Testing Circuit-Partitioned 3D IC Designs

Dean L. Lewis Hsien-Hsin S. Lee

School of Electrical and Computer Engineering
Georgia Institute of Technology
Atlanta, GA 30332
{dean, leehs}@ece.gatech.edu

ABSTRACT

3D integration is an emerging technology that allows for the vertical stacking of multiple silicon die. These stacked die are tightly integrated with through-silicon vias and promise significant power and area reductions by replacing long global wires with short vertical connections. This technology necessitates that neighboring logical blocks exist on different layers in the stack. However, such functional partitions disable intra-chip communication pre-bond and thus disrupt traditional test techniques.

Previous work has described a general test architecture that enables pre-bond testability of an architecturally partitioned 3D processor and provided mechanisms for basic layer functionality. This work proposes new test methods for designs partitioned at the circuits level, in which the gates and transistors of individual circuits could be split across multiple die layers. We investigated a bit-partitioned adder unit and a port-split register file, which represents the most difficult circuit-partitioned design to test pre-bond but which is used widely in many circuits. Two layouts of each circuit, planar and 3D, are produced. Our experiments verify the performance and power results and examine the test coverage achieved.

Keywords

DFT, Die Stacking, 3D ICs, Memory Test, BIST

1. INTRODUCTION

As the IC industry continues the push to smaller and smaller device geometries, the cost of each new process generation is steadily on the rise while the returns continue to diminish. In an effort to keep up with Moore's Law in spite of these difficulties, manufacturers are increasingly turning to new fabrication technologies. 3D integration is one such technology which allows for the integration of multiple silicon die into a single chip stack. Vertical integration is completely orthogonal to device scaling, making it an excellent complementary technology to help keep Moore's Law on track for at least another decade.

Previous works on 3D design have studied a number of different partitioning schemes [3, 4, 9, 14, 10]. These designs range from simply stacking SRAM die on top of a processor die (to form a massive last-level cache) to splitting a single microarchitectural block or even circuit (such as an adder) across multiple die. Some of the latter advanced designs promise increased performance while simultaneously reducing both power consumption and area. However, one major problem remains largely unaddressed: how do we test these individual die before bonding them together to form the complete chip? Note that, without this pre-bond test, a defect in a single die could ruin the entire stack, which reduces manufacturing yield exponentially as the number of die increases. This work proposes and evaluates test strategies for two of the most ambitious 3D designs, a bit-split Kogge-Stone adder and a port-split register file, extending the previous work by Lewis and Lee on a general pre-bond test strategy [8].

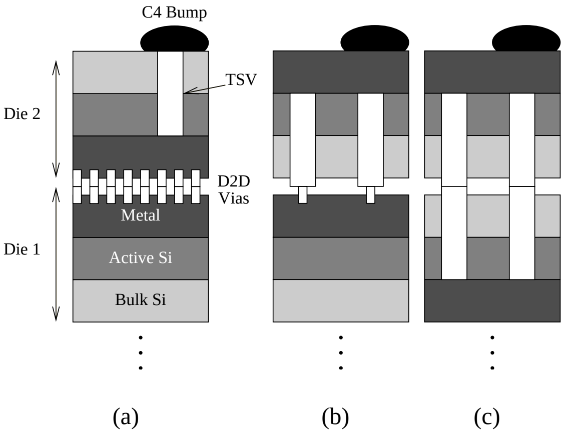


Figure 1: Three die stacks, each comprised of two layers using three possible bond styles: (a) face-to-face, (b) face-to-back, and (c) back-to-back.

The rest of this paper is organized as follows. Section 2 introduces 3D technology, explores the contributions of previous work, and the problem addressed in this work; Section 3 will present the 3D designs we have considered and our extensions to these designs to enable pre-bond testability; Section 4 presents our experimental setup and results; Section 5 discusses related work. Section 6 concludes the paper with a summary and discussion of results.

2. 3D INTEGRATION AND TEST

3D integration (die stacking) is an emerging technology in which multiple silicon die are stacked and tightly integrated with short, dense die-to-die vias. Designing in the third dimension has many advantages. First, it allows for die manufactured in incompatible processes to be tightly integrated—for example, logic and DRAM [3]. Second, it can increase routability [13]. Third, the high density of die-to-die vias can provide a plethora of memory bandwidth, which has been constrained by the pin count on the package. Last, and possibly most important, it can substantially reduce wire length, which in planar die both degrade performance and increase power consumption [14].

2.1 3D Technology

Figure 1 illustrates the general concept of 3D integration. Two die, previously manufactured in any VLSI process, are bonded together with short, high-density die-to-die (d2d) vias. These d2d vias come in two flavors, faceside and backside. Faceside vias, manufactured on top of the metal interconnect layers, can be produced on a pitch of a few hundred nanometers [15]. Backside vias, also called *through silicon vias* (TSVs), are manufactured through the bulk silicon on a pitch of microns. To keep these TSVs small, the bulk silicon must be thinned, usually with a CMP process, to a few tens of microns. D2d vias on different die are then fused together to bond the die together [12]. A face-to-face bond, Figure 1(a), is best, providing the shortest, highest density interface. However, stack heights greater than two layers require the use of face-to-back or back-to-back bonds, shown in Figure 1(b) and Figure 1(c) respectively. Once the stack is complete, normal C4 solder bumps can be placed either on TSVs (Figure 1(a)) or on top of the metal layers as in a traditional planar design.

2.2 3D Partitioning

Generally speaking, there are three distinct granularities of 3D partitioning schemes. The coarsest granularity is the *technology partitioning*. Disparate technologies, like high-speed CMOS and high-density DRAM, are manufactured in separate, optimized processes and then tightly integrated with 3D technology. This integration allows for high-speed, high-bandwidth interconnections between technologies that simply are not possible with planar manufacturing.

The next finer level of partitioning is the architectural level. Here, die are manufactured in the same process. The goal is to partition the microarchitectural blocks across the different layers such that the total wire length is minimized. For example, adders could be stacked, allowing the bypass bus to make short vertical connections instead of long horizontal ones. Architectural partitioning generally makes much better use of the available d2d vias than technology partitioning.

The finest partitioning granularity is the circuit level. Here, individual blocks or even individual circuits are partitioned across multiple layers. A large range of possibilities exist at this granularity. At one end of the spectrum is sub-block partitioning where a block is split along logical boundaries. For example, a cache bank could be folded in half, significantly reducing the load on the word- or bit-lines [9, 14]. At the other end, individual circuits are partitioned. For example, the ports in a multi-ported register file can be partitioned across layers, greatly reducing the area and thus wirelengths of the register file [14]. Such designs make best use of the available d2d vias and thus promise the best improvements in power and performance.

2.3 3D Test

3D integration suffers from the same problem as multi-chip modules (MCMs), IC boards, and other integration schemes: one bad component can kill the system. As more components are integrated, the yield of the final product falls off exponentially. The solution is to test components before integration, finding so called “known good die” (KGD) parts. We propose this same approach to pre-bond test for 3D integration.

At the technology granularity, there is little challenge. Each layer is a complete, functional design that can be tested in a normal planar method (as is done for MCMs). The only challenge lies in the coexistance of probe pads for test and d2d vias for 3D integration. But since these designs consume relatively few vias, there is room to spare, so this is really just an engineering problem to be tackled on a per-design basis.

At the architectural granularity, things get trickier. Buses that connect neighboring blocks in a planar design will likely be non-functional in a pre-bond test situation. Worse, global signals—clock, power, reset, etc.—may not be functional pre-bond. These challenges were partly addressed in previous work by Lewis and Lee [8]. They showed that application of the scan island methodology, first implemented in the Alpha 21364 processor, could sufficiently test blocks pre-bond. Additionally, design options for ensuring power, clock, and reset distribution pre-bond were presented. However, this work was limited in scope to the architectural granularity and did not consider finer partitions.

At the circuit granularity, pre-bond test becomes quite a challenge. Individual transistors from a single circuit may be partitioned across the stack. This leads to a bit of a paradox in that the circuits are functionally broken pre-bond, yet we want to test them for correct functionality. Additionally, the large number of d2d vias in some designs makes traditional scan-based test impractical. To address these issues, we consider two designs: a bit-split Kogge-Stone adder and a port-split register file. The adder represents a sub-block partition where the number of vias is small enough to allow scan-based test. The register files represents the opposite end of the spectrum, and a new test methodology is required. Taken together, these examples demonstrate that the range of circuit-partitioned designs can be practically tested pre-bond.

3. 3D DESIGN AND TEST

Previous work in 3D design has examined different partitioning schemes for key functional units in high-performance microprocessors. These units include caches, instruction schedulers, arithmetic units, and register files [14]. Some of these—the cache designs in particular—involve what is best described as sub-block partitioning. These designs are easily testable using the scan island test strategy in [8]. Others, most notably the port-split register file design, are partitioned at a very fine granularity and seem completely untestable by known techniques.

To cover this range of partitioning options, two designs are selected as representative cases. These are the bit-partitioned Kogge-Stone adder and the port-partitioned register file. The Kogge-Stone adder represents the easiest of the circuit-partitioned cases, using only a few internal vias and mostly resembling an architecture-partitioned design (i.e. most functionality is still intact pre-bond). The port-split register file, on the other hand, makes extensive use of internal vias and heavily divides functionality across layers, representing a unique and difficult pre-bond test challenge. These two functional units, an adder and SRAM memory array, also represent the most commonly seen components inside a microprocessor. The particulars of each 3D design and the necessary test strategy are discussed below.

3.1 Kogge-Stone Adder

The planar and 3D designs of an eight-bit adder are shown in Figure 2. A Kogge-Stone adder makes heavy use of prefix units to minimize the fanout of each unit and increase addition speed. As shown, prefix values are shifted left after each stage by an exponentially increasing distance to produce the carry values. As the bit count increase to 32, 64, and 128 bits, the wiring costs explode. To alleviate this problem, the 3D design proposes a modulus partitioning of the original operand bits. Figure 2(c) shows a planar representation of a modulus two (i.e. odd and even) partitioning; Figure 2(d) shows the same partition when stacked. In the first level of logic, the even bits and odd bits are exchanged across vias. In all other logic levels, the even and odd halves of the adder do not communicate. The logic circuit for a single bit is shown in Figure 2(b), including the location of the TSV and scan register (a control latch on the TSV output; an observation latch on the TSV input is not required because the signal is observable elsewhere). While the planar implementation had to wire these non-communicative blocks side-by-side, the 3D partitioning enables the independent wiring to get out of each others’ way, greatly reducing wiring area. Note that the wiring complexity of the

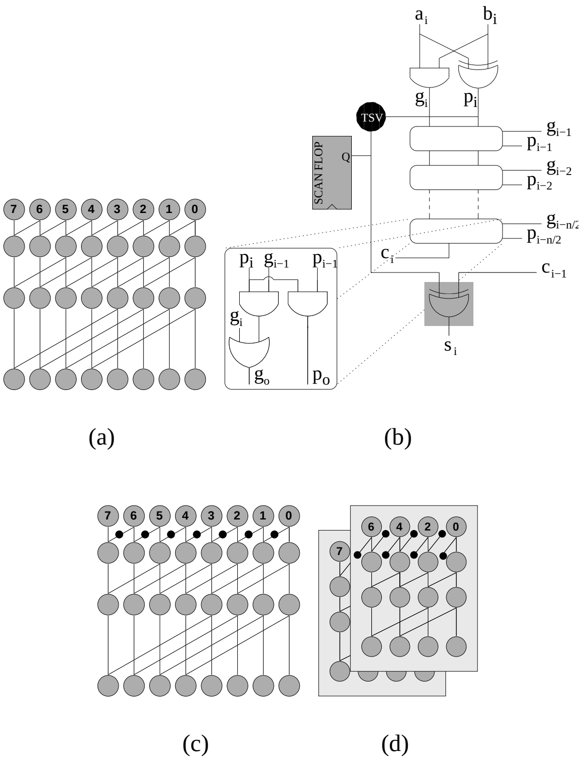


Figure 2: An eight-bit Kogge-Stone adder. (a) shows the planar implementation with its massive wiring area. (b) shows a single column of the adder in detail; shaded blocks are on the opposite layer from non-shaded blocks. (c) shows the placement of the vias in the 3D design. (d) shows the true 3D design with the significant wiring reduction.

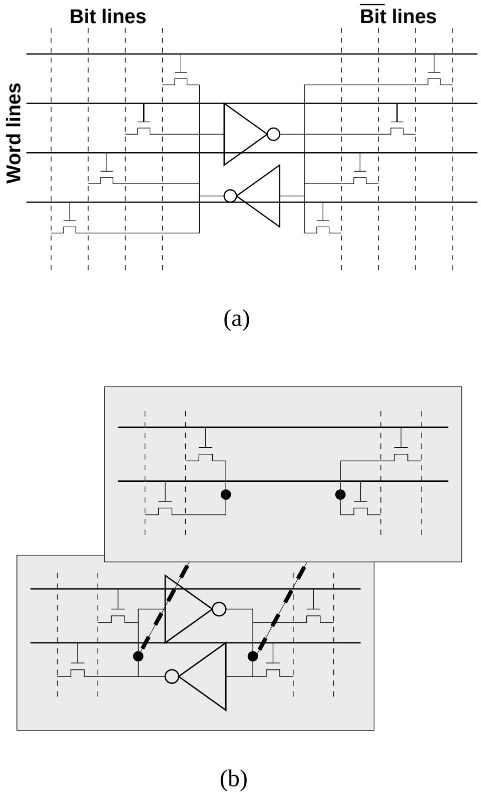


Figure 3: A four-port SRAM cell. This cell is laid out in an array to form a four-port register file. (a) shows the planar implementation with its massive wiring area. (b) shows the equivalent 3D design. Note that the lengths of the bitli.e. wordlines, and internal nets have all be significantly reduced.

3D implementation resembles that of a planar four-bit adder, a significant improvement over the eight-bit planar adder. So modulus two bit-partitioning has the effect of replacing the last, most-complex tract of wiring with a via tract (with wiring complexity equal to the first, simplest wiring tract), significantly increasing addition speed while simultaneously cutting power consumption.

Though only a modulus two partitioning is shown, higher moduli can be used in stacks of more die. For example, with four layers, each group of four bits could be partitioned across the stack. This would replace the two last, most complex wiring tracts with two via tracts of complexity equal only to the first two wiring tracts. Thus the design is very extensible to higher layer counts.

3.2 Testing the 3D Kogge-Stone Adder

The 3D Kogge-Stone adder has vias only in the first level of logic.[1] Thus, these vias are easily accessible from outside the adder as control points. To test the adder pre-bond, we simply add scan registers at the edge to provide test values on these nets. This enables full structural test of each half of the adder pre-bond.

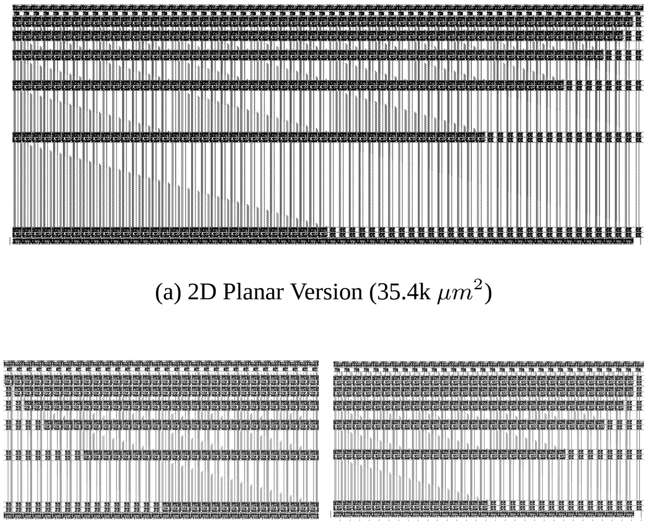
Because test cost (i.e. number of applied patterns) in general grows superlinearly with the complexity of the circuit under test, 3D designs naturally reduce total test time. That is, the number of patterns required to test each layer independently pre-bond and then test the connecting vias post-bond is less than the number of patterns required to test the planar implementation. To be fair, the planar design could be augmented to artificially divide it into independently-testable circuits similar to the 3D division. However, this would be more costly than the 3D split because it would require insertion of multiplexors into the adder's critical path to disable functional data during test. Since there is no functional data in the 3D adder pre-bond, this extra delay can be avoided, reducing the impact of test on the normal operation of the chip. Of course, the test data must be gated post-bond, but this gating would be off the critical path and thus less of a concern.

3.3 Port-Split Register File

Current high-performance microprocessors require simultaneous access to many operands from the register files to maintain high instruction throughput. Typically, the requirement is two read ports and one write port per parallel instruction plus a few extra for functions such as reads for data forwarding in the load-store queue that manage memory accesses. Modern superscalar processor designs execute between two and six instructions in parallel, which would require a minimum of six ports up to twenty or more ports.

Figure 3(a) shows the planar implementation of a single bit of a four-port register file. Note how the size of each bit grows quadratically with the port count, as each port requires dedicate bit- and wordlines. For a high-end, twenty-ported register file, the capacitances on the internal nets is massive, which is not desirable as the register file is critical in determining the operating frequency. To overcome this quadratic growth, an aggressive port-partitioning design was proposed in which some of the ports (half the ports, in the case of the two-layer design shown in Figure 3(b)) are placed on other layers.

[1] Vias would be required in more logic layers for partitions across more die; e.g. vias in the first two levels for a four-die partition.



(a) 2D Planar Version (35.4k μm^2)

(b) 3D Top (11.7k μm^2) (c) 3D Bottom (11.8k μm^2)

Figure 4: Layout for a 64-bit Kogge-Stone Adder.

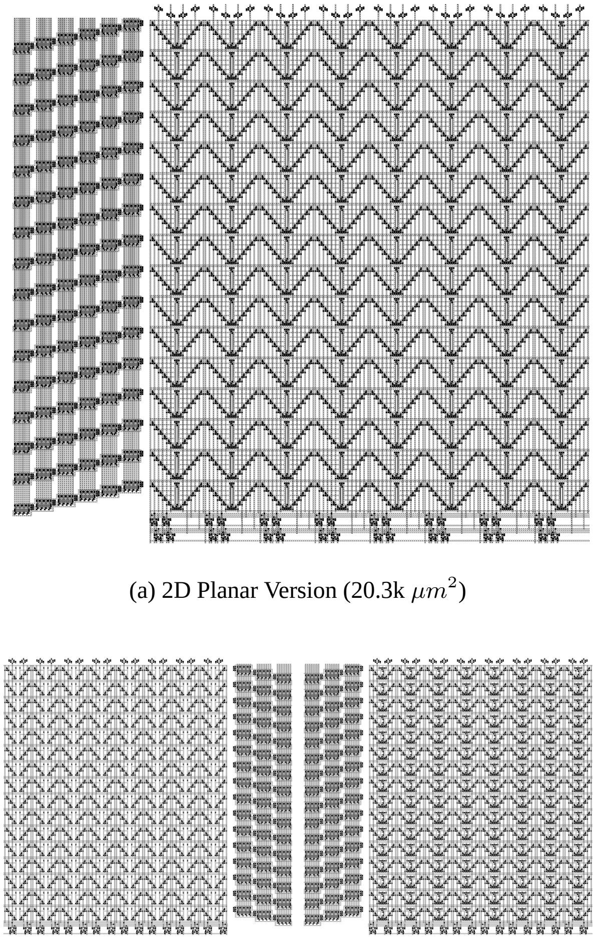
All these layers share a single cross-coupled inverter pair, with the ports on other layers connected back through vias. In the two-layer design, this reduces the size of the internal nets by a factor of four. With two layers, this adds up to half that size of the planar design. But not only are the internal nets significantly reduced, but all the bitlines and wordlines are also cut in half, effectively reducing the wiring load of the entire register file by half. This leads to significant, simultaneous performance improvement and power reduction.

3.4 Testing the 3D Register File

While the benefits of port-splitting are impressive, such a design poses serious pre-bond test challenges. Most notably, before the die are bonded, only one layer has access to the actual storage cell. The other layers have ports to nothing; they are functionally broken. This prevents the application of traditional memory test techniques such as Walking Ones [2] to any of these layers. To test these layers, a new approach is required.

Obviously, the layer with the memory cell can be tested using a classic algorithm. For the other layers, even though the memory cell is missing and the circuits cannot be tested as a memory unit, there is still sufficient functionality left in the circuit to test it. To enable test, we split the ports in such a way as to ensure that there is at least one write port and at least one read port on each layer. If the partitioning of a particular design has only read (or only write) ports on a given layer, one port could be converted to a combination read/write port to enable pre-bond test, a minimal overhead. It is now possible to stream test data through the ports to ensure they are functioning properly. This strategy tests each write port serially. A test vector is applied to the write port. Then the address of the write port and each read port is stepped through sequentially (Figure 6). This has the effect of the write port placing a value on the internal nodes and the read ports immediately reading it. Thus, we can verify the proper functioning of the ports by observing the initial test vectors on the read ports.

Notice that this strategy tests all memory components: address decoder, write hardware, bitlines and wordlines, ports, and sense amplifiers. The latter four all participate directly in passing the test data, so it is easy to see how they are tested. The address decoders, on the other hand, are tested in a slightly indirect manner. Since the write decoder and all read decoders should be receiving the same address and producing the same one-hot register entry, a fault in one of them will activate the wrong entry and produce an error on the output. It is possible that all ports suffer from the same error and thus produce the correct output, but this would be an exceedingly rare occurrence, and such a situation could still be detected in the final memory test of the bonded die stack. Thus, full test of the memory-less ports is achieved pre-bond.



(a) 2D Planar Version (20.3k μm^2)

(b) 3D Top (6.24k μm^2) (c) 3D Bottom (6.24k μm^2)

Figure 5: Planar and 3D layout for a six-port 16x8b register file. Despite the large area difference, these two designs have equal storage capacity.

4. EXPERIMENTS AND ANALYSIS

4.1 Power and Performance

To evaluate our test strategy on these two circuits, planar and 3D versions were implemented in 3DMagic [5], an extension to the open-source Magic VLSI tool [1], that enables the creation of 3D layouts. Both implementations were partitioned across two die layers. Our Kogge-Stone implementation is a full 64 bits as shown in Figure 4. To compute a 64-bit sum, the Kogge-Stone adder requires eight levels of logic. The first level, located at the top of the layout, computes the *generate* and *propagate* signals. The next six levels incremen-

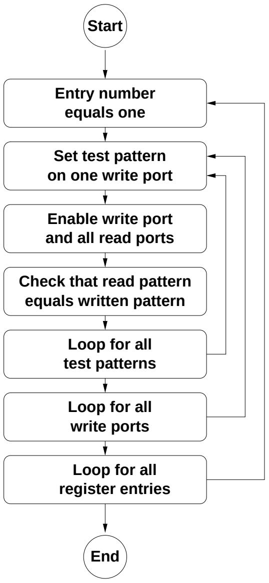

Start
Entry number equals one
Set test pattern on one write port
Enable write port and all read ports
Check that read pattern equals written pattern
Loop for all test patterns
Loop for all write ports
Loop for all register entries
End

Figure 6: Flowchart of the 3D register file test algorithm.

	2D Adder	3D Adder	%
Area (μm^2)	35.4k	23.5k	66%
Footprint (μm^2)	35.4k	11.8k	33%
Delay (ns)	7.46	6.08	82%
Power (mW)	26.1	22.6	87%

Table 1: This table lists the area, footprint, delay, and power requirements for the planar and 3D Kogge-Stone adders. The percentage listed is the ratio of 3D to planar.

tally gather the *p* and *g* signals to produce a carry for each bit. As Figure 4(a) demonstrates, this process is completely dominated by the wires shuffling the *p* and *g* signals around. The final logic level, located at the bottom of the layout, produces a summation from the carry bits.

We were able to extract the Kogge-Stone adder from Magic to produce a generic, lambda-based circuit description that can then be used with any transistor generation description. We then exported the extracted circuits to HSPICE and simulated them using a 130nm, level 49 transistor model. The power and performance numbers for the Kogge-Stone adder are presented in Table 1. The 3D adder obtains, simultaneously, a 18% cycle time and 13% power reduction. This means that a 3D adder can run at a significantly higher frequency than a planar version for equal power consumption, or it can run at equal speed for a nice power savings, depending on the needs of the design.

Our register file implementation shown in Figure 5 is a six-port (four read and two write ports), eight-bit, sixteen-entry design appropriate for a two-instruction-wide processor. The layout consists of four main components. First and most important is the actual SRAM cell array, which dominates each layout. Beside the SRAM array is the address decoder logic with six decoders per row, one per port. Above the array are the write drivers, two per column for the write ports. Last are the sense amplifiers below the array, four per column for the read ports. It is important to note that, within the SRAM array, each dark spot is a transistor. Because multi-ported register files are wire-dominated, the transistor density is very low and a lot of silicon is going to waste.

The 3D implementation, in contrast has a much higher transistor density and makes much better use of the available silicon. In this implementation, two read ports and one write port were placed on each layer. As reported in Table 2, the 3D implementation achieves the same memory capacity as the standard register file while significantly improving upon every metric. This 3D design consumes 40% less area and occupies a footprint over three times smaller, which may be a crucial objective for package-constrained system designs. Additionally, both power and delay are reduced. This once again offers the designer more speed for the same power level or a significant power reduction for the same performance as a planar design. This work verifies the power and performance results of the previous work [14] which were based on critical path estimations of the circuits.

		2D RF	3D RF	%
Area (μm^2)		20.3k	12.5k	61%
Footprint (μm^2)		20.3k	6.24k	31%
Delay (ps)	Read ’0’	1401	1043	74%
	Read ’1’	1407	1050	75%
	Write ’0’	520	308	59%
	Write ’1’	1381	735	53%
Energy (pJ)	Read ’0’	0.149	0.126	85%
	Read ’1’	0.149	0.127	85%
	Write ’0’	2.342	1.704	73%
	Write ’1’	2.342	1.710	73%

Table 2: This table lists the area, footprint, delay, and power requirements for the planar and 3D register files. The percentage listed is the ratio of 3D to planar.

Design		Pattern Count
2D Adder		313
3D Adder	Top	146
	Bottom	145
	Vias	10
	Total	301

Table 3: Listed are the pattern counts required to test each part of the design. These patterns were obtained from deterministic ATPG.

4.2 Test Cost and Coverage

To evaluate the test cost and coverage for the Kogge-Stone adder, we used the Mentor Graphics tool set. First, gate-level structural Verilog models of both the 2D and 3D implementations were produced and verified in ModelSim. For the 3D case, we produced three model files: one file describing the bottom layer, one file describing the top layer, and one file describing the via connections. This division of the model ensured an accurate description of the model was available for both pre- and post-bond test simulation.

The actual test simulation was produced using FlexTest. This tool provides a list of faults, a set of test vectors, and the fault coverage achieved. In order to achieve a fair comparison between the planar and 3D cases, we ran three fault simulations for the 3D implementations. The first two targeted all faults within the two independent layer models, simulating pre-bond test. The last simulation targeted faults on the via nets between the two layers, simulating a post-bond test verifying that the two die were successfully bonded. Summing the cost of these three tests estimates the total cost of testing the 3D design fairly,

The test simulation results are reported in Table 3. In confirmation of our earlier hypothesis, the combination of testing the top layer, bottom layer, and interconnecting vias required less patterns than testing the singular planar design. More importantly, note that the top and bottom layers, being independent DUTs during layer test, may be tested in parallel. This means that while the 3D design uses only 0.4% fewer patterns, it can be tested in just 156 cycles (146 for Top

Test Access		
Delay (ps)	Transmit ’0’	1346
	Transmit ’1’	1744
Energy (pJ)	Transmit ’0’	0.189
	Transmit ’1’	0.139

Table 4: Performance metrics for testing the top layer of the register file pre-bond. ‘Transmit’ means applying the test value to the write drivers and receiving that value from the SAs.

plus 10 for Vias) or in 49.8% of the time required for the 2D test.

The register file, being a RAM structure, requires a test methodology very different from the adder. Because this register file is a relatively small structure, we can reasonably apply a fairly complex test pattern. For comparison, we use Suk and Reddy’s Test B [2], adapted to multi-ported structures. The single-ported algorithm requires $16n$ accesses, where n is the number of bits (128 for our register file). To accommodate multiple ports, we multiple by $max(readports, writeports)$. This comes out to 8192 accesses to test the planar register file.

For our 3D register file, we apply Test B to the bottom layer (containing the state logic), requiring 4096 accesses. Implementing the algorithm described in Figure 6 requires $2n$ accesses, another 256 patterns. Of course, once the layers are bonded, we must test the via connections, which requires $4n$ or 512 patterns. Thus, in total, testing the 3D version of this register file requires just 4864 accesses, which is just 60% of the cost for planar test. In this case, simplifying the circuit with partitioning has greatly improved the test situation.

Performance metrics for the pre-bond test are given in Table 4. As these numbers show, the new test strategy we have proposed can be applied at nearly the same frequency and within the same power envelope as traditional planar test. This confirms that this test strategy is a viable solution to the challenge of pre-bond test.

5. RELATED WORK

Research in 3D-IC test area is still in the early stage. Mak [11] first identified several generic research directions in testing 3D circuits. In [8], Lewis and Lee proposed a scan-island based technique to enable pre-bond test for 3D microprocessors partitioned at the architectural level. Wu *et al.* [16] studied the scan chain ordering in 3D ICs for minimizing the total wire length. Jiang *et al.* [6] studied 3D-aware test access mechanisms by taking pre-bond test times into account to optimize the overall test time. More recently, Lee and Chakrabarty [7] overviewed the research challenges to be addressed in 3D-ICs to make them a market success.

6. CONCLUSION

This work investigated test strategies for circuit-partitioned 3D designs in which a functional unit can be partitioned into incomplete circuits across different die layers. Our techniques present standard scan registers that can be integrated into the layer scan chains, allowing the ATE to (in the standard scan case) directly test the circuit or (in the PRPG/MISR case) initialize the registers for BIST. To demonstrate our methodology, we performed two case studies using a prefixed parallel adder and a register file. In the case of the bit-split 3D Kogge-Stone adder, pre-bond test involved a simple extension to scan-based test. The port-split 3D register file was much more difficult, requiring a new test strategy to enable pre-bond test. Our full layout implementations confirmed the power and performance improvement estimates reported by previous work, and our fault simulations based on detailed Verilog models demonstrated high fault coverage at reduced cost compared to equivalent planar designs. We have shown that even the most difficult 3D partitioning schemes can be tested pre-bond, ensuring the viability of many-layer die stacks.

7. ACKNOWLEDGMENT

This research is supported in part by the C2S2 center of the SRC’s Focus Center Research Program and an NSF grant CCF-0811738.

8. REFERENCES

[1] Magic VLSI Layout Tool.
http://opencircuitdesign.com/magic/release.html.

[2] Magdy S. Abadir and Hassan K. Reghbati. Functional testing of semiconductor random access memories. *Computing Surveys*, 15(3), September 1983.

[3] B. Black, M. Annavaram, N. Brekelbaum, J. DeVale, L. Jiang, G. H. Loh, D. McCauley, P. Morrow, D. W. Nelson, D. Pantuso, P. Reed, J. Rupley, S. Shankar, J. Shen, and C. Webb. Die Stacking (3D) Microarchitecture. In *Proceedings of the 39th International Symposium on Microarchitecture*, 2006.

[4] Bryan Black, Donald 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, 2003.

[5] Shamik Das, Anantha Chandrakasan, and Rafael Reif. Design Tools for 3-D Integrated Circuits. In *Asia South Pacific Design Automation Conference (ASP-DAC)*, pages 53–56, 2003.

[6] Li Jiang, Lin Huang, and Qiang Xu. Test Architecture Design and Optimization for Three-Dimensional SoCs. In *Proceedings of the Design Automation and Test in Europe*, 2009.

[7] Hsien-Hsin S. Lee and Krishnendu Chakrabarty. Test Challenges for 3D Integrated Circuits. *To appear in IEEE Design and Test of Computers, Special Issue on 3D IC Design and Test*, Sep/Oct 2009.

[8] Dean L. Lewis and Hsien-Hsin S. Lee. A Scan-Island Based Design Enabling Pre-bond Testability in Die-Stacked Microprocessors. In *IEEE International Test Conference (ITC)*, October 2007.

[9] F. Li, C. Nicopoulos, T. Richardson, Y. Xie, N. Vijaykrishnan, and M. Kandemir. Design and Management of 3D Chip Multiprocessors using Network-in-Memory. In *Proceedings of the International Symposium on Computer Architecture*, 2006.

[10] G. H. Loh, Y. Xie, and B. Black. Processor Design in Three-Dimensional Die-Stacking Technologies. *IEEE Micro*, May/June 2007.

[11] T. M. Mak. Test challenges for 3D circuits. In *Proceedings of the IEEE On-Line Testing Symposium*, 2006.

[12] R. Patti, M. Hilbert, S. Gupta, and S. Hong. Techniques for producing three dimensional integrated circuits with high density interconnect. In *International VLSI Multilevel Interconnection Conference*, 2004.

[13] V. Pavlidis and E. Friedman. 3-d topologies for networks-on-chip. In *International SOC Conference*, pages 285–288, 2006.

[14] K. Puttaswamy and G. H. Loh. Thermal Herding: Microarchitecture Techniques for Controlling Hotspots in High-Performance 3D-Integrated Processors. In *Proceedings of the 13th International Symposium on High-Performance Computer Architecture*, 2007.

[15] Tezzaron. http://www.tezzaron.com/technology/fastack.htm. 2006.

[16] Xiaoxia Wu, Paul Falkenstern, and Yuan Xie. Scan Chain Design for Three-dimensional Integrated Circuits (3D ICs). In *Proceedings of the International Conference on Computer Design*, 2007.