microprocessors.

This paper makes two primary contributions. The first is an overview of recent advances in 3D technology for the computer architecture and microarchitecture commu-

nity. The second is a case study quantifying and analyzing the value of 3D integration for on-chip caches in modern high-performance processors. The results are very encouraging and motivate further 3D research. Our goal is to understand how an entire microprocessor should be designed to best exploit the benefits of a 3D technology, but the design of a full 3D microarchitecture is beyond the scope of this study.

The rest of the paper is organized as follows. Section 2 presents an overview of 3D technology and its potential impact on processor design. Section 3 describes several approaches for using 3D in the design of on-chip caches. Section 4 provides the details about our experimental framework. Section 5 presents the results and analysis of our 3D caches. Section 6 further discusses limitations and design alternatives for 3D caches. Section 7 summarizes our contribution.

2. 3D Integration Technology

2.1. Organization of 3D Structures

3D integration takes two or more die and stacks them in a 3D structure. There are several possible implementation technologies currently being considered. The face-to-face (F2F) organization illustrated in Figure 1 provides the greatest die-to-die via density [3, 7, 17]. This particular process for F2F bonding consists of depositing "stub" vias on to the top-level metal of each die as if another metal layer were to be implemented. The two die are then placed face-to-face such that each stub from the first die is directly pressed against the corresponding stub on the second die. Under the proper conditions, the via stubs will fuse to simultaneously hold the two die together as well as form the die-to-die vias needed to communicate between the two die. These are fabricated like conventional inter-level metal vias, and therefore have similar sizes and RC characteristics. The size and pitch of the die-to-die vias is not limited by lithographic constraints, but rather by the accuracy of the mechanism for aligning the two die. Wafer-to-wafer bonding provides higher throughput; die-to-die techniques can provide finer alignment; and there are other hybrid techniques such as die-to-wafer and partial-wafer bonding that trade throughput and alignment accuracy. After bonding, one die is mechanically and/or chemically thinned, and through-silicon vias (TSV) are etched through the back side of the thinned die to provide I/O and power/ground connections.

An alternative to face-to-face bonding is face-to-back (F2B) bonding [8, 21]. In this technology, die-to-die vias must be etched through the back-side (device-side) of a die. The etching process has less resolution than a F2F technology. As a result, the effective density of the vias for a F2B process is reduced. Furthermore, the etching process requires the die-to-die via to pass through the device layer, which may seriously disrupt the crystal structure of a strained-silicon process. This will result in degraded device performance in areas near the back-side vias. An advantage of F2B bonding is that it can be repeated for an arbitrary number die stacked together. With F2F bonding, only alternating layers may be bonded with the dense face-to-face interconnect, while the back-to-back interface will be limited to a coarser interconnect. In this paper, our detailed circuit analysis only considers the design of caches using a two-die F2F stack, but we will discuss some of the design implications and possibilities of stacking more than two die.

2.2. Physical Characteristics of 3D Structures

One of the critical parameters of a 3D technology is the physical characteristics of the die-to-die interconnect. The density of the interconnect will determine at what granularity 3D can be used to affect the microarchitecture. The latency to drive a signal from one die to the other also affects how to partition a processor across multiple die.

For F2F 3D stacking, the currently implemented die-to-die via pitch varies from 3μm to 10μm, depending on the technology [8]. Note that these via pitches are for existing implementations; as the alignment technology matures, the via pitch will improve to 1μm or denser [8]. As a point of comparison, a dense 6T SRAM cell with an area of $0.7\mu\text{m}^2$ [10] in a 65nm technology can support about one 1μm-pitch die-to-die via per two SRAM cells. While this suggests that it may not be feasible to split an individual 6T cell across two die, the via density is great enough to support 3D folding at the level of wordlines and bitlines.

The die-to-die vias in a F2F 3D technology are simply short lengths of metal. It is important to realize that the die-to-die interconnect does not behave like I/O pads but like regular vias. The total length of the die-to-die via to connect the two die varies from $<5\mu$m to $\sim 30\mu$m [8]. In a 70nm technology (see Section 4 for the details of our experimental methodology) the delay for an inverter (at $4\times$ minimum size) to drive through a via-stack, across the die-to-die via, and back down another via stack is only 8ps. For comparison, the FO4 delay in the same technology is 22ps, and the delay for a 4x inverter to drive 1mm of Metal6 is 225ps. The combination of high density and low latency make die-to-die vias a very effective alternative to conventional on-die routing.

3. 3D Cache Designs

We describe the design of an SRAM cache in 3D. The SRAM cache has a very regular structure that makes it easy to partition across multiple die. More specifically, we evaluate 3D cache organizations that stack a large cache on top of the main CPU core, banks on banks, and array-level partitioning.

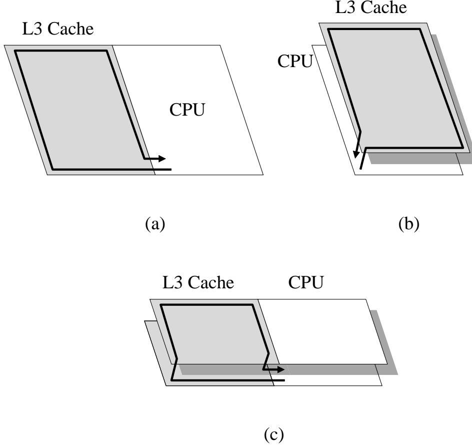

Figure 2. (a) A planar layout of a CPU and L3 cache with the L3 critical wire path, (b) a 3D implementation with the cache stacked on top of the CPU, and (c) a 3D implementation with the cache partitioned and stacked upon itself.

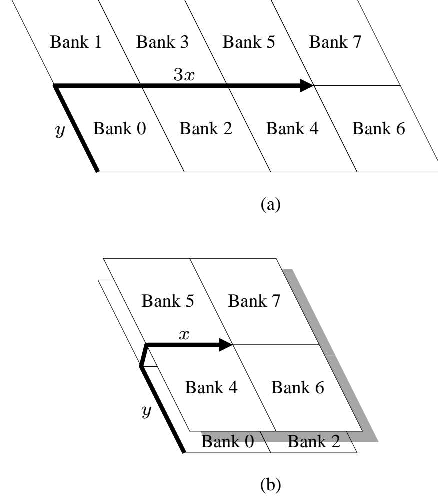

Figure 3. (a) A planar eight-way banked cache showing the worst-case distance to the farthest bank, and (b) a 3D bank-stacked cache organization.

This study does not assume changes in the cache interface. This simplifies the study; future research will consider the complete datapath, as changing the interface will require that many other components such as ALUs, bypass paths, the instruction scheduler payload RAM, etc. all be reimplemented in 3D. This will likely provide even greater benefits, but it is beyond the scope of this paper. We are taking a bottom-up approach by first understanding how 3D impacts individual units, and then leveraging these insights to devise how to best compose them into a complete 3D datapath.

3.1. Cache on Processor

An intuitive application of 3D integration in the context of on-chip caches is the stacking of a very large last-level cache, e.g., L3 cache, on top of a traditional execution core. Figure 2(a) shows a conventional planar processor with a large L3 cache, and Figure 2(b) illustrates a 3D implementation with a stacked L3 cache. The benefit of such an organization is an approximate footprint reduction of 50%.

There are several disadvantages with this approach that are less obvious. While a cache-on-CPU approach takes advantage of the increased integration capacity of a 3D process, it fails to utilize the benefits of having a dense die-to-die interconnect structure and the flexibility of vertical routing. Figure 2(a) shows the critical path for accessing the planar cache array. Note that in the cache-on-CPU implementation in Figure 2(b), the structure of the cache array has not changed and therefore the critical path is identical. The net result is that neither latency nor power have been improved by this organization, and it thus provides no performance or power benefit. Furthermore, stacking a large cache on top of the CPU prevents any of the modules within the CPU from taking advantage of a 3D organization.

3.2. Bank Stacking

In this study, we evaluate *self-stacked* caches. Figure 2(c) shows the concept of splitting and stacking a cache to reduce the critical wire lengths within the structure. The wire length reduction can be exploited to reduce latency, reduce power, or increase the cache size.

Long metal wires used to route long-haul signals suffer from large RC delays and power consumption. The wires that route from the edge of the cache to a bank are an example of such wires. Figure 3(a) shows an eight-bank cache array with the critical wire path from the bottom left corner of the cache to the farthest bank. One option for a 3D-integrated cache is to stack banks on top of each other. Figure 3(b) shows the same array arranged in a 3D bank-stacked organization. A 3D-bank stacking approach was previously proposed by Reed et al. [20].

There are two possible orientations for bank stacking. The cache can be stacked left-to-right as shown in Figure 3(b), or the banks can be stacked top-to-bottom. Figure 3(b) shows how the bank stacking results in a 67% reduction ($3x \rightarrow x$, where x is the bank width) in the horizontal component of the wiring to and from the banks. Because the stacking of the cache occurs only in one direction, the vertical component of the bank wiring is unaffected, thus reducing the wire length savings to 50% (assuming $x = y$, where y is the bank height). Overall, the reduction in wire length translates into a reduction of power and delay.

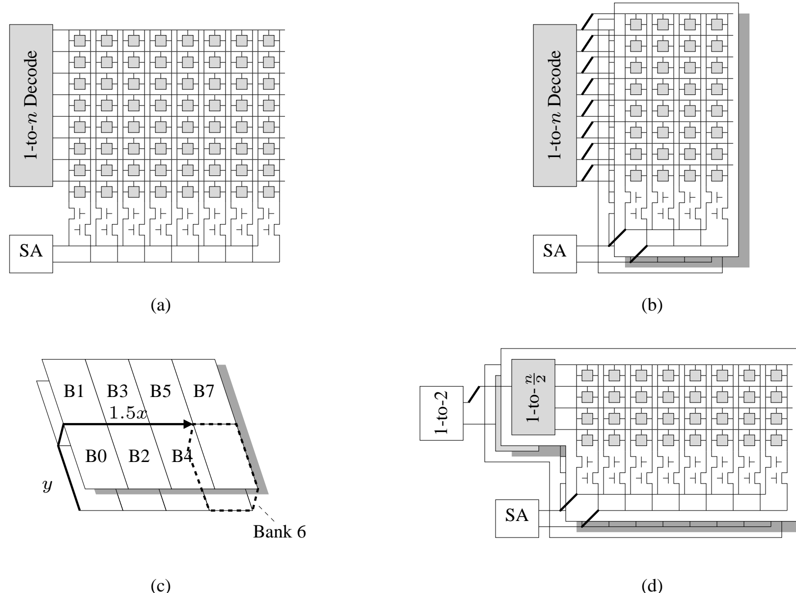

Figure 4. (a) A conventional 2D/planar SRAM array, (b) the same array in 3D with column-on-column stacking, (c) the bank-level organization of a cache using column-stacked arrays, (d) a 3D array using row-on-row stacking.

3.3. Array Splitting

Another level of cache stacking that we evaluate is partitioning individual SRAM arrays and stacking them upon themselves. Array splitting can reduce the lengths of the wordlines or bitlines depending on the orientation of the split. Figure 4(a) shows the details of a small SRAM data array. With bank-stacking, the decision for a horizontal or vertical split largely depends on which dimension has more wire delay. At the array level, the orientation of splitting has a greater impact on the implementation of the array itself.

The first split-array configuration that we consider comes from stacking columns on columns. Figure 4(b) illustrates the 3D array implemented with column stacking. In this design, the single long wordline has been replaced by a pair of parallel wordlines. The row decoder must discharge the same amount of capacitance, but the parallel wires reduce resistance by about one half. This arrangement requires one die-to-die via per wordline. At the bottom of the array, the column select muxes have also been split across the two die thus requiring two additional die-to-die vias. This organization provides some reduction in latency and power. As shown in Figure 4(c), these benefits are in addition to reductions in the bank-level wire lengths.

The second split-array organization stacks rows on rows. Figure 4(d) shows the row-stacked 3D array. The row-stacked array requires that the row decoder is also partitioned across the two die. We first decompose the 1-to-n decoder into a 1-to-2 decoder followed by two 1-to-$\frac{n}{2}$ decoders. Even though the two 1-to-$\frac{n}{2}$ decoders are stacked on top of each other, the 1-to-2 decoder will only activate one of them which avoids thermal stacking of active components. The length and loading of the wordlines remain the same as in the planar organization, but the length of the bitlines have been halved. The shorter wire lengths result in a decrease in access latency. Similar to the stacked-column organization (not shown in Figure 4), there are latency and power benefits due to wire reduction at both the array- and bank-levels. From a certain perspective, this split-array organization looks like 3D sub-banking where the top and bottom halves of the 3D array correspond to the sub-banks of a conventional planar SRAM.

3.4. Qualitative Comparison of 3D Caches

The cache-on-CPU approach reduces the chip footprint, but there are no substantial power or performance benefits. In the embedded domain, the overall package size can be a very important design factor, and a cache-on-CPU approach may be quite attractive, especially when the alternatives are to have an off-chip cache, an on-chip cache but with a larger package, or no cache at all. In the context of high-performance processors, the cache-on-CPU organization provides some benefit, but overall it is a sub-optimal way to exploit a 3D technology.

At a finer level, we have proposed two possible 3D organizations for typical cache structures. The array-splitting approach provides greater overall benefit because it provides latency and power savings both within individual arrays as well as for bank-level routing. Bank-stacking

only provides the savings at the global routing-level, and it under-utilizes the dense die-to-die via interface. Bank-stacking is simpler to implement because at the level of individual SRAM arrays, the design is no different from a conventional 2D process, whereas the array-stacked approach requires a redesign of the array.

4. Experimental Framework

We describe our experimental framework to evaluate the overall circuit- and system-level performance of 3D caches. For our performance studies, we used a combination of circuit-level SPICE simulations to obtain the latencies and power of the 2D and 3D caches, and SimpleScalar-based cycle-level simulation for IPC.

4.1. Circuit Timing and Power Evaluation

We have built a framework to obtain the critical path latency and power of SRAM arrays for conventional planar and 3D caches. Our framework consists of a frontend netlist generator and a backend circuit simulator. The frontend generates the SPICE netlist for the critical path through a cache circuit for varying transistor sizing parameters. We simulate the critical path circuitry of the cache with user selectable parameters, such as bank organization, array capacity, associativity and planar vs. 3D organization.

Taking the generated netlist, our backend uses SPICE to obtain the critical path latency of a read operation and the power consumption of the entire array. We use the BSIM 70nm transistor models from Berkeley and extrapolate wire parameters obtained from the MOSIS/TSMC 180nm process. The distance between the top metal layers on the two die is very small [21, 24], and the pitch of the die-to-die vias are on the same order as the top level metal [9]. Therefore we model the cross-die interconnect as 10μm of top-level metal plus the corresponding via contact resistance. To optimize our cache designs, we sweep through a range of transistor sizings as well as bank organizations. We use the configurations that minimize overall cache access time.

We simulate a cache read operation that includes the bank-level and array-level circuit details. The simulations account for all of the wiring and drivers to get from the cache boundary to the farthest bank. The request propagates through the decoder tree, the wordline, the SRAM cell itself, the bitlines, the column mux, the sense amplifier and the way-mux. This signal is then driven back through a series of wires and bank muxes to finally return to the edge of the cache. While the critical timing path only involves a single SRAM entry and the corresponding pair of bitlines, our power simulations include the activity associated with enabling *all* of the SRAM cells and their bitlines in the selected array row. We perform a similar analysis for the critical path through the tag array, including the tag comparators and the interaction with the way-select muxes on the data side. We use the worst case timing path through either array.

Size (KB)	Latency 2D (ns)	Latency 3D (ns)	Savings %	Cycles 2D @ 4GHz	Cycles 3D @ 4GHz
16	0.754	0.714	5.3	4	3
32	0.815	0.754	7.6	4	4
64	0.944	0.816	13.6	4	4
128	1.164	0.945	18.8	5	4
256	1.296	1.213	6.4	6	5
512	1.709	1.341	21.5	7	6
1024	2.037	1.763	13.5	9	8
2048	2.609	2.087	20.0	11	9

Table 1. The impact of 3D on cache latency.

4.2. Performance Analysis

We use the MASE timing simulator [14] from SimpleScalar 4.0 [2] for the Alpha instruction set architecture. We simulate a 6-wide processor, with a 128-entry ROB, 64-entry scheduler, 32 entry load and store queues, 16KB IL1 and DL1 caches, a 256KB L2 cache, and a 1MB L3 cache. We set our clock frequency to 4GHz assuming a 70nm technology.

We evaluate the performance impact of 3D caches with the SPEC2000 benchmark suite. We use the Alpha binaries that are available on the SimpleScalar website. We simulate the applications from both SpecINT and SpecFP using reference inputs. To reduce the total simulation time, we use single 100 million instruction simulation samples chosen with the SimPoint 2.0 toolset [18]. We use the geometric mean when reporting overall IPC.

5. Latency, Power and Performance Impact

5.1. Design Trade-Offs

We evaluate our 3D cache designs against a conventional planar 2D cache. We consider cache sizes varying from small 16KB level-one (L1) caches up to large 2MB last-level (L2/L3) caches. Our baseline for comparison is a 2D cache with transistor sizings and bank configurations chosen to minimize latency.

5.1.1. Impact on Access Latency

The 3D reorganization of cache structures can substantially reduce wire delays. This latency reduction is due to savings in routing to and from the individual banks, as well as wiring within the SRAM arrays. Table 1 reports the size and latency of the baseline 2D cache, the latency of the 3D caches, the relative latency reduction, and the latency in terms of 4GHz clock cycles. The observed latency benefit varies by cache size. The 3D organization provides more benefit to the larger caches because these structures have

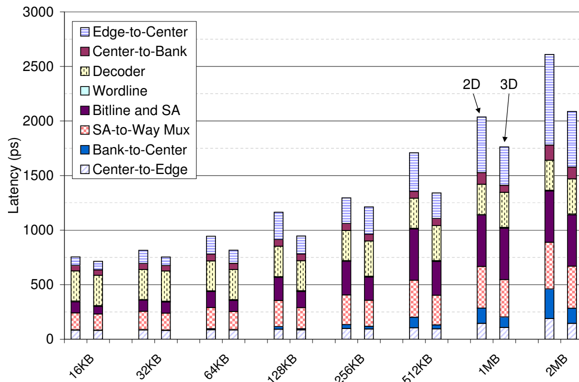

Figure 5. The component-wise breakdown of the latency for a cache read operation.

Size (KB)	Energy 2D (pJ)	Energy 3D (pJ)	Savings %
16	14.61	12.43	14.9
32	15.52	14.69	5.4
64	19.92	15.61	21.6
128	26.87	20.05	25.4
256	33.92	27.07	20.2
512	49.62	34.30	30.9
1024	54.68	49.97	8.6
2048	65.28	54.87	16.0

Table 2. The impact of 3D on cache energy.

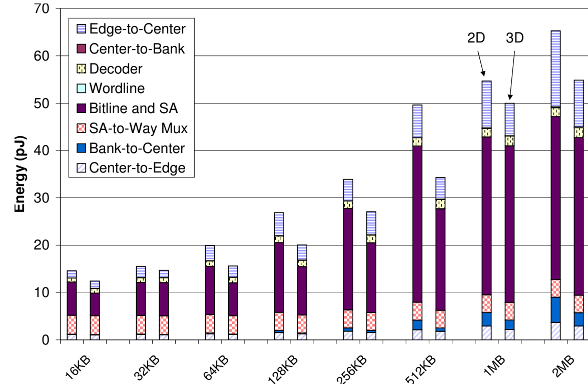

Figure 6. The component-wise breakdown of the energy for a cache read operation.

substantially longer global wires to route signals between the cache edge and the farthest bank. The latency improvement in Table 1 does not increase monotonically because the best configuration between the 2D and 3D caches may involve a different number of banks, which in turn changes the relative benefit. For many 3D configurations, the latency reduction is sufficient to reduce the overall number of clock cycles for the access. For the smaller caches, the wire delays comprise a smaller relative fraction of the overall cache latency and therefore the effect of reducing these delays is less.

To understand where 3D has the greatest benefit, Figure 5 illustrates the latency contribution of the different components that comprise a cache. Depending on the cache size, different 3D topologies may be required to provide the best latency improvements. For example, the bank-level routing latency for the 2MB 2D cache makes up over 55% of the total latency. Correspondingly, the fastest 3D 2MB organization exhibits the greatest latency reduction in these timing components, as shown in Figure 5. On the other hand, the moderate-sized 64KB-512KB cache delays are not as dominated by the global routing. In these instances, a 3D organization that targets the intra-SRAM delays provides more benefit. Figure 5 shows that, in particular, the delay reduction associated with toggling and sensing the bit-lines accounts for a substantial amount of the overall benefit, although other timing components also observe some improvement. In general, circuit paths dominated by long wire RC delays have the greatest potential for improvement from a 3D reorganization. In contrast, the row decoder consists primarily of logic, and the results in Figure 5 are consistent with the expectation that 3D would not provide much benefit for this component of the overall cache latency.

5.1.2. Impact on Access Energy

The 3D organization of the cache structure reduces critical wire lengths, which reduces the total wire capacitance providing both performance *and* power benefits. For the same cache configurations discussed in the previous section, Table 2 shows the energy consumed per access for both the 2D and 3D configurations. The overall savings range from 5.4-30.9%, varying due to differences in the optimal banking configurations, transistor sizings, and SRAM aspect ratios. Figure 6 shows the energy consumption for the various components of the cache read operation.

A component's relative contribution to total energy does not always reflect its impact on power. For example, the row decoder logic accounts for a considerable amount of latency, but it only needs to switch a single wordline and so it does not consume much energy. On the other hand, a single bitline switch and the corresponding sense amp only incur a modest energy cost, but this cost must be multiplied by the number of bits read and by the set associativity, which adds up to account for a substantial portion of total cache access energy. The 3D cache organization reduces the energy in the most wire-dominated portions of the cache, namely the bank-level routing and the bitlines.

5.2. Performance Analysis

We used cycle-level processor simulations to evaluate the performance impact of 3D caches. Our microarchitecture has four caches: the L1 instruction and data caches, and the L2 and L3 shared caches. Based on the results presented in the previous section, we can choose to use 3D to reduce the access latency or to increase the capacity of each of these caches. While a typical cache hierarchy IPC design study may not be of great interest in and of itself, it serves to demonstrate that 3D integration can be translated into real performance gains.

The two simplest ways to exploit the benefits of 3D caches are to either make all of the caches faster, or make all of the caches larger. Using 3D to implement all caches in the hierarchy to be faster results in a 3.5% speedup in the geometric mean IPC. Using 3D to implement larger caches results in a 10.6% speedup. There are also fourteen remaining possibilities that involve a mix of making some caches larger and other caches faster, of which the best configuration results in a 12.1% performance gain. This speedup is what 3D can deliver for our microarchitecture when considering *only the caches*. We believe that the combination of our performance, timing and energy results provides great motivation to find ways to exploit 3D throughout the entire processor.

6. Alternatives, Trade-offs, Limitations

The primary limiter on what can be implemented in a 3D technology is the die-to-die via interface. Apart from this, there are no fundamental restrictions on what can or cannot be done in 3D, although economic factors such as extra processing steps, yield, design and validation effort, may all restrict what is practical. Caches can be split between two die as advocated by this study: the cache can be stacked on top of the CPU to reduce footprint, or could even placed multiple cache layers above and below the processor in a cache-CPU-cache stack. We have already discussed in Section 3 why a cache-on-CPU approach does not really exploit the strengths of a 3D technology. Similar arguments hold for a cache-CPU-cache organization. There are many other possible arrangements, all of which could be implemented in 3D; it is just a question of which organization provides the greatest benefit and value.

The die-to-die via pitch determines the granularity at which 3D folding can occur. Furthermore, there may be multiple interfaces with different via pitches, such as in the case of multiple die stacked in an alternating face-to-face and back-to-back organization. With this type of 3D structure, one could self-stack the CPU across two face-to-face bonded die, implement the large L3 cache as a self-stacked structure across a second pair of face-to-face die, and then connect the L3 cache to the CPU through back-to-back vias. This approach exploits the high-density face-to-face vias to reduce the wire delay and latency within and between smaller functional units, and it simultaneously matches the larger-pitched back-to-back interface to a microarchitectural boundary that does not require a high communication density.

We believe that self-stacked microarchitectural modules not only provide the best performance benefits, but they are also necessary to manage the thermals of a 3D processor. By stacking multiple die, we create the potential for significantly increased power density which may lead to thermal problems. Using the cache-on-CPU example, simply taking conventional planar implementations of blocks (CPU, cache) and reorganizing them in 3D does nothing to reduce the power within the blocks. At most, some power is saved at the interface between the CPU and the last-level cache which is only a small savings due to the low activity factor of the L3 cache. The power consumption of the CPU itself as well as the L3 cache have not changed, and stacking them only serves to increase overall power density. Implementing all or most of a processor's components as self-stacked 3D structures can reduce power throughout the entire chip. This reduction of chip-wide total power consumption will be important to keep thermals under control.

7. Summary

Three-dimensional integration provides significant benefits in terms of transistor density and wire reduction. This study has demonstrated how these benefits may be used to implement faster and lower power on-chip caches. The effects of this new technology on the design of high-performance microprocessors is still a largely unexplored research topic. We believe that 3D integration can have a profound impact on processor performance and power, and the ability to stack multiple die may delay the end of Moore's Law for a few more process generations.

The on-chip caches account for only a few of the many components in a modern processor microarchitecture. Much research remains to understand how to best redesign the other functional unit blocks to exploit the strengths (integration capacity, vertical routing) of 3D integration technology. Our overall goal is to understand the impact of 3D integration on microarchitecture and to use this understanding to change how we design future microprocessors.

Acknowledgments

This research is generously supported by funding and equipment from Intel Corporation, and a grant from MARCO (2003-DT-660). We would like to thank the program committee and blind reviewers for their criticisms and feedback. We are also grateful to Sally McKee of Cor-

nell University for her insightful and detailed comments in preparation of the final version of the paper.

References

[1] Vikas Agarwal, M. S. Hrishikesh, Stephen W. Keckler, and Doug Burger. Clock Rate Versus IPC: The End of the Road for Conventional Microarchitectures. In *Proceedings of the 27th International Symposium on Computer Architecture*, pages 248–259, Vancouver, Canada, June 2000.

[2] Todd Austin, Eric Larson, and Dan Ernst. SimpleScalar: An Infrastructure for Computer System Modeling. *IEEE Micro Magazine*, pages 59–67, February 2002.

[3] Bryan Black, Don Nelson, Clair Webb, and Nick Samra. 3D Processing Technology and its Impact on IA32 Microprocessors. In *Proceedings of the 22nd International Conference on Computer Design*, pages 316–318, San Jose, CA, USA, October 2004.

[4] Shekhar Borkar. Design Challenges of Technology Scaling. *IEEE Micro Magazine*, 19(4):23–29, July 1999.

[5] L. Cheng, L. Deng, and M. Wong. Floorplan Design for 3-D VLSI Design. In *Proceedings of the Asia South Pacific Design Automation Conference*, Shanghai, China, January 2005.

[6] Jason Cong and Y. Zhang. Thermal-Driven Multilevel Routing for 3-D ICs. In *Proceedings of the Asia South Pacific Design Automation Conference*, Shanghai, China, January 2005.

[7] Shamik Das, Anantha Chandrakasan, and Rafael Reif. Three-Dimensional Integrated Circuits: Performance, Design Methodology, and CAD Tools. In *Proceedings of the International Symposium on VLSI*, pages 13–18, Tampa, FL, USA, February 2003.

[8] Shamik Das, Andy Fan, Kuan-Neng Chen, and C. S. Tan. Technology, Performance, and Computer-Aided Design of Three-Dimensional Integrated Circuits. In *Proceedings of the International Symposium on Physical Design*, pages 108–115, Phoenix, AZ, USA, April 2004.

[9] Yangdong Deng and Wojciech Maly. 2.5D System Integration: A Design Driven System Implementation Schema. In *Proceedings of the Asia South Pacific Design Automation Conference*, pages 450–455, Yokohama, Japan, January 2004.

[10] F. Arnaud et al. A Functional $0.69\mu\text{m}^2$ Embedded 6T-SRAM Bit Cell for 65nm CMOS Platform. In *Proceedings of the 19th Symposium on VLSI Technology*, pages 342–351, Kyoto, Japan, June 2003.

[11] Brent Goplen and Sachin Sapatnekar. Efficient Thermal Placement of Standard Cells in 3D ICs Using a Force Directed Approach. In *Proceedings of the International Conference on Computer-Aided Design*, pages 81–85, San Jose, CA, USA, November 2003.

[12] K. W. Guarini, A. W. Topol, M. Ieong, R. Yu, L. Shi, M. R. Newport, D. J. Frank, D. V. Singh, G. M. Cohen, S. V. Nitta, D. C. Boyd, P. A. O'Neil, S. L. Tempest, H. B. Pogge, S. Purushothaman, and W. E. Haensch. Electrical Integrity of State-of-the-Art 0.13μm SOI CMOS Devices and Circuits Transferred for Three-Dimensional (3D) Integrated Circuit (IC) Fabrication. In *Proceedings of the International Electron Devices Meeting*, pages 943–945, December 2002.

[13] Ron Ho, Kenneth W. Mai, and Mark A. Horowitz. The Future of Wires. *Proceedings of the IEEE*, 89(4):490–504, April 2001.

[14] Eric Larson, Saugata Chatterjee, and Todd Austin. MASE: A Novel Infrastructure for Detailed Microarchitectural Modeling. In *Proceedings of the 2001 International Symposium on Performance Analysis of Systems and Software*, pages 1–9, Tucson, AZ, USA, November 2001.

[15] John Mayega, Okan Erdogan, Paul M. Belemjian, Kuan Zhou, John F. McDonald, and Russel P. Kraft. 3D Direct Vertical Interconnect Microprocessors Test Vehicle. In *Proceedings of the ACM Great Lakes Symposium on VLSI*, pages 141–146, Washington, DC, USA, April 2003.

[16] Patrick Morrow, Mauro J. Kobrinsky, Shriram Ramanathan, Chang-Min Park, Michael Harmes, Vijay Ramachandrarao, Hyun mog Park, Grant Kloster, Scott List, and Sarah Kim. Wafer-Level 3D Interconnects Via Cu Bonding. In *Proceedings of the 21st Advanced Metallization Conference*, San Diego, CA, USA, October 2004.

[17] Don Nelson, Clair Webb, Don McCauley, Kartik Raol, Jeff Rupley II, John DeVale, and Bryan Black. A 3D Interconnect Methodology Applied to iA32-class Architectures for Performance Improvements through RC Mitigation. In *Proceedings of the 21st International VLSI Multilevel Interconnection Conference*, Waikoloa Beach, HI, USA, September 2004.

[18] Erez Perelman, Greg Hamerly, and Brad Calder. Picking Statistically Valid and Early Simulation Points. In *Proceedings of the 2003 International Conference on Parallel Architectures and Compilation Techniques*, pages 244–255, New Orleans, LA, USA, September 2004.

[19] Arifur Rahman and Rafael Reif. System Level Performance Evaluation of Three-Dimensional Integrated Circuits. *IEEE Transactions on VLSI*, 8(6):671–678, June 2000.

[20] Paul Reed, Gus Yeung, and Bryan Black. Design Aspects of a Microprocessor Data Cache using 3D Die Interconnect Technology. In *Proceedings of the International Conference on Integrated Circuit Design and Technology*, pages 15–18, Austin, TX, USA, May 2005.

[21] Rafael Reif, Andy Fan, Kuan-Neng Chen, and Shamik Das. Fabrication Technologies for Three-Dimensional Integrated Circuits. In *Proceedings of the 3rd International Symposium on Quality Electronic Design*, pages 33–37, San Jose, CA, USA, March 2002.

[22] Ronny Ronen, Avi Mendelson, Konrad Lai, Shih-Lien Lu, Fred Pollack, and John P. Shen. Coming Challenges in Microarchitecture and Architecture. *Proceedings of the IEEE*, 89(3):325–340, March 2001.

[23] Samsung Electronics Corporation. Samsung Electronics Develops World's First Eight-die Multi Chip Package for Multimedia Cell Phones. Press Release from `http://www.samsung.com`, January 10 2005.

[24] S. Strickland, E. Ergin, D. R. Kaeli, and P. Zavracky. VLSI Design in the 3rd Dimension. *Integration: the VLSI Journal*, 25(1):1–16, September 1998.

[25] Thitipong Tanprasert. An Analytical 3-D Placement that Reserves Routing Space. In *Proceedings of the International Symposium on Circuits and Systems*, volume 3, pages 69–72, Geneva, Switzerland, May 2000.

[26] Tezzaron Semiconductor. WWW Site. `http://www.tezzaron.com`.