Designing Vertical Processors in Monolithic 3D

Bhargava Gopireddy and Josep Torrellas
University of Illinois at Urbana-Champaign
http://iacoma.cs.uiuc.edu

ABSTRACT

A processor laid out vertically in stacked layers can benefit from reduced wire delays, low energy consumption, and a small footprint. Such a design can be enabled by *Monolithic 3D* (*M3D*), a technology that provides short wire lengths, good thermal properties, and high integration. In current M3D technology, due to manufacturing constraints, the layers in the stack are asymmetric: the bottom-most one has a relatively higher performance.

In this paper, we examine how to partition a processor for M3D. We partition logic and storage structures into two layers, taking into account that the top layer has lower-performance transistors. In logic structures, we place the critical paths in the bottom layer. In storage structures, we partition the hardware unequally, assigning to the top layer fewer ports with larger access transistors, or a shorter bitcell subarray with larger bitcells. We find that, with conservative assumptions on M3D technology, an M3D core executes applications on average 25% faster than a 2D core, while consuming 39% less energy. With aggressive technology assumptions, the M3D core performs even better: it is on average 38% faster than a 2D core and consumes 41% less energy. Further, under a similar power budget, an M3D multicore can use twice as many cores as a 2D multicore, executing applications on average 92% faster with 39% less energy. Finally, an M3D core is thermally efficient.

CCS CONCEPTS

• **Hardware → 3D integrated circuits**; **Die and wafer stacking**; **Emerging architectures**; *Chip-level power issues*; *Partitioning and floorplanning*.

KEYWORDS

Processor Architecture, 3D Integration, Monolithic 3D.

ACM Reference Format:
Bhargava Gopireddy and Josep Torrellas. 2019. Designing Vertical Processors in Monolithic 3D. In *The 46th Annual International Symposium on Computer Architecture (ISCA '19), June 22–26, 2019, Phoenix, AZ, USA.* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3307650.3322233

1 INTRODUCTION

Vertical processors — i.e., processors laid out vertically in stacked layers — can reap major benefits in reduced wire delays, low energy consumption, and small footprint. Currently, 3D integration consists of stacking dies and using Through-Silicon Vias (TSVs) for inter-die communication [11, 16, 41]. In this paper, we call this approach *TSV3D*. Unfortunately, TSV3D is a poor match for vertical processors. Specifically, the thick TSVs inhibit fine-grained hardware partitioning across dies. Further, the challenge of cooling the layers that are far from the heat sink limits the flexibility of TSV3D designs [16, 41].

Monolithic 3D (*M3D*) [6, 8, 14] is a 3D integration technology that allows high-bandwidth communication across layers and ultra high-density integration. Rather than bonding together pre-fabricated dies as in TSV3D, an M3D chip is built by sequentially fabricating multiple layers of devices on top of one another.

Using M3D to build vertical processors is attractive for three reasons. First, the active layers in M3D are separated by a distance of less than 1μm, which is one to two orders of magnitude shorter than in TSV3D [5, 20, 22]. Such short distance reduces the communication latency between the layers of a processor and allows for very compact designs.

Second, heat flows vertically easily, thanks to a low thermal resistance. This is in contrast to TSV3D designs, which include relatively thick, thermally-resistive layers such as the die-to-die layers [1]. As a result, temperatures away from the heat sink in M3D can be kept moderate.

Third and most importantly, the layers communicate using Monolithic Interlayer Vias (MIVs), which have diameters that are two orders of magnitude finer than TSVs [5, 7, 14, 20, 22, 31]. The tiny diameters of MIVs allow designers to use many of them, dramatically increasing the bandwidth of inter-layer communication. They enable the exploitation of fine-grain partitioning of processor structures across layers, reducing wire length, energy consumption, and footprint.

M3D is a promising technology to continue increasing transistor integration as Moore's law sunsets. As a result, there has been significant recent interest in surmounting the challenges of fabricating M3D chips [14, 44, 45]. The 2017 IRDS roadmap [21] predicts that vertical nanowires will be realized in several years' time, followed by M3D. Prototypes of M3D systems have been demonstrated, signaling that this technology is feasible [6, 14, 46]. Finally, CAD tools required for 3D floorplanning are being actively developed as well [12, 39, 44].

As M3D becomes feasible, it is essential for computer architects to understand the opportunities and challenges of building vertical processors with this technology. As hinted above, M3D offers short wire lengths, good thermal properties, and high integration. However, an important constraint of current M3D technology is that different layers in an M3D stack have different performance. Specifically, the bottom-most layer is built with high-performance transistors. However, any subsequent layer built on top of it must be fabricated at low temperature, to avoid damaging the bottom-layer devices. As a result, the transistors of any layer beyond the

ISCA '19, June 22–26, 2019, Phoenix, AZ, USA

ACM ISBN 978-1-4503-6669-4/19/06…$15.00
https://doi.org/10.1145/3307650.3322233

bottom-most have a lower performance [35, 43, 45]. This imbalance has implications on how to design processor structures.

This work is the first one to show how to partition a processor for M3D. We design a vertical processor by taking logic, storage, and mixed logic-storage pipeline stages, and partition each of them into two layers. Our partition strategy is aware of the fact that the top layer has lower-performance transistors. Specifically, in logic structures, we place the critical paths in the bottom layer and the non-critical ones in the top one. In multi-ported storage structures, we asymmetrically partition the ports, assigning to the top layer fewer ports with larger access transistors. For single-ported storage structures, we asymmetrically partition the bitcell array, assigning to the top layer a shorter subarray with larger bitcells.

With conservative assumptions on M3D technology, our M3D core executes applications on average 25% faster than a 2D core, while consuming 39% less energy. With aggressive technology assumptions, the M3D core is on average 38% faster than a 2D core and consumes 41% less energy. Further, under a similar power budget, an M3D multicore can use twice as many cores as a 2D multicore, executing applications on average 92% faster with 39% less energy. Finally, the M3D core is thermally efficient.

Overall, our contributions are:

- First work to partition cores for an M3D stack.
- Novel partition strategies of logic and storage structures for an environment with heterogeneous layers.
- Performance, power, and thermal evaluation of a single and multiple M3D cores.

2 3D MONOLITHIC INTEGRATION

3D Monolithic Integration (M3D or 3DMI) is a device integration technology that allows ultra-high density, fine-grained 3D integration. It involves fabricating two or more silicon layers sequentially on the same substrate. The bottom layer of transistors is fabricated first, using the same techniques as in a traditional die. Later, a layer of active silicon is grown on top of the bottom layer using novel techniques at a lower temperature [5, 8, 14]. Transistors are then formed on the top layer using a low-temperature process. The resulting top layer is often very thin, namely 100nm or less [8].

The integration process is fundamentally different from the conventional 3D integration, where dies are pre-fabricated and later connected using TSVs. For this reason, M3D is also referred to as sequential 3D, while TSV3D is known as parallel 3D. Figure 1 shows a cross-section of an M3D stack. When the chip is placed on the board, the heat sink is at the top. The layers of an M3D stack are connected by *Monolithic Inter-layer Vias* (MIVs).

2.1 Comparing M3D to TSV3D

2.1.1 Physical Dimensions of Vias. A major advantage of M3D is the very small size of the MIVs. According to CEA-LETI [5, 7, 14], they have a side equal to ≈50nm at the 15nm technology node. This is in contrast to TSVs, which are very large in comparison. Specifically, ITRS projects that TSVs will have a diameter greater than 2.6μm in 2020 [22]. Hence the granularity and placement of TSVs is heavily constrained, whereas MIVs provide great flexibility. To be conservative in our comparisons, this paper will assume an aggressive TSV with half the ITRS diameter, namely 1.3μm.

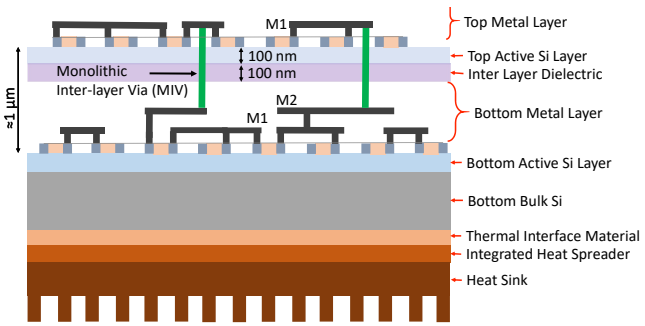

Figure 1: M3D integration of two layers.

Figure 2 shows the relative area of an FO1 inverter, an MIV, an SRAM bitcell, and a TSV at 15nm technology. An MIV uses 0.07x the area of the inverter, while a TSV uses 37x the area of the inverter.

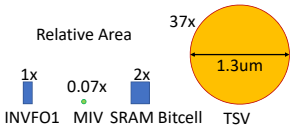

Figure 2: Relative area of an FO1 inverter, an MIV, an SRAM bitcell, and a TSV.

Ultra thin MIVs are possible due to two unique characteristics of M3D integration. First, as shown in Figure 1, the Inter-Layer Dielectric (ILD) and the active silicon layer are very thin (≈100nm) [5]. This is a result of the sequential manufacturing of the top silicon layer. Second, M3D enables very precise alignment of layers through the use of standard lithography tools [6, 14]. Hence, the diameter of an MIV is equal to the pitch of the lowest metal layer.

Table 1 compares the area overhead of an MIV and a TSV to a 32-bit adder and a 32-bit SRAM cell at 15nm technology. The areas of the adder and SRAM cell are obtained from Intel [24, 34]. Note that, because an MIV is so small, it is assumed to be a square. For the TSV, we consider both our aggressive design with a 1.3μm diameter, and the most recent TSV design produced in research [20], which has a 5μm diameter. For the TSV, we add the area of the Keep Out Zone (KOZ) around it; for the MIV, there is no need for a KOZ.

Structure	MIV(50nm)	TSV(1.3um)	TSV(5um)
32bit Adder (77.7 um^2)	<0.01%	8.0%	128.7%
32bit SRAM Cell (2.3 um^2)	0.1%	271.7%	4347.8%

Table 1: Area overhead of an MIV and a TSV compared to a 32-bit adder and a 32-bit SRAM cell at 15nm.

As we can see from Table 1, the MIV area accounts for a negligible overhead for both the 32-bit adder and the 32-bit SRAM cell. In contrast, even the most aggressive TSV implementation has noticeable overheads: its area (plus the KOZ) is equivalent to 8% of an adder or 272% of 32 SRAM cells. Therefore, unlike TSV3D, M3D can provide connectivity to support the ultra-fine partition of components of a core across layers [6, 7, 14].

2.1.2 Electrical Properties. Table 2 shows the capacitance and resistance of an MIV and the two designs of TSV. We obtain the numbers for the 5μm TSV from the literature [15, 20], and use them to estimate the numbers for the 1.3μm TSV.

Parameter	MIV	TSV	
Diameter	$50nm$	$1.3\mu m$	$5\mu m$
Via Height	$310nm$	$13\mu m$	$25\mu m$
Capacitance	$\approx 0.1fF$	$2.5fF$	$37fF$
Resistance	5.5Ω	$100m\Omega$	$20m\Omega$

Table 2: Physical dimensions and electrical characteristics of typical copper MIV and TSVs [15, 20, 45].

MIVs are shorter and thinner than TSVs. As a result, they have a significantly smaller capacitance but a higher resistance. The overall RC delay of the MIV and TSV wires is roughly similar. However, the wire power and the gate delay to drive the wire are mostly dependent on the capacitance of the wire. Both are much smaller in the case of MIVs. For example, Srinivasa et al. [47] show that the delay of a gate driving an MIV is 78% lower than one driving a TSV.

2.1.3 Thermal Properties. TSV3D stacks have die-to-die (D2D) layers in between the dies. Such layers have ≈13-16x higher thermal resistance than metal and silicon layers [1]. Therefore, vertical thermal conductance is relatively limited in TSV3D, and there are substantial temperature differences across layers.

M3D integration requires only a few metal layers in the bottom layer, as they route mostly local wires. Hence, the two active layers in M3D are physically close to each other — typically less than $1\mu m$ apart, even with 3-4 metal layers [2, 25]. Therefore, thermal coupling between the layers is high. In addition, the inter-layer dielectric is only 100nm thick. As a result, vertical thermal conduction is higher than in TSV3D. Hence, the temperature variation across layers is small.

2.2 Partitioning Granularity and Trade-offs

M3D technology is capable of supporting the partitioning of logic and memory structures across layers in a very fine-grained manner [6, 7, 14]. We briefly discuss the trade-offs in selecting the partitioning granularity.

Transistor level (or N/P) partitioning places "N-type" and "P-type" transistors on two different layers. It allows independent optimization of each layer for the type of transistor. It also does not require any metal layers in between the two layers, simplifying the manufacturing process. However, it requires a redesign of standard library cells to use 3D stacked transistors. Further, static CMOS designs require a via for each N/P transistor pair, which results in a ≈10-20% area overhead [28, 29].

Gate level or intra-block partitioning partitions logic or memory blocks at a gate level granularity. Adjacent gates can either be in the same layer or in a different layer. This approach allows the use of standard CMOS libraries and also has a lower via area overhead (at most 0.5% [40]).

Intra-block partitioning into two layers can reduce the footprint of a core by up to 50%. A small footprint reduces the intra-block wire length, and reduces the latency of some critical paths in the core, such as the results bypass path, the load-to-use, and the notification of branch misprediction. It also reduces the length of the clock tree and power delivery networks, and their power consumption.

Block level partitioning partitions the design by placing individual blocks such as ALUs, register files, or instruction decoders, as units in the different layers. It primarily has the same trade-offs as intra-block partitioning. However, there is much less flexibility in 3D routing and, correspondingly, the wire length reductions are much smaller. Further, it delivers no gains when the critical path is within a block as opposed to across blocks.

In this paper, based on the capabilities of M3D, and our desire to keep the analysis at the architecture level, we focus on intra-block partitioning.

2.3 Prior Work on 3D Partitioning

2.3.1 Partitioning for TSV3D. Prior architectural work on partitioning for TSV3D has examined placing cores on top of other cores [16], block level partitioning [11], and intra-block partitioning [41, 42]. In this section, we discuss the intra-block partitioning work, and leave the other, less relevant work, for Section 8.

Puttaswamy and Loh [41] examine several modules within a core and partition them into up to four layers, based on the activity of the gates. Since the gates with the highest activity are likely to consume the most power, the authors place them in the highest layer, which is closest to the heat sink. The goal is to alleviate thermal issues. For example, the least significant bits of an adder are placed in the top layer. However, such partition is not desirable with TSV technology. As we show in Table 1, the area of a single 1.3μm-diameter TSV, together with its KOZ, is equal to 8.0% of an adder's area. Hence, the overhead of the 16 TSVs proposed in [41] would be 128% of the area of the adder itself. This approach would negate any wire delay benefits. The same conclusion is reached by other researchers using 3D floor-planning tools on general logic stages [26, 44]. They find no wire delay reductions due to the high area overhead of TSVs.

Puttaswamy and Loh [42] also examine the 3D partitioning of SRAM array structures to reduce the wire delay of an access. The proposed strategies are bit partitioning (BP), word partitioning (WP), and port partitioning (PP). They are shown in Figure 3. These techniques partition the bits, words and ports, respectively, into two or more layers using TSVs. As indicated above, TSVs take too much area to make these designs attractive. For example, the area of a single SRAM bitcell is $\approx 0.05\mu m^2$ at 14nm [24], whereas the area of a single 1.3μm-diameter TSV, together with its KOZ, is $\approx 6.25\mu m^2$.

2.3.2 Partitioning for M3D. Some researchers have proposed exploiting the multi-layer capabilities of M3D integration to enhance the SRAM structures. Specifically, Srinivasa et al. [47] use the second layer in an M3D design to add column access capability to a regular SRAM cell. Further, in [48], they place a few transistors on top of the SRAM cell either to improve the robustness/noise margins or to provide compute in memory support by performing simple operations such as AND and OR.

Several designs propose to partition the SRAM cell into two levels by placing n-type and p-type transistors on different levels [13, 32]. As we discuss in Section 2.2, we choose to partition the design at a gate level, which has different tradeoffs than transistor-level partitioning.

Kong et al. [27] study the benefits of using M3D integration to build large SRAM arrays such as the last-level cache. They focus on large single-ported structures. In this paper, we focus on partitioning the processor core, where several key SRAM structures are small and multi-ported.

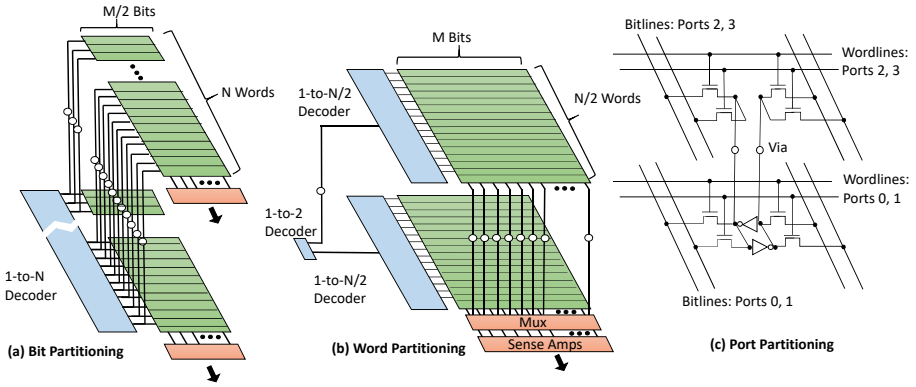

Figure 3: Partitioning an SRAM array using bit partitioning (a), word partitioning (b), and port partitioning (c). The figure is taken from [42].

Most of these works [27, 47, 48] use CACTI [4] to obtain the access energy and delay of SRAM structures. We use the same tool.

2.4 M3D: Opportunities and Challenges

2.4.1 Opportunities. Modern core designs are constrained due to the historically slower scaling of wire delay relative to transistor delay. This is evident in wire-dominated structures such as SRAM arrays and wire-dominated critical paths such as the results bypass path. M3D integration provides a great opportunity to reduce the wire lengths and therefore the delays by partitioning at gate-level granularity.

M3D integration allows the optimization of the manufacturing process of each layer separately, to attain different power-performance points. For example, the bottom layer can use bulk transistors for a high-performance (HP) process, whereas the top layer can use the slower but lower power FDSOI transistors. This offers an opportunity for power savings beyond the simple choice of transistors with different V_t.

2.4.2 Challenges. The primary challenge for M3D is manufacturability issues. The top layer in M3D is fabricated sequentially on top of the bottom one. This step usually involves high temperature, and may damage the bottom layer's devices and interconnects. One option is to use a tungsten-based interconnect in the bottom layer, as it has a higher melting point than copper [5, 14]. Unfortunately, tungsten has 3x higher resistance than copper and results in a significant wire delay increase. Further, the use of tungsten may still not be sufficient, as the bottom layer transistors can still be damaged.

Alternatively, the top layer can be processed at a significantly lower temperature, using laser-scan annealing techniques [35, 43]. However, the M3D IC manufactured using this process showed a performance degradation of 27.8% and 16.8% for PMOS and NMOS devices, respectively [43]. A more recent study estimates that the delay of an inverter in the top layer degrades by 17% [45]. As a result, the authors found that gate-level partitioning of LDPC and AES blocks causes their frequency to go down by 7.5% and 9%, respectively. Overall, the lower performance of the top layer poses challenges to the partitioning of the core.

A second challenge is a scarcity of CAD tools for 3D, which are currently being developed [12, 39]. In this paper, we do not address this challenge.

3 PARTITIONING A CORE IN M3D

In this paper, we examine how to partition a core into two layers in M3D. For now, we assume that both M3D layers have the same performance. We present a hetero-layer design in Section 4. We consider in turn the logic stages, storage structures, and other structures.

3.1 Logic Stages

The wires in a logic pipeline stage can be classified into local, semi-global, and global. Local wires connect gates that are close to each other. These wires comprise most of the intra-stage critical path wire delay. To optimize these wires, the best approach is to use CAD tools for place and route. It has been shown that 3D floor-planners customized for M3D integration reduce the lengths of local wires by up to 25% [38, 44]. There is little scope for further optimization of local wires using micro-architectural insights.

Semi-global wires connect one logic block to another logic block within a stage. These wires are often critical to performance from a micro-architectural viewpoint. Some examples are wires in the micro-architectural paths that implement the ALU plus bypass network, the load to use, and the branch misprediction notification. M3D integration of a pipeline stage can reduce the footprint of the stage by up to 50% — in this case reducing the distance traversed by the semi-global wires by up to 50%. Like in Black et al. [11], we estimate that a 3D organization reduces the latency of the load to use and branch misprediction notification paths, possibly saving one cycle or more in each path.

Global wires span a significant section of the chip and may take multiple clock cycles. The global footprint reduction reduces the length of a global wire. An example of global wire is a link in a Network-on-Chip (NoC). If we manage to fold each core into about half of its original area, two cores can share a single NoC router stop (Figure 4). In this case, we halve the distance between neighboring routers and reduce the number of hops. This design reduces the average network delay for the same number of cores.

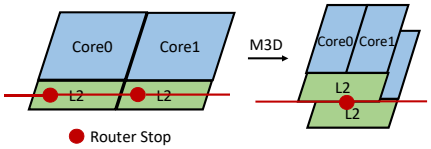

Figure 4: Two cores sharing the L2s and the router stop.

To verify the impact of wirelength reduction in logic stages, we synthesize and lay out a 64-bit adder along with a bypass path in 45nm technology. We use the M3D place and route tools developed by Lim et al. [39, 44]. The results show that a two-layer M3D implementation achieves a 15% higher frequency. Moreover, the footprint reduction observed is 41%. This reduction is in line with numbers reported elsewhere [38, 44]. If we lay out multiple ALUs with their bypass paths, the contribution of wire delay toward the stage delay is higher, since the length of the bypass path increases quadratically with the number of ALUs. In the case of four ALUs with bypass paths, we estimate a 28% higher frequency, 10% lower energy, and 41% lower footprint than a 2D design. Further, we note that, at the 15nm technology node that we consider in this paper, the wire delay contribution is higher and, therefore, the frequency gain would be higher.

3.2 Storage Structures

The storage structures in a core consist of SRAM structures such as the register file and branch prediction table, and CAM structures such as the issue queue and load/store queue. CAM and RAM structures are structurally very similar in their layout. Therefore, we treat them similarly for the purpose of partitioning them in 3D.

An SRAM array is given by its height and width. The height is the number of words (N_{words}). The width is the number of bits per word (N_{bits}). A wordline is as long as the width of the array; a bitline is as long as the height of the array. As we partition an array into two layers, we keep in mind two basic rules. First, the area is proportional to the square of the number of ports. Second, both the array access latency and the energy consumed depend in large measure on the length of the wordlines and bitlines.

We model the partitioning of the SRAM arrays using CACTI [4]. We use high performance (HP) transistors and, to be conservative, 22nm technology parameters. MIV and TSV overheads are modeled using 50nm and 1.3μm diameters, respectively, as per Section 2.1.1.

We partition the following SRAM arrays in the core: register file (RF), issue queue (IQ), store queue (SQ), load queue (LQ), register alias table (RAT), branch prediction table (BPT), BTB, data and instruction TLB, data and instruction L1, and L2 cache. We partition them using bit partitioning (BP), word partitioning (WP), and port partitioning (PP) (Section 2.3.1), and measure the reduction (or increase) in access latency, access energy, and area footprint compared to a 2D design. As examples, we describe the partitioning of a register file and a branch prediction table. The former has 160 words of 64 bits, 12 read ports, and 6 write ports. The latter has 4096 words of 8 bits and 1 port.

3.2.1 Bit Partitioning (BP). BP spreads half of each word in each layer, and places a driver in each layer. As a result, the effective length of each wordline is halved. Each word requires a via across the layers (Figure 3(a)). Table 3 shows the percentage of improvement we attain by bit partitioning our two structures using M3D and TSV3D, compared to a 2D structure.

From the table, we observe that M3D performs better than TSV3D in all metrics. This is expected, as the diameter of an MIV is smaller than that of a TSV. Furthermore, we see that the gains in the multi-ported register file are higher than in the single-ported branch prediction table. There are two reasons for this fact, both of which

	Register File (RF)			Branch Pred. Table (BPT)		
	Laten.	Ener.	Footpr.	Laten.	Ener.	Footpr.
M3D	28%	22%	40%	14%	15%	37%
TSV3D	25%	19%	31%	4%	-3%	4%

Table 3: Percentage reduction in access latency, access energy, and area footprint through bit partitioning.

are related to the larger area required by multi-ported structures. First, when the area is large, the wire component of the SRAM access delay is relatively higher; hence, partitioning the structures into layers is relatively more beneficial. Second, when the area is large, the overhead of the vias becomes less noticeable, which results in higher improvements for partitioning. TSV3D only marginally improves the BPT due to the large size of the TSVs.

3.2.2 Word Partitioning (WP). WP spreads half of the words in each layer, and places a driver in each layer. As a result, the effective length of each bitline is halved. The number of vias needed is equal to the array width (Figure 3(b)). Table 4 shows the percentage of improvement we attain by word partitioning our two structures using M3D and TSV3D, compared to a 2D structure.

	Register File (RF)			Branch Pred. Table (BPT)		
	Laten.	Ener.	Footpr.	Laten.	Ener.	Footpr.
M3D	27%	35%	43%	14%	36%	57%
TSV	24%	32%	39%	-6%	9%	19%

Table 4: Percentage reduction in access latency, access energy, and area footprint through word partitioning.

The observations from BP hold true for WP as well. WP and BP are both affected in similar ways by the larger area induced by multiple ports, and the larger size of TSVs. However, in general, BP partitioning is preferable over WP because we especially want to reduce the access latency, and wordlines are responsible for more delay than bitlines. Interestingly, the branch prediction table is an exception: WP proves to be a better design than BP in M3D. The reason is the aspect ratio of the branch prediction table's array. The array's height is much longer than its width. Hence, WP's ability to halve the bitlines delivers significant savings.

3.2.3 Port Partitioning (PP). PP places the SRAM bit cell with half of its ports in one layer and the rest of the ports with their access transistors in the second layer. It needs two vias per SRAM bit cell as shown in Figure 3(c). Table 5 shows the percentage of improvement we attain by port partitioning our two structures using M3D and TSV3D, compared to a 2D structure.

	Register File (RF)			Branch Pred. Table (BPT)		
	Laten.	Ener.	Footpr.	Laten.	Ener.	Footpr.
M3D	41%	38%	56%	-	-	-
TSV	-361%	-84%	-498%	-	-	-

Table 5: Percentage reduction in access latency, access energy, and area footprint through port partitioning.

Generally, halving the number of ports is an excellent strategy: it reduces both the wordline length and the bitline length nearly by half. In M3D, this effect reduces the latency, energy, and area by a large fraction. As can be seen in Table 5, the improvements in the RF are large. Of course, PP cannot be applied to the BPT because the latter is single-ported.

M3D can use PP because MIVs are very thin. It is possible to place two vias per RF SRAM cell — especially since a multiported SRAM cell is large. However, TSVs are too thick to be used in PP. As shown in Table 5, the cell area increases by 498%, creating large increases in access latency and energy.

Table 6 shows the best partitioning strategy that we find for each SRAM structure in the core that we evaluate in Section 6 — both for M3D and TSV3D. The table shows the percentage of reduction that we attain in access latency, access energy, and footprint compared to a 2D structure. Our preferred choice are designs that reduce the access latency. With this in mind, we conclude that, for M3D, PP is the best design for multiported structures, while BP is usually the best one for single-ported structures. The exception to the latter is when the SRAM array has a much higher height than width, in which case WP is best. TSV3D is less effective, and is not compatible with PP.

Structure [Words; Bits per Word] × Banks	Best Partition		Latency Reduc.(%)		Energy Reduc.(%)		Footprint Reduc.(%)	
	M3D	TSV.	M3D	TSV.	M3D	TSV.	M3D	TSV.
RF [160; 64]	PP	BP	41	25	38	19	56	31
IQ [84; 16]	PP	BP	26	17	35	5	50	32
SQ [56; 48]	PP	BP	14	-3	21	-18	44	0
LQ [72; 48]	PP	BP	15	2	36	8	48	10
RAT [32; 8]	PP	WP	20	10	32	5	45	-11
BPT [4096; 8]	WP	BP	14	4	36	-3	57	4
BTB [4096; 32]	BP	BP	15	-6	20	-10	37	-20
DTLB [192; 64] ×8	BP	BP	26	18	28	20	35	22
ITLB [192; 64] ×4	BP	BP	20	7	28	11	36	11
IL1 [256; 256] ×4	BP	BP	30	14	36	23	41	25
DL1 [128; 256] ×8	BP	BP	41	31	40	33	44	34
L2 [512; 512] ×8	BP	BP	32	24	47	42	53	46

Table 6: Best partition method for each structure, and percentage reduction in latency, energy and area footprint.

Finally, since the distance from a core to another core and its L2 cache is now reduced, in both 3D designs, two cores now share their two L2 caches as shown in Figure 4.

3.3 Clock Tree and Power Delivery Network

The clock-tree network and the power delivery network (PDN) only have to cover about half of the footprint of a 2D design. Since the clock tree consumes substantial dynamic power, the power savings due to the reduced footprint can be significant. There are two options for designing the PDN in M3D chips. One option is to give each of the two layers its own PDN. This design increases the number of metal wires, which increases both the via routing complexity and the cost. Alternatively, one can use a single PDN that is present in the top layer and then supply power to the bottom layer through MIVs. Billoint et al. [10] suggests that this second approach is preferable.

4 HETERO-LAYER PARTITIONING

Low temperature processing of the top layer in M3D causes the top layer to have lower performance than the bottom one. As indicated in Section 2.4.2, Shi et al. [45] found that the delay of an inverter in the top layer is 17% higher than in the bottom one. Consequently, we modify the core partitioning algorithms of Section 3 to alleviate the impact of the slowdown. With these algorithms, we design a *hetero-layer M3D core*.

Our approach is shown in Table 7. In logic pipeline stages, we identify the critical paths in the stage and place them in the bottom layer. The non-critical paths are placed in the top layer and do not slow down the stage. This is possible because more than 60% of the transistors in a typical stage are high V_t, and fewer than 25% are low V_t [3] (the rest are regular V_t). Hence, there are always many non-critical paths.

Structure	Partitioning Technique	
Logic Stage	Critical paths in bottom layer; non-critical paths in top	
Storage Structure	Port Partitioning	Asymmetric partitioning of ports, and larger access transistors in top layer
	Bit or Word Partitioning	Asymmetric partitioning of array, and larger bit cells in top layer
Mixed Stage	Combination of the previous two techniques	

Table 7: Partitioning techniques for a hetero-layer M3D core.

In storage structures, the critical path spans the entire array. Hence, we cannot use the same approach as for the logic. Instead, we use two separate techniques based on the partitioning strategy applied in Section 3.2. Specifically, in port partitioning, we exploit the fact that the two inverters in the SRAM bit cell are in the bottom layer (Figure 3(c)). Hence, we partition the ports asymmetrically between the two layers, and increase the sizes of the access transistors in the ports in the top layer, which increases their speed. In bit/word partitioning, we partition the array asymmetrically between the layers, giving a smaller section to the top layer. Further, we use the area headroom in the top layer to increase the sizes of the bit cells. Finally, in mixed stages, we combine the two techniques. In this section, we present these techniques.

4.1 Logic Stages

We analyze an out-of-order superscalar processor and identify three mostly-logic stages: decode, dispatch, and execute in integer and FP units. We partition them as per Table 7. We now give two examples.

4.1.1 Partitioning an Integer Execution Unit. Figure 5 shows a 64-bit carry skip adder. The critical path is shown shaded. It consists of a carry propagate block, a sum block, 15 muxes and a final sum block. The majority of the blocks are not in the critical path — i.e., the remaining 15 4-bit carry propagate blocks and 14 sum blocks. The farther away a propagate block is from the LSB, the higher slack it has. Therefore, we place the carry propagate blocks of bits {32:63} and the sum blocks of bits {28:59} in the top layer. There is no impact on the critical path and hence the stage delay.

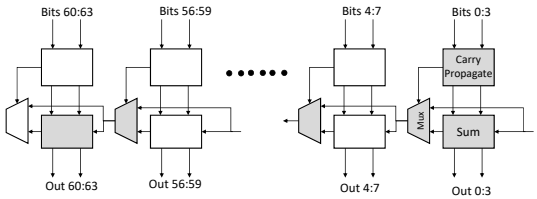

Figure 5: ALU with shaded critical-path blocks.

Using our M3D place and route tools of Section 3.1, we find that only 1.5% of the gates in the 64-bit adder are in the critical path. We place them in the bottom layer. It can be shown that, even if

we assumed that the top layer was 20% slower — and, hence, we needed a 20% slack — we would only have 38% of the gates in the critical path. Hence, we can always find 50% of gates that are not critical and place them in the top layer.

4.1.2 Partitioning Decode. Modern x86 processors have a set of simple decoders and a complex decoder. Most common instructions that translate into a single μop are processed by the simple decoders. More complex and uncommon instructions utilize the complex decoder and, occasionally, a special μcode ROM to generate multiple μops. In our hetero-layer M3D design, we place the simple decoders in the bottom layer. The complex decoder and the μcode ROM are placed in the top layer and take one additional cycle. The μcode ROM access already takes multiple cycles.

4.2 Storage Structures

4.2.1 Port Partitioning (PP). As shown in Table 6, PP is the best strategy for multiported arrays such as the RF, IQ, SQ, LQ, and RAT. In a port-partitioned cell, the two inverters are left in the bottom layer, while the ports are divided between the two layers. In a hetero-layer M3D, we assign fewer ports to the top layer than to the bottom one, and double the width of transistors of the ports in the top layer. The goal is to make the top layer's transistors as fast as the bottom layer ones and still use the same footprint in both layers.

We measure that the area of the two inverters in a bitcell is comparable to that of two ports. However, the optimal port partitioning depends on the number of ports. For example, consider a register file with 12 read and 6 write ports. We find that the partition with the smallest footprint places 10 ports in the lower layer and 8 ports (with double-width transistors) in the top one. With this partition, Table 8 shows that the register file uses 47% less area than in a 2D layout. This is 9 fewer percentage points than the partition for same-performance M3D layers (Table 6).

	RF (%)	IQ (%)	SQ (%)	LQ (%)	RAT (%)	BPT (%)	BTB (%)	DTLB (%)	ITLB (%)	IL1 (%)	DL1 (%)	L2 (%)
Latency	40	24	13	13	20	13	13	23	18	27	37	29
Energy	32	30	17	30	24	30	16	25	25	33	36	42
Area	47	47	43	47	44	40	26	25	28	30	31	42

Table 8: Percentage reduction in access latency, access energy, and area footprint with the best hetero-layer partitioning compared to a 2D layout.

The wider access transistors alleviate the impact of the bitline delay in the top layer. However, they increase the capacitance on the wordlines slightly. This increases the cell access energy and wordline delay slightly. We measured the resulting access latency, access energy, and footprint of the RF, IQ, SQ, LQ, and RAT structures. Table 8 shows the savings compared to a 2D structure. These are substantial reductions. Compared to the partition for same-performance layers in Table 6, the numbers are only slightly lower.

4.2.2 Bit/Word Partitioning (BP/WP). Our technique to alleviate the impact of the slower top layer in structures using BP/WP consists of two steps. First, we perform BP/WP asymmetrically across layers, giving a larger section of the array to the bottom layer. Next, we use larger transistors in the top layer. We tune these two operations to obtain the minimum access latency, while tolerating a less improved access energy and footprint. In general, a partition that gives 2/3 of the array to the bottom layer, together with doubling the transistor widths in the top layer works well. Table 8 shows the reductions of access latency, access energy, and footprint of these structures compared to a 2D layout. Again, these are large reductions, only slightly smaller than those in the same-performance partition (Table 6).

In Table 8, we see that the L1 and L2 caches have large latency reductions. Since the core's frequency is determined by the slowest pipeline stage, we can tune the caches' partitions to save more on footprint at the expense of less on access latency.

4.3 Stages with Logic and SRAM Structures

Most storage structures are part of a stage that also contains logic components. We discuss the modifications to such stages in two parts: this subsection covers SRAMs and the next one CAMs. In each part, we discuss two stages as examples.

4.3.1 Rename. The rename stage reads from and writes to the Register Alias Table (RAT), which is a multiported structure. We use PP for the RAT as per Section 4.2.1. In parallel to the RAT access, a dependence check is performed among the registers being renamed. This check is not in the critical path [37]. Hence, we place this checking logic and the shadow RAT tables used for checkpointing in the top layer. We place other critical structures such as the decoder to the RAT's RAM array in the bottom layer.

4.3.2 Fetch & Branch Prediction. The fetch unit mainly consists of accessing the IL1 cache and computing the next PC. The IL1 cache uses BP as per Section 3.2. Computing the next PC has a few different parallel paths: BTB access, branch prediction, Return Address Stack (RAS) access, and incrementing PC. Of these, only branch prediction and BTB access are critical to stage delay. Hence, we place the RAS and the PC increment in the top layer.

Since the BTB is critical, we use asymmetric BP coupled with larger transistors in the top layer. As shown in Table 8, we reduce its access energy by 16% compared to a 2D layout, which is 4 percentage points fewer than the partition for same-performance M3D layers (Table 6).

Our core employs a tournament branch predictor, which contains a selector table indexed by a hash of PC and global branch history. The selector output drives a mux that chooses between a local and global predictor. We observe that the critical path is formed by the selector and mux, and not by the local or global predictors. Therefore, we propose an organization where we use asymmetric BP for the selector, local predictor, and global predictor. However, we place the larger section of the selector array in the bottom layer, and the larger sections of the two predictors in the top layer. With this design, we save 40% of the footprint relative to a 2D layout (BPT entry in Table 8).

4.4 Stages with Logic and CAM Structures

CAM arrays are similar to SRAM arrays, except that they include an additional wire per cell called the *Match* line, which is connected to the cell. A few additional transistors associated with the cell (usually 4) pull the match line low when the bit stored in the cell is different than the one being compared to it (i.e., the *Tag* bit). The critical path in this structure is the time it takes to drive the tag

line, plus the time for the match line to go low, plus any peripheral logic that operates on the output of the match lines.

In a core, CAM structures are found in the IQ, LQ, SQ, and the tag arrays of caches. For the tag arrays of caches, we use the same organization as the associated cache, namely, BP. The IQ is multi-ported with as many ports as the issue width, and the LQ and SQ have two ports each. These structures use asymmetric PP and larger transistors in the top layer.

4.4.1 Issue. Issue in modern processors consists of two pipeline stages, namely *Wakeup* and *Select*. During the wakeup stage, the IQ is accessed to mark the dependent operands as ready. The dependent operands determine the instructions that can be executed next. The IQ is a CAM structure and is partitioned as indicated above.

During the select stage, the select logic is activated to pick the instructions to execute. The selection logic is made up of multi-level arbitration steps, and consists of two phases: a *Request* phase in which the ready signal is propagated forward from each arbiter, and a *Grant* phase in which one of the requesters is selected. At a particular arbitration level, the generation of the grant signal is in turn decomposed into two parts. First, the local priorities at this level are compared and one requester is selected for the grant signal. We call this part of grant phase as the *local grant generation*. Second, the local grant signal from this requester is ANDed with an incoming grant signal from a higher level in the arbitration hierarchy, and an output is generated. We call this part as the *arbiter grant generation*. Such an organization minimizes the critical path delay in selection logic [36].

The first part of grant phase, i.e., the local grant generation, is not critical. Hence, we place this logic in the top M3D layer. The second part, i.e., the arbiter grant generation, when the grant signals are ANDed and propagated, is critical. Further, the entire *Request* phase is critical. Therefore, we place the arbiter grant generation logic and the request phase logic in the bottom layer. With this, it can be shown that the select stage has the same latency as in the partition for same-performance layers, without increasing the area or power.

4.4.2 Load Store Unit. The LQ and SQ in the Load Store Unit (LSU) are CAM structures searched by incoming store and load requests. When a load searching the SQ hits, the value from the youngest matching store (that is older than the load) is forwarded to the load. This comprises the critical path in an LSU [9]. The search in the LQ, and the corresponding squash on a match are not critical to the stage delay. Therefore, the critical path in the stage consists of the CAM search of the SQ, a priority encoder to find the youngest store, and the read from the store buffer. For the SQ, we use the usual PP methodology with bigger transistors in the top layer. We place the priority encoder in the bottom layer. The store buffer uses asymmetric BP with more bits in the bottom layer. Finally, the less critical LQ uses asymmetric PP, occupying more area in the top layer. Compared to the partition for same-performance layers in Table 6, this design of the SQ and LQ attains roughly similar footprint reductions, and only a slight increase in access latencies and energy (Table 8).

5 ARCHITECTURES ENABLED BY M3D

Based on the previous discussion, we observe that M3D enables several types of architectures.

1. Exploiting Wire Delay Reduction in Conventional Cores. One approach is to exploit the M3D-enabled reduction in wire delay in conventional cores. This approach can be followed in three ways. First, wire delay reductions can be translated into cycle-time reductions, which allow higher core frequencies. However, this approach increases power density.

Alternatively, one can increase the sizes or the number of ports for some of the storage structures, while maintaining the same frequency as a 2D core. Note that most of the structures that are bottlenecks in wide-issue cores benefit significantly from M3D. They include multi-ported issue queues, multi-ported register files, and bypass networks. Therefore, one can increase the issue width of the core while maintaining the same frequency.

Finally, another alternative design is to operate the M3D design at the same frequency as the 2D core, and lower the voltage. Reducing the voltage lowers the power consumption and the power density. The M3D design can now operate more cores for the same power budget. We evaluate these three designs in Section 7.2.

2. Hetero M3D design. We have partitioned a core assuming heterogeneous M3D layers. However, it is possible that, in the future, M3D may support same-performance layers. Even in this case, our partitioning techniques may be applicable. For example, one may choose to build the two layers with different technology to save energy: place high performance bulk transistors in the bottom layer, and low performance FDSOI transistors in the top one. We evaluate such a scenario in Section 7.1.2.

3. Novel Architectures. In specialized architectures, it is often necessary to provide a tight integration between specialized engines and general-purpose cores, to support fine-grain communication between the two. However, such architectures have to make compromises in a 2D design. M3D integration facilitates such tight integration, e.g., by supporting a set of accelerators in the top layer without compromising the layout of general-purpose cores in the bottom layer.

Further, M3D technology allows the integration of heterogeneous process technologies, such as Non-Volatile Memory (NVM) on top of regular logic cells [46]. This ability enables new computing paradigms that can take advantage of large amounts of NVM very close to the compute engines.

6 EVALUATION SETUP

We evaluate the performance of our designs using the Multi2Sim [49] architectural simulator. We model a multicore with 4 cores. Each core is 6-issue wide and out of order. Table 9 shows the detailed parameters of the modeled architecture. We model the SRAM and CAM arrays using CACTI. We obtain the power numbers of the logic stages by using the HP-CMOS process of McPAT [30]. We set the nominal voltage at 22nm to 0.8V following ITRS [22]. Using this voltage, different processor designs can attain different frequencies, as we discuss in Section 6.1.

We model 3D storage structures using CACTI as follows. First, we partition the structure's bits, words, or ports into two layers

Parameter	Value
Cores	4 out-of-order cores, V_{dd}=0.8V
Core width	Dispatch/Issue/Commit: 4/6/4
Int/FP RF; ROB	160/160 registers; 192 entries
Issue queue	84 entries
Ld/St queue	72/56 entries
Branch pred.	Tournament, with 4K entries in selector, in local predictor, and in global predictor; 32-entry RAS
BTB	4K-entry, 4-way
FUs & latencies:	
4 ALU	1 cycle
2 Int Mult/Div	2/4 cycles
2 LSU	1 cycle
2 FPU	Add/Mult/Div: 2/4/8 cycles; Add/Mult issue every cycle; Div issues every 8 cycles
Private I-cache	32KB, 4-way, 32B line, Round-trip (RT): 3 cycles
Private D-cache	32KB, 8-way, WB, 32B line, RT: 4 cycles
Private L2	256KB, 8-way, WB, 64B line, RT: 10 cycles
Shared L3	Per core: 2MB, 16-way, WB, 64B line, RT: 32cycles
DRAM latency	RT after L3: 50ns
Network	Ring with MESI directory-based protocol

Table 9: Parameters of the simulated architecture.

based on the partitioning strategies of Section 3.2 (i.e., BP, WP, and PP). Next, we compute the number of vias needed to connect the two layers in each of the three partitioning strategies. Based on the number of vias, we then calculate the total via overhead, and estimate the increase in dimensions of the SRAM cell and the SRAM array. In the case of TSVs, we also perform further layout optimizations by considering different via placement schemes to minimize the overhead.

After computing the array dimensions, we estimate the capacitance and the resistance of the wordlines and bitlines for the entire array. Based on these values, we obtain the delay, energy, and area estimates of the 3D storage structure by using regular CACTI functionality. Finally, we also verify the projected benefits of 3D structures over 2D structures for TSVs and some M3D technologies by comparing them with previously-published results.

We estimate the power consumed by a 3D logic structure as follows. First, we obtain the power consumption of the corresponding 2D structure. We then take the switching power of such a structure and reduce it by a factor that is equal to the reduction in the switching power of the ALU circuit discussed in Section 3.1. Recall that the ALU circuit was synthesized and laid out using M3D place and route tools [39, 44]. For the clock tree, the process is different. For the clock tree, we reduce the switching power by a constant factor of 25% [42]. Finally, we keep the leakage power of the structure unchanged.

For all the M3D and TSV3D designs, we leverage the reduction in the latency of logic paths, as indicated in Section 3.1. Specifically, compared to 2D designs, we reduce the latency of the load-to-use and the branch misprediction paths by 1 cycle and 2 cycles, respectively, out of the 4 and 14 cycles taken by these two paths in 2D designs. This optimization increases the IPC of the 3D designs.

We model M3D with the layers shown in Figure 1. The core blocks are partitioned across the two silicon layers. Note that M3D requires only 3-4 stacked metal wires in the bottom metal layer [2]. Moreover, the inter-layer dielectric is very thin. As a result, the distance between the two active silicon layers is ≈1μm [25].

The dimensions and thermal conductivity of the different layers are shown in Table 10. These values are obtained from recent work on thermal models for TSV3D [1]. Note that, currently, silicon layers in TSV3D have a thickness of at least 100μm. However, we set the thickness of the top silicon layer in TSV3D to a very low 20μm, to model an aggressive futuristic design. This assumption makes our numbers for TSV3D optimistic.

Layer	M3D Dimensions	TSV3D Dimensions	Thermal Conductivity
Top Metal	12 μm	12 μm	12 W/m-K
Top Silicon	100nm	20μm	120 W/m-K
ILD	100nm	20 μm	≈1.5 W/m-K
Bottom Metal	<1μm	12μm	12 W/m-K
Bottom Silicon	100μm	100μm	120 W/m-K
TIM	50 μm	50 μm	5 W/m-K
IHS	3.0x3.0x0.1 cm^3	3.0x3.0x0.1 cm^3	400 W/m-K
Heat Sink	6.0x6.0x0.7cm^3	6.0x6.0x0.7cm^3	400 W/m-K

Table 10: Thermal modeling parameters for M3D and TSV3D. In the table, ILD, TIM, and IHS mean Inter-Layer Dielectric, Thermal Interface Material, and Integrated Heat Spreader.

We use HotSpot's extension [19, 33] to model the effects of heterogeneous components in the same layer. We model both lateral and vertical thermal conduction using the more accurate grid-model.

We evaluate M3D and TSV3D designs for both single cores and multicores. For the single-core experiments, we run 21 SPEC2006 applications; for the multicore ones, we run 12 SPLASH2 applications and 3 PARSEC ones.

6.1 Architecture Configurations Evaluated

We compare several architecture configurations. In the following, we refer to M3D with same-performance layers as *iso-layer* M3D, and M3D with different-performance layers as *hetero-layer* M3D.

2D Baseline. Traditionally, the clock cycle time of a microprocessor has been limited by the wakeup plus select operations in the issue stage, or by the ALU plus bypass paths [37]. However, in recent processors, the wakeup and select steps are split into two stages [17]. Further, the register file access has emerged as a key bottleneck for wide-issue cores in addition to the ALU plus bypass paths. Of all the core structures we discussed, we measure with CACTI that the one that limits the core cycle time is the access time of the register file. Based on our measurements, we set the frequency of our baseline 2D core (*Base*) to 3.3 GHz (Table 11).

Iso-layer M3D. Table 6 shows the reduction in the access latency of different structures in iso-layer M3D relative to 2D. We see that the access latency of the RF and IQ decrease by 41% and 26%, respectively. Further, in Section 3.1, we estimate that four ALUs with bypass paths can sustain a 28% higher frequency. If we consider the traditional frequency-critical structures (similar to [41]), we find that the frequency is limited by the reduction of IQ access delay at 26%. The frequency is then 3.3/(1-0.26)=4.46 GHz. We call this design, *M3D-IsoAgg*, but do not evaluate it due to space limits.

Instead, to be very conservative, we assume that all the array structures in Table 6 are in the critical path. In particular, we assume that the BPT and BTB arrays need to be accessed in a single cycle.

Name	Configuration
Single Core	
Base	Baseline 2D, f=3.3GHz
M3D-Iso	Iso-layer M3D, f=3.83GHz
M3D-HetNaive	Hetero-layer M3D without modifications, f=3.5GHz
M3D-Het	Hetero-layer M3D with our modifications, f=3.79GHz
M3D-HetAgg	Aggressive M3D-Het, f=4.34GHz
TSV3D	Conventional TSV3D, f=3.3GHz
MultiCore	
M3D-Het	M3D-Het + Shared L2s, 4 cores, f=3.79GHz
M3D-Het-W	M3D-Het + Shared L2s, Issue=8, 4 cores, f=3.3GHz
M3D-Het-2X	M3D-Het + Shared L2s, 8 cores, f=3.3GHz, V_{dd}=0.75V
TSV3D	Conventional TSV3D + Shared L2s, 4 cores, f=3.3GHz

Table 11: Core configurations evaluated.

Based on this assumption, we identify the structure with the least reduction in access time, i.e., SQ and BPT with 14%. With this estimate, we set the frequency of our *M3D-Iso* core to 3.3/(1-0.14)=3.83GHz (Table 11).

TSV3D. The corresponding numbers for TSV3D in Table 6 are sometimes negative. Hence, TSV3D may result in a core slowdown. The large footprint of TSVs makes intra-block 3D partitioning undesirable. Therefore, we keep the frequency of the *TSV3D* core the same as the 2D *Base* (Table 11). However, like all the other 3D designs, *TSV3D* has a lower load-to-use and branch misprediction path latencies compared to *Base* (Section 6).

Hetero-layer M3D. We consider three designs. The first one, called *M3D-HetNaive*, simply takes the *M3D-Iso* design and slows its frequency by 9% — which is the loss in frequency estimated by Shi et al. [45] in an AES block due to the slower top layer (Section 2.4). Hence, we set the frequency of the *M3D-HetNaive* core to 3.83×0.91 ≈ 3.5GHz (Table 11).

The second design, called *M3D-Het*, is the result of our asymmetric partitioning of structures in Section 4. We consider all the array structures in Table 8, and take the one that reduces the access latency the least. Specifically, the SQ, LQ, BPT, and BTB only reduce the access latency by 13% relative to 2D. Consequently, we set the frequency of the *M3D-Het* core to 3.3/(1-0.13)≈3.79GHz (Table 11). Finally, we evaluate another design, *M3D-HetAgg*, that is derived in a manner similar to M3D-IsoAgg: the frequency is limited by the reduction in IQ access time, which is 24% in this case. The corresponding frequency is 3.3/(1-0.24)≈4.34GHz (Table 11).

Multicore hetero-layer M3D. We consider several multicore designs, shown in Table 11. In this case, pairs of cores share their L2 caches as in Figure 4. We consider a 4-core *M3D-Het* and two related designs we describe next: a 4-core *M3D-Het-W* (where W stands for wide) and an 8-core *M3D-Het-2X* (where 2X stands for twice the cores). We also evaluate a 4-core *TSV3D* where cores share L2 caches (Table 11).

To configure M3D-Het-W, we take M3D-Het, set its frequency to that of the 2D Base core (3.3GHz), and increase the core's width as much as possible. The maximum width is 8. To configure M3D-Het-2X, we again take M3D-Het, set its frequency to 3.3GHz, and reduce the voltage as much as possible. Following curves from the literature [18, 23], the maximum reduction is 50mV, which sets the voltage to 0.75V. At this point, the multicore consumes relatively little power. Hence, we increase the number of cores as much as possible until it reaches the same power consumption as *four cores* of our 2D *Base*. The number of cores is in between 7 and 8. We pick 8 as some parallel applications require a power-of-two core count.

7 EVALUATION

We organize the evaluation in two parts. First, we consider single core designs, and then multicore designs.

7.1 Single Core M3D Designs

We consider performance, energy, and thermals in turn.

7.1.1 Performance. Figure 6 shows the speed-up of different single-core M3D designs over *Base* for SPEC2006 applications. The figure shows a set of bars for each application and the average. For each application, there is a bar for *Base*, *TSV3D*, *M3D-Iso*, *M3D-HetNaive*, *M3D-Het*, and *M3D-HetAgg*. The bars are normalized to *Base*.

M3D-Iso, where both layers have the same performance, is on average 28% faster than *Base*. The performance improvement is due to two reasons. First, *M3D-Iso* executes at a 16% higher frequency than *Base*. Second, as indicated in Section 6, some of the critical pipeline paths, such as the load-to-use and the branch-misprediction paths, are shorter. Therefore, the IPC is higher as well. Note that this is despite the increase in memory latency in terms of core clocks. *M3D-HetNaive* has more conservative assumptions [45], and operates at only 6% higher frequency than *Base*. Even with its higher IPC, the speedup is only 1.17 over *Base*.

The critical path optimizations that we proposed in Section 4 prove useful for *M3D-Het*. The average stage delay is now 13% shorter than in *Base*. As a result, *M3D-Het*'s performance is very close to that of *M3D-Iso*. In effect, *M3D-Het* recovers most of the performance lost due to the slower top layer. On average, *M3D-Het* is 25% faster than *Base*. Finally, the aggressive M3D configuration, *M3D-HetAgg*, operating at 32% higher frequency than *Base*, provides a speed-up of 1.38 over *Base* on average. This improvement is substantial.

TSV3D, which operates at the same frequency as *Base*, is only 10% faster than *Base*. The gains are due to the reduction in the critical path latencies that we discussed above.

7.1.2 Energy. Figure 7 shows the energy consumption of different M3D designs normalized to *Base*. The designs considered and the organization of the figure are the same as in Figure 6. We can see that all of the M3D designs consume, on average, about 40% less energy than *Base*. This is due to a few factors. First, as shown in Table 6, the energy consumption of many SRAM structures is significantly smaller. Second, the footprint of the clock tree network is only about half that of *Base*. Finally, M3D designs execute faster, saving leakage power on top of the previous reductions.

On average, *M3D-Iso* consumes 41% lower energy than *Base*. The simple *M3D-HetNaive* has a similar power consumption as *M3D-Iso*, but executes for longer. Therefore, the average energy consumed is 3 percentage points higher. The performance oriented design decisions made for *M3D-Het* increase the power consumption. The increase in power consumption is due to using larger transistors in the top layer of some structures. Further, *M3D-Het* executes faster than *M3D-HetNaive*. Overall, the total energy consumption decreases slightly when compared to *M3D-HetNaive*. When compared to *Base*, the overall energy is reduced by 39% on average.

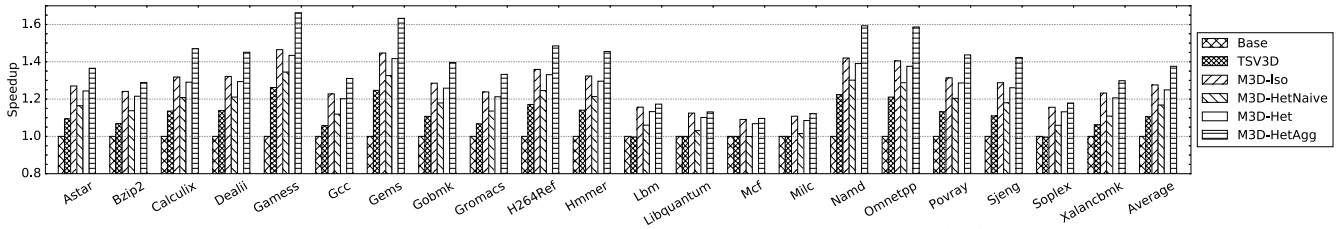

Figure 6: Speed-up of different M3D designs over *Base* (2D).

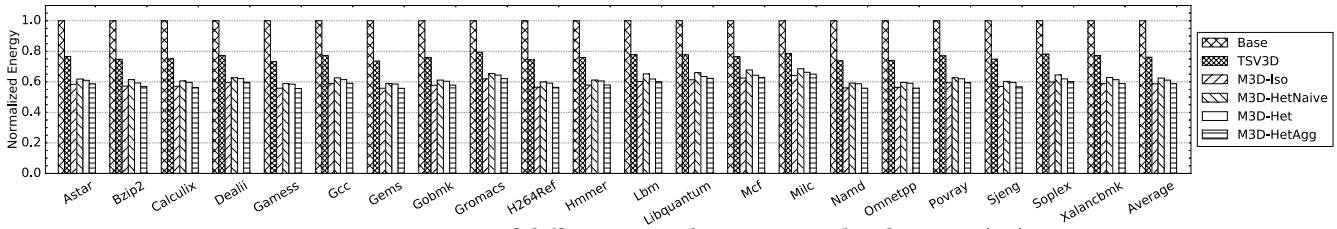

Figure 7: Energy of different M3D designs normalized to *Base* (2D).

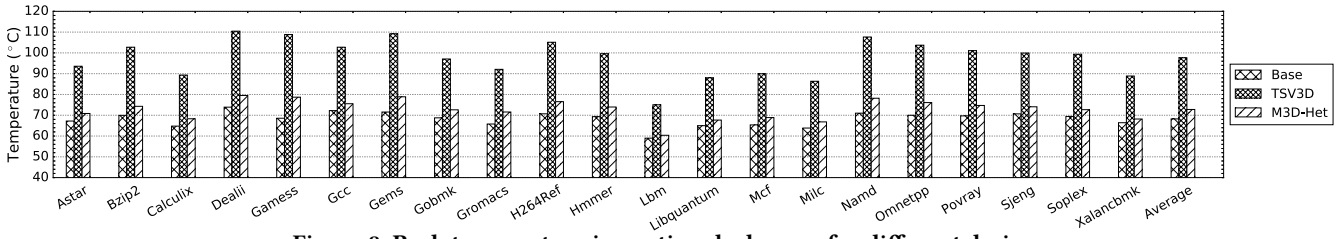

Figure 8: Peak temperature in centigrade degrees for different designs.

The more aggressive *M3D-HetAgg* executes faster, and lowers the energy consumption further, bringing the total energy savings to 41% on average.

In comparison, energy reductions in *TSV3D* are smaller at 24%. Similar to M3D designs, the energy savings in *TSV3D* are due to the reductions in SRAM array and clock tree power. However, as shown in Table 6, the magnitude of the array savings is smaller.

Hetero-Layers Using LP Process for Top Layer. As we discussed in Section 5, in an environment where it is feasible to manufacture M3D chips with iso-performance in both layers, we can combine a top layer in LP process and a bottom layer in HP process. Such a design, together with our techniques, would have the same performance as *M3D-Het*. We evaluate that this design reduces the energy further by on average 9 percentage points over *M3D-Het*.

7.1.3 Thermal Behavior. To study the thermal behavior of different designs, we measure the maximum temperature reached by a given design using the HotSpot tool as discussed in Section 6. We observe that the average power consumption of *Base* across all applications is 6.4W for a single core excluding the L2/L3 caches. The floorplan of our chip is based on AMD Ryzen [3], and we assume a 50% footprint reduction (only for calculating the peak temperature). This is a conservative assumption because it leads to higher temperatures.

Figure 8 shows the peak temperature in centigrade degrees reached within the core across all the applications for *Base*, *M3D-Het*, and *TSV3D*. The values for other M3D designs are very similar. The hottest point in the core varies across applications. For example, the hottest point is in the IQ for DealII, whereas it is in the FPU for Gems. From the figure, we see that the peak temperature in *M3D-Het* is, on average, only 5°C higher than in *Base*. The maximum increase across any application is 10°C, and is observed in the IQ for Gamess. These temperature increases are considerably smaller than the 30°C average temperature increase observed for *TSV3D*. In fact, *TSV3D* exceeds the maximum operating temperature of a transistor (T_{jmax} ≈100°C) for a few applications. The high temperatures in *TSV3D* are due to its thermal conduction bottlenecks [1, 41].

The temperature increases in *M3D-Het* are small due to two reasons. First, as we discussed in Section 2, the vertical thermal conduction across the layers is high. Second, in *M3D-Het*, we observe that some of the hot spots such as IQ, RAT, and RF have relatively large power reductions. For example, it can be shown that IQ consumes 34% less power in *M3D-Het*, which is higher than the 24% power reduction for the whole core. The reason for this is that port partitioning, as used in structures such as IQ, RAT, and RF, is relatively more effective at reducing energy than other forms of partitioning. Therefore, it can be shown that, even though the overall power density of *M3D-Het* increases by up to 52%, the increase in power density of hotter regions is smaller (i.e., 32% for IQ). Overall, M3D designs are more thermally efficient than *TSV3D*.

7.2 Multicore M3D Designs

In this section, we explore different multicore designs enabled by M3D running parallel applications, as we discussed in Section 6. The designs, shown in the multicore section of Table 11, are: *M3D-Het* with pairs of cores sharing their L2s and one NoC router stop, *M3D-Het-W* (which also uses wider-issue cores), *M3D-Het-2X* (which also

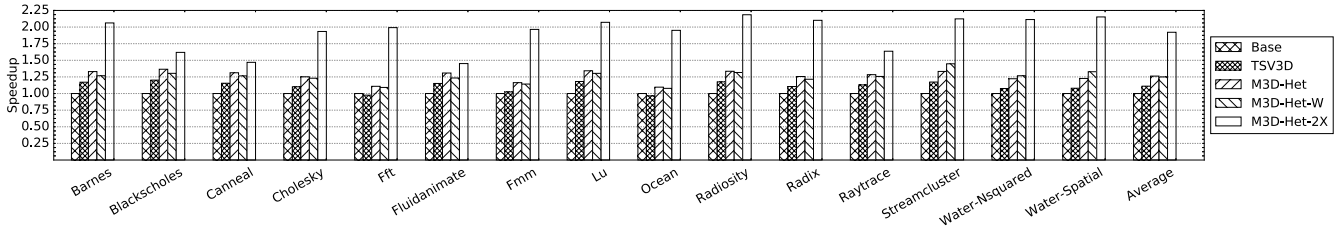

Figure 9: Speed-up of different multicore M3D designs over a four-core *Base* multicore (2D).

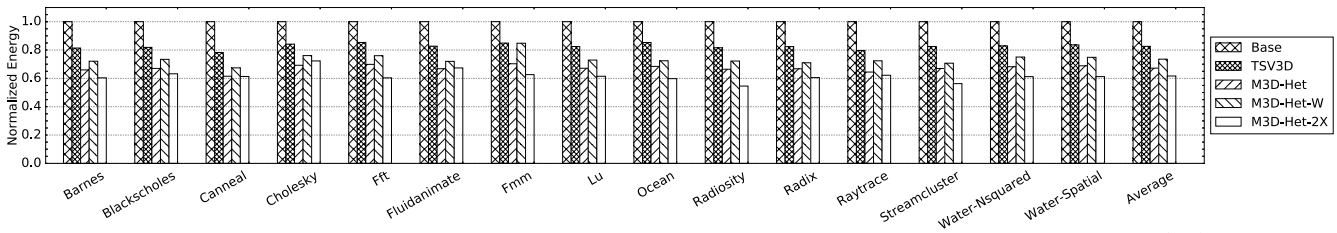

Figure 10: Energy of different multicore M3D designs normalized to a four-core *Base* multicore (2D).

operates more cores at iso-power conditions), and *TSV3D*. Figures 9 and 10 show the speed-up and energy consumption, respectively, of all the applications. The bars are normalized to a 4-core *Base* multicore.

7.2.1 Multicores with Four Cores. *M3D-Het*, *M3D-Het-W*, and *TSV3D* have four cores. On average, we see that *M3D-Het* provides a speed-up of 1.26 over *Base*, while reducing energy consumption by 33%. The performance gains include the benefits of shared L2s and NoC router on top of the gains seen in the single-core environment. With respect to the energy, we observe a slight reduction in dynamic energy due to the reduction in the network traffic, in addition to the energy saving factors discussed previously.

M3D-Het-W is an *M3D-Het* design with cores whose issue width is increased from 6 to 8, while operating at the same frequency as *Base* (Table 11). Its average speed-up and reduction in energy consumption are 1.25 and 26%, respectively. These numbers are worse than *M3D-Het*, which simply increases the frequency.

TSV3D is not competitive. Its average speed-up over *Base* is only 1.11, while it reduces the energy consumption by 17%.

7.2.2 Iso Power Environment. *M3D-Het-2X* is an *M3D-Het* design operating at the same frequency as *Base*, but at a lower voltage and thus with lower power. As a result, it has twice as many cores as *Base* with about the same power budget as *Base*. From Figure 9, we see that *M3D-Het-2X* is 92% faster the *Base*. This speed-up is due to both executing with more cores, and the factors discussed previously. At the same time, *M3D-Het-2X* consumes 39% lower energy than *Base*. Note that these results are for the hetero-layer M3D design, and with conservative slowdown assumptions (i.e., building on top of *M3D-Het* rather than *M3D-HetAgg*).

Overall, *M3D-Het-2X*, operating twice as many cores in a power budget that is, on average, only 13% higher than *Base*, provides substantial performance improvement while consuming lower energy.

8 OTHER RELATED WORK

In addition to the TSV-based intra-block 3D partitioning of the core that we discussed in Section 2.3 [41, 42], prior work has also considered partitioning the core at block-level granularity using TSVs. Specifically, Black et al. [11] study the benefits of such core partitioning, as well as the benefits of placing DRAM/SRAM on top of a core. They also note that a TSV3D core can have thermal challenges. In this paper, we compared an M3D core against a TSV3D core with intra-block partitioning, which has more benefits than block-level partitioning.

Emma et al. [16] limit themselves to core-level partitioning across the different layers, while sharing different resources such as caches or NoC. They focus on the impact of 3D partitioning from a thermal and yield perspective, and discuss the tradeoffs between power and performance in a 3D setting. Our analysis of M3D core design is at a much finer granularity of partitioning.

9 CONCLUSION

This paper showed how to partition a processor for M3D. We partition logic and storage structures into two layers, taking into account that the top layer has lower-performance transistors. In logic structures, we place the critical paths in the bottom layer. In storage structures, we asymmetrically partition them, assigning to the top layer fewer ports with larger access transistors, or a shorter bitcell subarray with larger bitcells. With conservative assumptions on M3D technology, an M3D core executed applications on average 25% faster than a 2D core while consuming 39% less energy. A more aggressive M3D design was on average 38% faster than a 2D core while consuming 41% lower energy. Moreover, under a similar power budget, an M3D multicore could use twice as many cores as a 2D multicore, executing applications on average 92% faster with 39% less energy. Finally, the M3D core was thermally efficient.

ACKNOWLEDGMENTS

This work was funded in part by NSF under grants CNS-1763658 and CCF-1649432. We sincerely thank Prof. Sung-Kyu Lim and his team from Georgia Institute of Technology, and Ashutosh Dhar from University of Illinois for their gracious help with 3D tools and modeling. We greatly thank the anonymous reviewers for their extensive feedback.

REFERENCES

[1] A. Agrawal, J. Torrellas, and S. Idgunji. 2017. Xylem: Enhancing Vertical Thermal Conduction in 3D Processor-Memory Stacks. In *Proceedings of the 50th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO-50)*. https://doi.org/10.1145/3123939.3124547

[2] F. Andrieu, P. Batude, L. Brunet, C. Fenouillet-Beranger, D. Lattard, S. Thuries, O. Billoint, R. Fournel, and M. Vinet. 2018. A review on opportunities brought by 3D-monolithic integration for CMOS device and digital circuit. In *2018 International Conference on IC Design Technology (ICICDT)*. https://doi.org/10.1109/ICICDT.2018.8399776

[3] AMD Ryzen Micro Architecture. 2017. https://arstechnica.com/gadgets/2017/03/amds-moment-of-zen-finally-an-architecture-that-can-compete/. [Online].

[4] Rajeev Balasubramonian, Andrew B. Kahng, Naveen Muralimanohar, Ali Shafiee, and Vaishnav Srinivas. 2017. CACTI 7: New Tools for Interconnect Exploration in Innovative Off-Chip Memories. *ACM Transactions on Architecture Code and Optimization* (June 2017). https://doi.org/10.1145/3085572

[5] P. Batude, T. Ernst, J. Arcamone, G. Arndt, P. Coudrain, and P. E. Gaillardon. 2012. 3-D Sequential Integration: A Key Enabling Technology for Heterogeneous Co-Integration of New Function With CMOS. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems* (Dec 2012).

[6] P. Batude, B. Sklenard, C. Fenouillet-Beranger, B. Previtali, C. Tabone, O. Rozeau, O. Billoint, O. Turkyilmaz, H. Sarhan, S. Thuries, G. Cibrario, L. Brunet, F. Deprat, J. E. Michallet, F. Clermidy, and M. Vinet. 2014. 3D sequential integration opportunities and technology optimization. In *IEEE International Interconnect Technology Conference*. https://doi.org/10.1109/IITC.2014.6831837

[7] Perrine Batude, Maud Vinet, Arnaud Pouydebasque, Laurent Clavelier, Cyrille LeRoyer, Claude Tabone, Bernard Previtali, Loic Sanchez, Laurence Baud, Antonio Roman, Veronique Carron, Fabrice Nemouchi, Stephane Pocas, Corine Comboroure, Vincent Mazzocchi, Helen Grampeix, Francois Aussenac, and Simon Deleonibus. 2008. Enabling 3D Monolithic Integration. *ECS Transactions* (2008). https://doi.org/10.1149/1.2982853

[8] P. Batude, M. Vinet, C. Xu, B. Previtali, C. Tabone, C. Le Royer, L. Sanchez, L. Baud, L. Brunet, A. Toffoli, F. Allain, D. Lafond, F. Aussenac, O. Thomas, T. Poiroux, and O. Faynot. 2011. Demonstration of low temperature 3D sequential FDSOI integration down to 50 nm gate length. In *2011 Symposium on VLSI Technology - Digest of Technical Papers*.

[9] L. Baugh and C. Zilles. 2006. Decomposing the load-store queue by function for power reduction and scalability. *IBM Journal of Research and Development* (March 2006). https://doi.org/10.1147/rd.502.0287

[10] O. Billoint, H. Sarhan, I. Rayane, M. Vinet, P. Batude, C. Fenouillet-Beranger, O. Rozeau, G. Cibrario, F. Deprat, A. Fustier, J. E. Michallet, O. Faynot, O. Turkyilmaz, J. F. Christmann, S. Thuries, and F. Clermidy. 2015. A comprehensive study of Monolithic 3D cell on cell design using commercial 2D tool. In *2015 Design, Automation Test in Europe Conference Exhibition (DATE)*. https://doi.org/10.7873/DATE.2015.1110

[11] B. Black, M. Annavaram, N. Brekelbaum, J. DeVale, L. Jiang, G. H. Loh, D. McCaule, P. Morrow, D. W. Nelson, D. Pantuso, P. Reed, J. Rupley, S. Shankar, J. Shen, and C. Webb. 2006. Die Stacking (3D) Microarchitecture. In *2006 39th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO'06)*. https://doi.org/10.1109/MICRO.2006.18

[12] Shashikanth Bobba, Ashutosh Chakraborty, Olivier Thomas, Perrine Batude, and Giovanni de Micheli. 2013. Cell Transformations and Physical Design Techniques for 3D Monolithic Integrated Circuits. *Journal on Emerging Technologies in Computing Systems (JETC)* (2013). https://doi.org/10.1145/2491675

[13] M. Brocard, R. Boumchedda, J. P. Noel, K. C. Akyel, B. Giraud, E. Beigne, D. Turgis, S. Thuries, G. Berhault, and O. Billoint. 2016. High density SRAM bitcell architecture in 3D sequential CoolCube 14nm technology. In *2016 IEEE SOI-3D-Subthreshold Microelectronics Technology Unified Conference (S3S)*. https://doi.org/10.1109/S3S.2016.7804376

[14] L. Brunet, P. Batude, C. Fenouillet-Beranger, P. Besombes, L. Hortemel, F. Ponthenier, B. Previtali, C. Tabone, A. Royer, C. Agraffeil, C. Euvrard-Colnat, A. Seignard, C. Morales, F. Fournel, L. Benaissa, T. Signamarcheix, P. Besson, M. Jourdan, R. Kachtouli, V. Benevent, J. Hartmann, C. Comboroure, N. Allouti, N. Posseme, C. Vizioz, C. Arvet, S. Barnola, S. Kerdiles, L. Baud, L. Pasini, C. V. Lu, F. Deprat, A. Toffoli, G. Romano, C. Guedj, V. Delaye, F. Boeuf, O. Faynot, and M. Vinet. 2016. First demonstration of a CMOS over CMOS 3D VLSI CoolCube integration on 300mm wafers. In *2016 IEEE Symposium on VLSI Technology*. https://doi.org/10.1109/VLSIT.2016.7573428

[15] G. Van der Plas, P. Limaye, I. Loi, A. Mercha, H. Oprins, C. Torregiani, S. Thijs, D. Linten, M. Stucchi, G. Katti, D. Velenis, V. Cherman, B. Vandevelde, V. Simons, I. De Wolf, R. Labie, D. Perry, S. Bronckers, N. Minas, M. Cupac, W. Ruythooren, J. Van Olmen, A. Phommahaxay, M. de ten Broeck, A. Opdebeeck, M. Rakowski, B. De Wachter, M. Dehan, M. Nelis, R. Agarwal, A. Pullini, F. Angiolini, L. Benini, W. Dehaene, Y. Travaly, E. Beyne, and P. Marchal. 2011. Design Issues and Considerations for Low-Cost 3-D TSV IC Technology. *IEEE Journal of Solid-State Circuits* (Jan 2011).

[16] P. Emma, A. Buyuktosunoglu, M. Healy, K. Kailas, V. Puente, R. Yu, A. Hartstein, P. Bose, and J. Moreno. 2014. 3D stacking of high-performance processors. In *2014 IEEE 20th International Symposium on High Performance Computer Architecture (HPCA)*. https://doi.org/10.1109/HPCA.2014.6835959

[17] A. Gonzalez, F. Latorre, and G. Magklis. 2010. Processor Microarchitecture: An Implementation Perspective. *Synthesis Lectures on Computer Architecture* (2010). https://doi.org/10.2200/S00309ED1V01Y201011CAC012

[18] B. Gopireddy, C. Song, J. Torrellas, N. S. Kim, A. Agrawal, and A. Mishra. 2016. ScalCore: Designing a Core for Voltage Scalability. In *2016 IEEE International Symposium on High Performance Computer Architecture (HPCA)*. https://doi.org/10.1109/HPCA.2016.7446104

[19] Wei Huang, S. Ghosh, S. Velusamy, K. Sankaranarayanan, K. Skadron, and M. R. Stan. 2006. HotSpot: A compact thermal modeling methodology for early-stage VLSI design. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* (May 2006). https://doi.org/10.1109/TVLSI.2006.876103

[20] S. Van Huylenbroeck, M. Stucchi, Y. Li, J. Slabbekoorn, N. Tutunjyan, S. Sardo, N. Jourdan, L. Bogaerts, F. Beirnaert, G. Beyer, and E. Beyne. 2016. Small Pitch, High Aspect Ratio Via-Last TSV Module. In *2016 IEEE 66th Electronic Components and Technology Conference (ECTC)*. https://doi.org/10.1109/ECTC.2016.155

[21] International Roadmap for Devices and Systems. 2017. IRDS. (2017). https://irds.ieee.org/roadmap-2017

[22] International Technology Roadmap for Semiconductors. 2015. ITRS 2.0. (2015). http://www.itrs2.net.

[23] S. Jain, S. Khare, S. Yada, V. Ambili, P. Salihundam, S. Ramani, S. Muthukumar, M. Srinivasan, A. Kumar, S. K. Gb, R. Ramanarayanan, V. Erraguntla, J. Howard, S. Vangal, S. Dighe, G. Ruhl, P. Aseron, H. Wilson, N. Borkar, V. De, and S. Borkar. 2012. A 280mV-to-1.2V wide-operating-range IA-32 processor in 32nm CMOS. In *2012 IEEE International Solid-State Circuits Conference*. https://doi.org/10.1109/ISSCC.2012.6176932

[24] C. H. Jan, F. Al-amoody, H. Y. Chang, T. Chang, Y. W. Chen, N. Dias, W. Hafez, D. Ingerly, M. Jang, E. Karl, S. K. Y. Shi, K. Komeyli, H. Kilambi, A. Kumar, K. Byon, C. G. Lee, J. Lee, T. Leo, P. C. Liu, N. Nidhi, R. Olac-vaw, C. Petersburg, K. Phoa, C. Prasad, C. Quincy, R. Ramaswamy, T. Rana, L. Rockford, A. Subramaniam, C. Tsai, P. Vandervoorn, L. Yang, A. Zainuddin, and P. Bai. 2015. A 14 nm SoC platform technology featuring 2nd generation Tri-Gate transistors, 70 nm gate pitch, 52 nm metal pitch, and 0.0499 um2 SRAM cells, optimized for low power, high performance and high density SoC products. In *2015 Symposium on VLSI Circuits (VLSI Circuits)*. https://doi.org/10.1109/VLSIC.2015.7231380

[25] C.-H. Jan, U. Bhattacharya, R. Brain, S.-J. Choi, G. Curello, G. Gupta, W. Hafez, M. Jang, M. Kang, K. Komeyli, T. Leo, N. Nidhi, L. Pan, J. Park, K. Phoa, A. Rahman, C. Staus, H. Tashiro, C. Tsai, P. Vandervoorn, L. Yang, J.-Y. Yeh, and P. Bai. 2012. A 22nm SoC platform technology featuring 3-D tri-gate and high-k/metal gate, optimized for ultra low power, high performance and high density SoC applications. In *2012 International Electron Devices Meeting*. https://doi.org/10.1109/IEDM.2012.6478969

[26] D. H. Kim, K. Athikulwongse, and S. K. Lim. 2009. A study of Through-Silicon-Via impact on the 3D stacked IC layout. In *2009 IEEE/ACM International Conference on Computer-Aided Design - Digest of Technical Papers*.

[27] J. Kong, Y. Gong, and S. W. Chung. 2017. Architecting large-scale SRAM arrays with monolithic 3D integration. In *2017 IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED)*. https://doi.org/10.1109/ISLPED.2017.8009157

[28] B. W. Ku, T. Song, A. Nieuwoudt, and S. K. Lim. 2017. Transistor-level monolithic 3D standard cell layout optimization for full-chip static power integrity. In *2017 IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED)*. https://doi.org/10.1109/ISLPED.2017.8009189

[29] Y. J. Lee, D. Limbrick, and S. K. Lim. 2013. Power benefit study for ultra-high density transistor-level monolithic 3D ICs. In *2013 50th ACM/EDAC/IEEE Design Automation Conference (DAC)*.

[30] S. Li, J. H. Ahn, R. D. Strong, J. B. Brockman, D. M. Tullsen, and N. P. Jouppi. 2009. McPAT: An integrated power, area, and timing modeling framework for multicore and manycore architectures. In *42nd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*.

[31] C. Liu and S. K. Lim. 2012. A design tradeoff study with monolithic 3D integration. In *Thirteenth International Symposium on Quality Electronic Design (ISQED)*. https://doi.org/10.1109/ISQED.2012.6187545

[32] C. Liu and S. K. Lim. 2012. Ultra-high density 3D SRAM cell designs for monolithic 3D integration. In *2012 IEEE International Interconnect Technology Conference*. https://doi.org/10.1109/IITC.2012.6251581

[33] J. Meng, K. Kawakami, and A. K. Coskun. 2012. Optimizing energy efficiency of 3-D multicore systems with stacked DRAM under power and thermal constraints. In *Design Automation Conference (DAC) 2012*.

[34] D. E. Nikonov and I. A. Young. 2015. Benchmarking of Beyond-CMOS Exploratory Devices for Logic Integrated Circuits. *IEEE Journal on Exploratory Solid-State Computational Devices and Circuits* (Dec 2015). https://doi.org/10.1109/JXCDC.2015.2418033

[35] C. Ortolland, T. Noda, T. Chiarella, S. Kubicek, C. Kerner, W. Vandervorst, A. Opdebeeck, C. Vrancken, N. Horiguchi, M. De Potter, M. Aoulaiche, E. Rosseel,

S. B. Felch, P. Absil, R. Schreutelkamp, S. Biesemans, and T. Hoffmann. 2008. Laser-annealed junctions with advanced CMOS gate stacks for 32nm node: Perspectives on device performance and manufacturability. In *2008 Symposium on VLSI Technology*. https://doi.org/10.1109/VLSIT.2008.4588612

[36] Subbarao Palacharla, Norman P. Jouppi, and James E. Smith. 1996. Quantifying the Complexity of Superscalar Processors. ftp://ftp.cs.wisc.edu/sohi/trs/complexity.1328.pdf

[37] Subbarao Palacharla, Norman P. Jouppi, and J. E. Smith. 1997. Complexity-effective Superscalar Processors. In *Proceedings of the 24th Annual International Symposium on Computer Architecture*. https://doi.org/10.1145/264107.264201

[38] S. Panth, K. Samadi, Y. Du, and S. K. Lim. 2013. High-density integration of functional modules using monolithic 3D-IC technology. In *2013 18th Asia and South Pacific Design Automation Conference (ASP-DAC)*. https://doi.org/10.1109/ASPDAC.2013.6509679

[39] S. Panth, K. Samadi, Y. Du, and S. K. Lim. 2014. Design and CAD methodologies for low power gate-level monolithic 3D ICs. In *2014 IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED)*. https://doi.org/10.1145/2627369.2627642

[40] S. Panth, K. Samadi, Y. Du, and S. K. Lim. 2014. Power-performance study of block-level monolithic 3D-ICs considering inter-tier performance variations. In *2014 51st ACM/EDAC/IEEE Design Automation Conference (DAC)*. https://doi.org/10.1145/2593069.2593188

[41] K. Puttaswamy and G. H. Loh. 2007. Thermal Herding: Microarchitecture Techniques for Controlling Hotspots in High-Performance 3D-Integrated Processors. In *2007 IEEE 13th International Symposium on High Performance Computer Architecture*. https://doi.org/10.1109/HPCA.2007.346197

[42] K. Puttaswamy and G. H. Loh. 2009. 3D-Integrated SRAM Components for High-Performance Microprocessors. *IEEE Trans. Comput.* (Oct 2009). https://doi.org/10.1109/TC.2009.92

[43] B. Rajendran, R. S. Shenoy, D. J. Witte, N. S. Chokshi, R. L. DeLeon, G. S. Tompa, and R. F. W. Pease. 2007. Low Thermal Budget Processing for Sequential 3-D IC Fabrication. *IEEE Transactions on Electron Devices* (April 2007). https://doi.org/10.1109/TED.2007.891300

[44] S. K. Samal, D. Nayak, M. Ichihashi, S. Banna, and S. K. Lim. 2016. Monolithic 3D IC vs. TSV-based 3D IC in 14nm FinFET technology. In *2016 IEEE SOI-3D-Subthreshold Microelectronics Technology Unified Conference (S3S)*. https://doi.org/10.1109/S3S.2016.7804405

[45] J. Shi, D. Nayak, S. Banna, R. Fox, S. Samavedam, S. Samal, and S. K. Lim. 2016. A 14nm FinFET transistor-level 3D partitioning design to enable high-performance and low-cost monolithic 3D IC. In *2016 IEEE International Electron Devices Meeting (IEDM)*. https://doi.org/10.1109/IEDM.2016.7838032

[46] M. M. Shulaker, T. F. Wu, A. Pal, L. Zhao, Y. Nishi, K. Saraswat, H. S. P. Wong, and S. Mitra. 2014. Monolithic 3D integration of logic and memory: Carbon nanotube FETs, resistive RAM, and silicon FETs. In *2014 IEEE International Electron Devices Meeting*. https://doi.org/10.1109/IEDM.2014.7047120

[47] S. Srinivasa, X. Li, M. Chang, J. Sampson, S. K. Gupta, and V. Narayanan. 2018. Compact 3-D-SRAM Memory With Concurrent Row and Column Data Access Capability Using Sequential Monolithic 3-D Integration. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* (April 2018). https://doi.org/10.1109/TVLSI.2017.2787562

[48] Srivatsa Srinivasa, Akshay Krishna Ramanathan, Xueqing Li, Wei-Hao Chen, Fu-Kuo Hsueh, Chih-Chao Yang, Chang-Hong Shen, Jia-Min Shieh, Sumeet Gupta, Meng-Fan Marvin Chang, Swaroop Ghosh, Jack Sampson, and Vijaykrishnan Narayanan. 2018. A Monolithic-3D SRAM Design with Enhanced Robustness and In-Memory Computation Support. In *Proceedings of the International Symposium on Low Power Electronics and Design (ISLPED '18)*. https://doi.org/10.1145/3218603.3218645

[49] R. Ubal, B. Jang, P. Mistry, D. Schaa, and D. Kaeli. 2012. Multi2Sim: A simulation framework for CPU-GPU computing. In *21st International Conference on Parallel Architectures and Compilation Techniques (PACT)*.