Designing 3D Test Wrappers for Pre-bond and Post-bond Test of 3D Embedded Cores*

Dean L. Lewis
dean@gatech.edu

Shreepad Panth
shreepad.panth@gatech.edu

Xin Zhao
xinzhao@gatech.edu

Sung Kyu Lim
limsk@ece.gatech.edu

Hsien-Hsin S. Lee
leeh@gatech.edu

School of Electrical and Computer Engineering
Georgia Institute of Technology, Atlanta, GA 30332

ABSTRACT

3D integration is a promising new technology for tightly integrating multiple active silicon layers into a single chip stack. Both the integration of heterogeneous tiers and the partitioning of functional units across tiers leads to significant improvements in functionality, area, performance, and power consumption. Managing the complexity of 3D design is a significant challenge that will require a system-on-chip approach, but the application of SOC design to 3D necessitates extensions to current test methodology.

In this paper, we propose extending test wrappers, a popular SOC DFT technique, into the third dimension. We develop an algorithm employing the *Best Fit Decreasing* and *Kernighan-Lin Partitioning* heuristics to produce 3D wrappers that minimize test time, maximize reuse of routing resources across test modes, and allow for different TAM bus widths in different test modes. On average the two variants of our algorithm reuse 93% and 92% of the test wrapper wires while delivering test times of just 0.06% and 0.32% above the minimum.

1. INTRODUCTION

3D integration is an exciting new processing technology that allows multiple active silicon tiers to be stacked vertically. These tiers are tightly integrated with short, fast *3D vias* (3DVs, which may be either faceside microbumps or backside through-silicon vias). Even relatively simple 3D designs have big advantages; one of the most popular is the memory-on-logic stack, which places gigabytes of high-quality memory just a few clock cycles away from the processor[2, 13, 24]. Testing such designs is a challenge[9], but significant progress is being made in 3D-aware *design for testability* (DFT) as detailed in Section 3.

However, 3D integration design gets more interesting at finer partitioning granularities. Tezzaron Semiconductor is currently offering a 3D DRAM product with circuit-level partitioning; the bitlines within the arrays are partitioned across tiers, reducing the overall access latency. Puttaswamy and Loh proposed the Thermal Herding processor architecture[21], a bit-wise 3D partitioning of a processor core that reduces latency and power while controlling thermal density. The same research group produced 3D designs for many individual processor units such as arithmetic units, register files and caches[20]. For each the results were the same, significant reductions in both latency and power consumption. As 3DVs continue to shrink into the micron and submicron regime—Tezzaron is currently developing 0.4μm 3DVs[19]—these advanced 3D designs become ever more enticing.

In order for cores based on these designs to be commercialized successfully, 3D extensions to current embedded core test technologies are required. Much work has already been done in this area, also discussed in Section 3, but this work has been largely limited to *planar cores* (i.e., traditional cores implemented on a single tier) in a 3D stack. For a 3D *system-on-chip* (SOC) with 3D cores, additional DFT support is required. True 3D test wrappers must adapt to the varying demands of pre-bond and post-bond test. In this paper, we propose a new algorithm for designing 3D wrappers that meet these demands, optimizing for both total test time and for wirelength.

The rest of this paper is organized as follows. Section 2 presents the 3D wrapper optimization problem. Section 3 reviews the previous work in this area. Section 4 expounds on our methodology for optimizing 3D wrappers for test time and wirelength. Section 5 reports our experimental results. Section 6 concludes.

*This work was funded by NSF grant CCF-0811738 and by the DOD Center for Exceptional Computing

2. PROBLEM DEFINITION

In wrapper-based DFT, one or more *test access mechanisms* (TAMs) are used to connect the embedded IP cores to the external tester. Generally, a TAM is a bus of some width (called the *test width*) through which the tester loads and receives test data. A *test wrapper* is used interface between the multitude of scannable cells within the *core-under-test* (CUT) and the several bits of the TAM by organizing them into *wrapper chains*, one per TAM bit. The goal of wrapper design is to minimize the *total test time*, the time needed to apply all the required test patterns to the CUT. This time is given by $T = (\ell+1) \times (p+1)$, where ℓ is length of the longest wrapper chain and p is the number of test patterns required. p cannot be altered by the wrapper design, so then to minimize the total test time, we must minimize the length of the longest wrapper chain.

2.1 Motivating Example

Figure 1 illustrates the challenge and opportunity of wrapper design for 3D IP cores. In this example, the 3D CUT consists of two tiers (labeled CUT_1 and CUT_2). Each tier consists of two scan chains (lengths five and two in the top tier and lengths five and three in the bottom tier) which must be ordered within both the pre-bond and post-bond wrappers. The filled circles indicate the locations of 3DVs when they are necessary. Assume that the pre-bond test width for each tier is a single bit. The two scan chains on each layer are then necessarily stitched together by a single wire in the pre-bond test wrapper to form a single wrapper chain as shown in Figure 1(a) by the thick black lines. Herein lies the optimization opportunity: it is desirable to reuse these two stitching wires in the post-bond wrapper in order to reduce the total wrapper wire length.

Figure 1(b) and Figure 1(c) illustrate this opportunity. For these two solutions, the post-bond test access width is two bits. In Figure 1(b), the long scan chain on each tier is stitched to the short chain on the other tier; this solution fails to reuse the pre-bond stitching (shown by thick gray lines) and necessitates the use of two additional 3DVs dedicated to test. Figure 1(c) on the other hand does reuse this

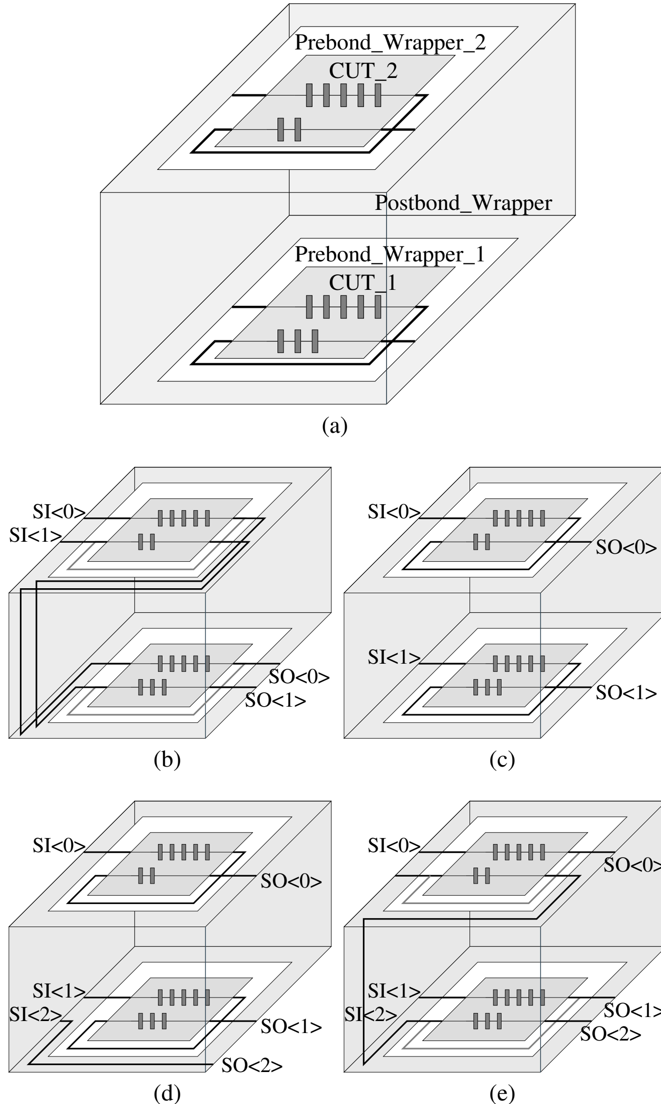

Figure 1: An example $3DP_W$ problem with four solutions (for TAM widths of two and three). (a) shows the pre-bond wrapper chain assignments. (c) and (e) are desired solutions while (b) and (d) are suboptimal.

stitching[1]. Both solutions are minimum test time for the given TAM width, so both solutions would be considered optimal solutions to the post-bond ordering problem under the test time objective. However, Figure 1(c) better minimizes the overall wirelength, so it is the superior solution.

The objectives then are to minimize the test time and the wirelength. We choose test time as the primary objective, as motivated by Figure 1(d) and Figure 1(e). For these two figures, the post-bond test access width is three bits. In Figure 1(d), wire length is the primary objective of the wrapper design algorithm, and the result is that one test bit is wasted[2]; even though that bit was assigned by the TAM architect to the CUT, it is simply left unused by the wrapper design algorithm because using it would increase the total wirelength. As a result, the test time increases significantly which is not acceptable because test time is a significant component of product cost. In Figure 1(e), test time is the primary objective instead. This solution does require additional wiring but the test time is significantly reduced, which is a more optimal design overall. Thus we choose wire length to be the secondary objective after test time when choosing our 3D test wrapper design algorithm.

2.2 Problem Formulation

We define the 3D IP wrapper design problem $3DP_W$ as follows. Given a 3D IP core test description (number of I/Os, number of scan chains, length of the scan chains, and a 3D partitioning of these resources), the set of pre-bond TAM widths, and the post-bond TAM width, determine an assignment of the I/Os and scan chains into both pre-bond and post-bond wrapper chains such that the test time is minimized and that the wirelength is minimized subject to the test time.

The wrapper design problem for planar cores was shown to be $\mathcal{NP}$-hard in [4]. The 3D wrapper design problem as defined requires the design of $N + 1$ wrappers (N pre-bond wrappers and a single post-bond wrapper). Thus this problem is also $\mathcal{NP}$-hard.

3. RELATED WORK

Wu et al.[25] designed 3D TAM architectures for optimal final stack test of 3D SOCs. Noia et al.[17] designed test time optimized wrapper chains for 3D cores. Jiang et al.[5] designed 3D TAM architectures that optimized the total test time if 3D SOCs; in a follow-up work[6], they designed independent but cooperative pre-bond and post-bond TAMs.

Marinissen et al.[16, 14] proposed extensions to the 1149.1 and 1500 standards for 3D ICs. This work allows the number of probe pads used in pre-bond test to differ from the number of inter-die test connections used in post-bond test but does not address how the die wrapper adapts to the diffing TAM widths. This work is being formalized as the proposed IEEE P1838 standard[1].

Koranne[7] proposed a reconfigurable test wrapper and a scheduling algorithm that utilizes this reconfigurability. Larsson and Peng[8] refined the design of reconfigurable wrappers to account for test power as well.

Quasem and Gupta[22, 23] proposed the union wrapper and a reconfigurable variant that allows inactive scan elements to be bypassed. Yoneda et al.[26] extended this reconfigurability to allow scan elements to migrate between wrapper chains within the union wrapper.

Here we present the first work in increasing the flexibility of a 3D test wrapper to operate optimally in both pre-bond and post-bond test with varying TAM bus widths. Our methodology is applicable to both planar cores with different pre-bond and post-bond TAM widths (as in [16]) and to more general true 3D cores that require a single post-bond wrapper and multiple pre-bond wrappers.

4. WRAPPER DESIGN ALGORITHM

4.1 Pre-bond Wrappers First

To design the 3D test wrappers, we use a three-step algorithm. We first describe its operation assuming the pre-bond wrappers are designed first and the post-bond wrapper second. We then discuss reversing this ordering at the end of this section. Briefly, the first step applies the *Best Fit Decreasing* (BFD) heuristic to design the pre-bond test wrappers for each tier. Step two uses the *Kernighan-Lin*

[1]SI and SO pins locations differ for clarity of the figure; in practice, these pin locations would be fixed as part of the contract between the wrapper designer and the TAM architect.

[2]The unused test bit could potentially be reassigned to another TAM as part of a wrapper-TAM co-optimization. This problem has been studied previously in [4], and the solution proposed there remains applicable in the 3D SOC case.

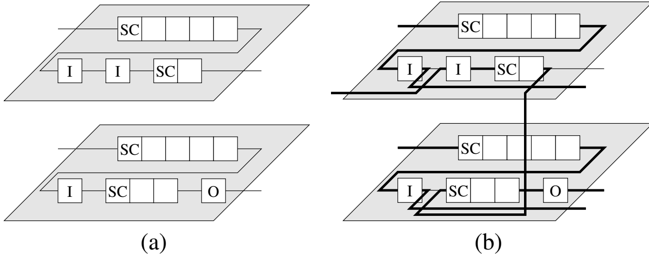

Figure 2: An example output of our algorithm: (a) the pre-bond wrapper assignments and (b) the post-bond wrapper assignment.

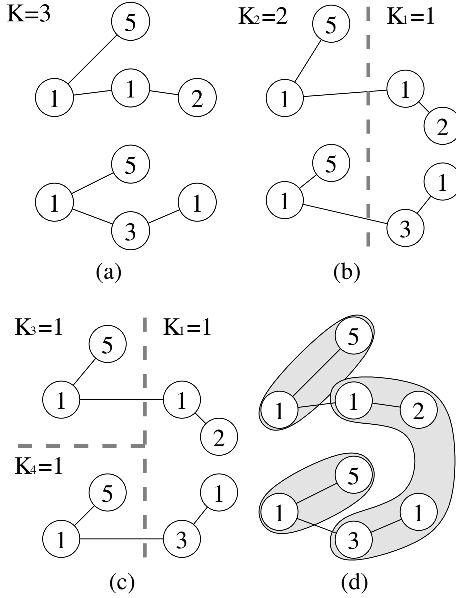

Figure 3: An example execution of our KL implementation on the pre-bond wrapper assignment from Figure 2(a). (a) shows the pre-bond assignment (input), (b) shows the first partition, (c) shows the second partition, and (d) shows the post-bond assignment (output).

Partitioning (KL) heuristic to determine optimal wrapper chain assignments for the post-bond wrapper. Finally, step three orders the post-bond wrapper chains to maximally reuse the pre-bond stitching. The output is a set of pre-bond wrapper assignments and a post-bond wrapper assignment; Figure 2 gives an example output, showing the pre-bond assignments in Figure 2(a) and the post-bond assignment in Figure 2(b).

4.1.1 Best Fit Decreasing

Designing each pre-bond wrapper is nearly identical to designing a planar wrapper. The only difference is the 3DVs. Here, the product engineer has a choice: either treat the 3DVs as pre-bond-untestable internal nets as in [10, 11], in which case they do not impact the wrapper design, or treat them as inter-core communications pins as in [16], in which case they are handled as any other boundary cell. This choice can be made on a 3DV-by-3DV case, designating each as is appropriate.

To solve the pre-bond wrapper design problem then, we use the BFD heuristic[4]. We choose this heuristic because it produces a near-optimal wrapper chain assignment while minimizing the use of TAM resources. BFD produces a set of wrapper chains composed of

KL Partitioning for Wrapper Design
Input: R - graph of pre-bond wrapper assignments
K - number of post-bond wrapper chains
Output: T - graph of post-bond wrapper assignments

1: `DesignWrapper`(Graph T, Graph R, int K)
2: **if** (K == 1) **then**
3: $T = T \cup R$; **return**
4: **for each**(scan element $se_i \in R$)
5: Assign se_i randomly to R_L or R_R
6: $K_L = \frac{K}{2}$; $K_R = K - K_L$
7: **while** (Balance is improving)
8: **while** (Have legal move)
9: R_V =*GreaterWeight*(R_L,R_R)
10: **if** (all $se \in R_V$ are locked) **then**
11: No legal move; **break**
12: **for each** (unlocked $se_i \in R_V$)
13: Calculate balance and cut gain
14: Move and lock se with highest balance gain
15: Record intermediate solution and gains
16: Search intermediate solutions for highest gain
17: **if** (all gains are negative) **then**
18: No longer gaining; **break**
19: Accept highest gain partition
20: Unlock all $se \in R_L$ and $\in R_R$
21: `DesignWrapper`(T, R_L, K_L)
22: `DesignWrapper`(T, R_R, K_R)
23: **return**

Figure 4: Pseudo code for applying KL partitioning to the wrapper design problem.

scan elements (the internal scan chains and I/O cells) and stitching wires between them. The goal of step two is to reuse these stitching wires to the greatest extent possible.

4.1.2 Kernighan-Lin Partitioning

To design the post-bond wrapper, we treat it as a partitioning problem, as shown in Figure 3. The input is a set of disjoint subgraphs and the required post-bond TAM width (Figure 3(a)). The subgraphs represent all the wrapper chains from all the tiers in the pre-bond assignments determined by the BFD algorithm previously. The vertexes represent the scan elements (weighted with the number of scan registers in that scan element), and the edges represent the stitching. The goal in designing the post-bond wrapper then is to determine a second set of disjoint subgraphs (the post-bond wrapper chains) such that

1. the maximum total weight of the vertexes in any subgraph is minimized (equivalent to minimizing the post-bond test time)
2. the greatest number of edges from the pre-bond subgraphs are reused in forming the post-bond subgraphs.

Formally, the input is an undirected graph R and a post-bond TAM bus width K. R is composed of a set of disjoint subgraphs, one subgraph per pre-bond wrapper chain per tier. Thus the number of subgraphs in R is $\sum_{i=1}^{n} k_i$, where n is the number of tiers and k_i is the number of pre-bond wrapper chains on the i-th tier. The output is an undirected graph T composed of K subgraphs representing the post-bond wrapper chain assignments.

The determination of the post-bond subgraphs is achieved through recursive application of the KL partitioning heuristic [12]. Psuedocode for applying KL to the 3D wrapper design problem is shown

in Figure 4. The optimization goals are represented by *balance* and *cut*. Balance is the ratio of the density of the first partition to the density of the second partition, where *density* is the ratio of the total weight of the scan cells in a partition to the number of wrapper chains assigned to that partition; an *overdense* partition has too many scan cells which would lead to a long test time while an *underdense* partition can accept more scan cells without affecting test time. The ideal balance is one, which indicates that all wrapper chains can have the same number of scan cells, a solution which offers the shortest test time. Cut is the number of edges in the post-bond subgraphs that do not overlap pre-bond edges. The ideal cut is zero, which indicates that no additional post-bonding stitching is required.

Our implementation of KL is initialized with all the scan elements from every layer grouped into a single pool and all the wrapper chains available for assignment[3] (exemplified in Figure 3(a), which is the graph representation of the pre-bond wrappers from Figure 2(a)).

Each KL step begins by assigning half of the available wrapper chains to each partition. Next the scan elements are randomly assigned to each partition while maintaining balance as best as possible. Next is the moving phase. Each unlocked scan element in the denser partition is evaluated, and the move producing the best balance is accepted (ties are broken with the cut gain). This is repeated until no unlocked scan elements are available in the denser partition. All discovered partitionings are evaluated and the one with the best balance is accepted (ties are once again broken by cut). All the scan elements are unlocked and the moving phase is repeated. This continues until no more gains in balance or cut are achieved. Figure 3(b) shows the result of a single KL step, where the scan elements have been partitioned into two sets, balanced according to the TAM width assigned to each partition.

The final step is recursion, where each partition is further subdivided into smaller partitions. This is shown in Figure 3(c), where the left partition from before is further partitioned into two subsets. Recursion halts when only a single TAM bit is assigned to a given partition. The scan elements in that partition are then assigned to that wrapper chain (Figure 3(d)).

4.1.3 Scan Element Pairing

Once the wrapper chain assignments are complete, the final step is to order the scan elements within the chains—both in the pre-bond and the post-bond wrappers—so as to minimize the cut. This simply requires searching the list of scan elements in the post-bond wrappers, identifying all those that are assigned to the same pre-bond wrapper chains, and stitching them together accordingly (Figure 2(b)). Final ordering of these short pre-stitched chains is a simple matter that can be handled with any traditional ordering scheme[15] and so is not discussed further here.

4.2 Post-bond Wrapper First

It is a trivial matter to reverse the order of design. In this case, we first determine the post-bond wrapper by applying the BFD heuristic to the complete set of scan elements on all tiers. The subgraphs representing the post-bond wrapper chains are then used to guide the design of pre-bond wrapper chains. Now KL is executed for each tier with the goal of producing maximally-balanced pre-bond subgraphs that maximally overlap the given post-bond subgraphs. Finally, the wrapper chains must be ordered as before.

[3]The scan elements are all grouped into one large pool regardless of tier because we consider 3DVs to be free resources. This is justified by the submicron size of present day state-of-the-art 3D processing as mentioned in Section 1.

	Tiers	Cells per Tier	Chains per Tier
ckt1	2	3016, 3021	6, 6
	4	1507, 1512, 1510, 1508	3, 3, 3, 3
ckt2	2	5329, 3479	11, 7
	4	2543, 1980, 2767, 1518	5, 4, 6, 3
ckt3	2	19,890, 19,228	40, 39
	4	9826, 9172, 10,757, 9363	20, 18, 22, 19
ckt4	2	37,359, 40,751	75, 82
	4	20,723, 18,135, 17,011, 22,241	41, 36, 34, 44

Table 1: Test complexity of the four benchmarks

.

5. EXPERIMENTS

5.1 Experimental Setup

To test our algorithm, we used a custom collection of benchmark circuits taken from the OpenCores database[18] as listed in Table 1. This benchmark suite includes a 80386 processor, a DES encryption engine, and two 256-bit pipelined multipliers of differing pipeline depths. These circuits were picked so as to cover a large range of embedded core complexities. To obtain the 3D placements of the scan chains, we first compiled the design with Design Compiler from Cadence. Next we partitioned the circuits with a custom FM partitioner[3] and performed 3D placement with Encounter from Cadence. Finally, Design Compiler was again used to partition and route the scan chains.

We developed our program in C++ and executed the benchmarks on a 2.40GHz Intel Xeon processor with 1GB RAM. No wrapper design took more than a few seconds to complete.

5.2 Methodology

To evaluate our algorithm, we ran a series of tests using different design modes and different wrapper configurations. Most importantly are the three design tools:

1. All BFD (`BFD`)—the BFD heuristic is used to design both the pre-bond and the post-bond wrappers with no feedback between the two processes. This is our baseline case.

2. Pre-bond First (`PRE`)—the pre-bond first variant of our algorithm: the BFD heuristic is used to design the pre-bond wrappers. These designs then drive the KL heuristic in designing the post-bond wrapper.

3. Post-bond First (`POST`)—the post-bond first variant of our algorithm: the BFD heuristic is used to design the post-bond wrapper. This design then drives the KL heuristic in designing the pre-bond wrappers.

To test our algorithm under different design constraints, we vary the number of TAM bits assigned to each wrapper. For the circuits *ckt1*, *ckt2*, *ckt3*, *ckt4*, we vary the post-bond TAM width from one to twelve, eighteen, forty, and sixty respectively. For each post-bond TAM width we run three experiments:

1. Half-width (`05`)—the total pre-bond TAM width is half the post-bond TAM width

2. Even-width (`10`)—the total pre-bond TAM width is equal to the post-bond TAM width

3. Double-width (`20`)—the total pre-bond TAM width is double the post-bond TAM width

Here, the *total pre-bond TAM width* is the sum of the TAM widths assigned to each tier. In assigning TAM bits to each pre-bond wrapper, we divide the total TAM width as evenly as possible.

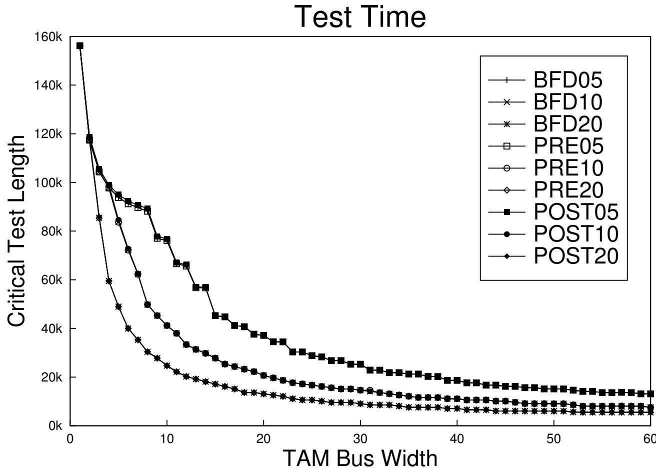

Figure 5: CTL versus post-bond TAM width for *ckt4* in four tiers.

Finally, for each experiment, we design 3D wrappers for both the two-tier and four-tier implementations of each circuit.

5.3 Results

In this section, we report two metrics. The first is the *critical test length* (CTL), which is defined as the sum of the length of the longest wrapper chain in each pre-bond wrapper and length of the longest wrapper chain in the post-bond wrapper. Since the total test time is the number of test patterns multiplied by the longest wrapper chain, the longest chain is proportional to the total test time. We therefore use the CTL metric because it correlates directly to the total test time for a 3D stack (i.e., pre-bond test time plus final stack test time) and thus represents the primary minimization objective of the wrapper design algorithms. A superior wrapper is one with a shorter CTL.

The second metric is the *cut*, which is defined as the number of stitching wires in the pre-bond test wrappers that are *not* reused in the post-bond wrapper, basically the number of wires not reused in the post-bond wire routing. We choose this metric because fewer reused wires correlates to greater wrapper wirelength and routing congestion and represents our secondary minimization objective. A superior wrapper is one with a smaller *cut*.

Figure 5 shows the CTL results for the four-tier design of *ckt4*. We show only one CTL graph because the results for all the other benchmarks and configurations show the exact same trends. The CTL drops continuously with some plateauing (beginning at local Pareto optimal points). `05` wrappers have the longest CTLs with `10` wrappers doing better and with `20` wrappers better still. This is simply because those designs have more pre-bond TAM bits and so shorter wrapper chains.

More importantly is how the CTL curves for the `PRE`- and `POST`-designed wrappers fit the `BFD` curves almost exactly (the match is so close that the nine results in Figure 5 appear to be just three). Since the BFD algorithm is producing near-minimum test time wrappers, this close fit demonstrates that our KL-based algorithm successfully minimizes the total test time as well. On average, `PRE` and `POST` CTLs are just 0.06% and 0.32% longer than `BFD` respectively. In the worst case, they are still just 4.2% and 3.0% longer respectively, and a product engineer could avoid these worse cases by simply running our algorithm several times on the same input set, utilizing the random initial partitions in the KL step.

The results for *cut* are shown in Figure 6. In these polar graphs, the angle represents the post-bond TAM width (normalized to the $[0, 2\pi]$ range), and the radius represents the *cut*. The greater the distance from the center, the higher the *cut* and so the worse the solution. Also

	Tiers	ckt1	ckt2	ckt3	ckt4	ALL
`BFD`	2	52%	15%	23%	16%	27%
	4	63%	53%	35%	31%	
`PRE`	2	12%	5.8%	5.0%	6.7%	6.6%
	4	15%	7.6%	5.0%	7.4%	
`POST`	2	13%	4.0%	7.6%	8.8%	8.4%
	4	16%	6.1%	7.3%	11%	

Table 2: Average percentage of pre-bond stitches for each experiment and for each method overall.

shown are four rings, indicating the max possible *cut* and the averages for `BFD`, `PRE`, and `POST`; these averages are also listed in Table 2.

In general, the results for `BFD` (plotted with the asterisk-style icons) are chaotic. Sometimes the *cut* is very low, and sometimes it is very high, but in general the results do not cluster at any one radius. Since there is no communication between the pre-bond and post-bond design steps, this result is expected. Sometimes the design tool gets lucky and groups the same scan chains together in both wrappers; sometimes it splits them up. `BFD` averages a 27% cut of the pre-bond stitching; it simply cannot reliably produce a low-*cut* design.

In significant contrast, both the `PRE` (represented by the open icons) and `POST` (represented by the filled icons) design tools consistently produce low-*cut* 3D wrappers. This result is highlighted by the tight clustering of these data points in the middle of the plots. These tools are not perfect; at some design points the *cut* spikes up significantly. This is attributed to the second-class nature of the *cut* objective. Because our tool is designed to minimize the maximum wrapper chain length first, the *cut* is sometimes sacrificed to create a shorter wrapper chain. These outliers in the *cut* clusters can be used to inform the TAM architecture design; if wirelength or routing congestion are concerns in a particular wrapper design, assigning an additional test bit or two could help reduce the problem.

The other important point to note is that while the `PRE` and `POST` design tools both produce consistently low-*cut* wrappers, the `PRE` tool in general is the better of the two (6.6% on average compared to 8.4% for `POST`) as evidenced by the slightly tighter clustering of the `PRE` results about the origin. We attribute this to the different scopes each method gives BFD and KL. The `POST` algorithm applies BFD to the global post-bond wrapper design problem and then KL to the local pre-bond wrapper design problems. In doing so, `POST` necessarily limits the opportunity for KL to optimize the pre-bond wrapper. In the worst case, every single scan chain in the post-bond wrapper would be stitched to scan chains from other tiers. This would leave KL with no opportunities to reuse the post-bond connections in the pre-bond wrappers. Conversely, the `PRE` tool applies BFD to the local problem and KL to the global problem. Unlike in `POST`, KL applied to the post-bond design problem is unrestricted by the physical layout of the stack and so is free to reuse any of the pre-bond stitches created by the BFD algorithm.

6. CONCLUSION

In this paper we have presented a methodology for designing 3D test wrappers for embedded 3D IP cores. We use the *Best Fit Decreasing* and *Kernighan-Lin Partitioning* heuristics to design flexible test wrappers that can adjust to varying test modes like pre-bond and post-bond test. This flexibility results in a lower total test time for the CUT and reduced wiring resource consumption in the 3D wrapper design—the `PRE` design tool reuses 93% of the pre-bond stitching while sacrificing just 0.06% of the minimum possible test time. Our methodology is applicable to both true embedded 3D cores and to simpler planar embedded cores in cases where variable TAM bus widths are a useful design feature.

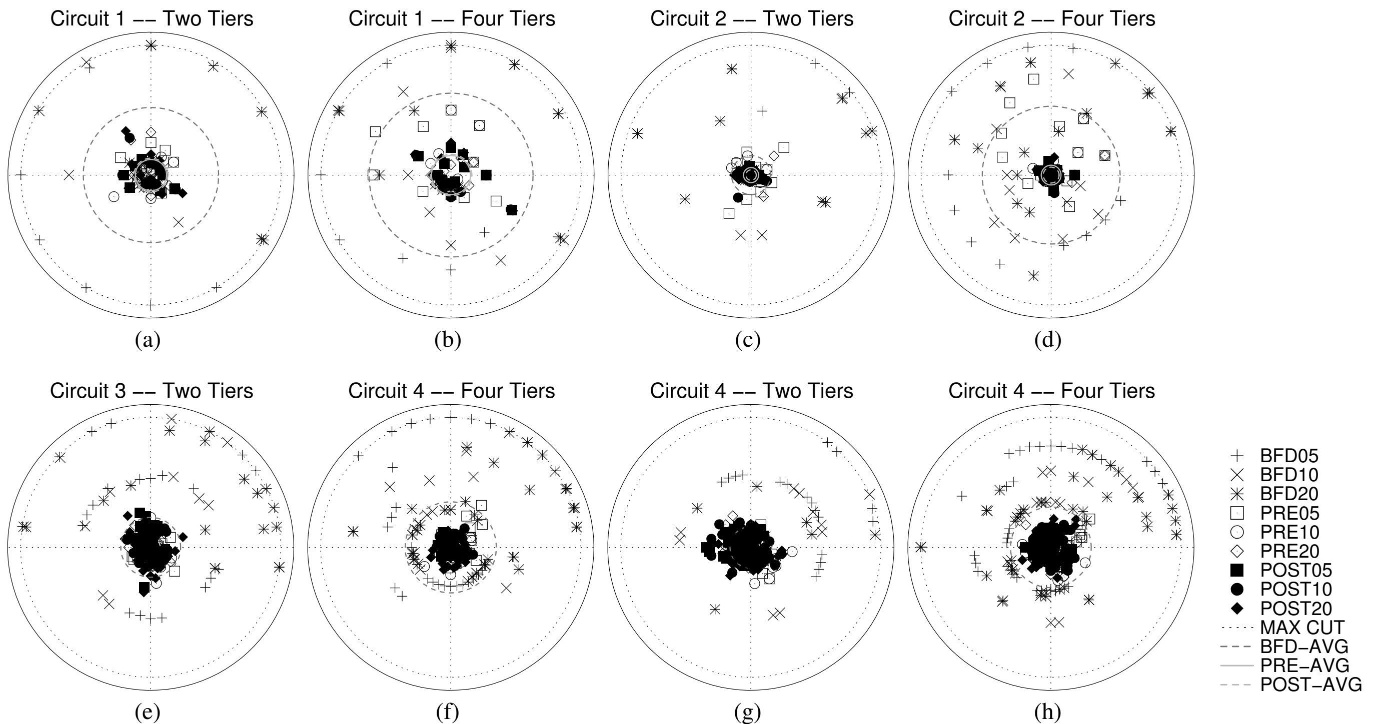

Figure 6: Polar plots of *cut* versus post-bond TAM width for the four circuits. The four rings highlight the averages and max *cut*.

7. REFERENCES

[1] Test Access Architecture for Three-Dimensional Stacked Integrated Circuits. *Proposed IEEE Standard P1838*.

[2] B. Black *et al*. Die stacking (3D) microarchitecture. In *Proceedings of the International Symposium on Microarchitecture*, 2006.

[3] J. Cong and S. K. Lim. Multiway partitioning with pairwise movement. In *Proceedings of the International Conference on Computer-Aided Design*, pages 512–516, 1998.

[4] V. Iyengar, K. Chakrabarty, and E. J. Marinissen. Test wrapper and test access mechanism co-optimization for system-on-chip. *Journal of Electronic Testing: Theory and Applications*, 18(2):213–230, 2002.

[5] L. Jiang, L. Huang, and Q. Xu. Test architecture design and optimization for three-dimensional SOCs. In *Proceedings of the Design, Automation, and Test in Europe Conference*, 2009.

[6] L. Jiang, Q. Xu, K. Chakrabarty, and T. M. Mak. Layout-driven test-architecture design and optimization for 3D SOCs under pre-bond test-pin-count constraint. In *Proceedings of the International Conference on Computer-Aided Design*, 2009.

[7] S. Koranne. A novel reconfigurable wrapper for testing of embedded core-based SOCs and its associated scheduling algorithm. *Journal of Electronic Testing: Theory and Applications*, 18(4):415–434, 2002.

[8] E. Larsson and Z. Peng. A reconfigurable power-conscious core wrapper and its application to SOC test scheduling. In *Proceedings of the International Test Conference*, pages 1135–1144, 2003.

[9] H.-H. S. Lee and K. Chakrabarty. Test Challenges for 3D Integrated Circuits. *IEEE Design and Test of Computers, Special Issue on 3D IC Design and Test*, 26(5):26–35, Sep/Oct 2009.

[10] D. L. Lewis and H.-H. S. Lee. A Scan-Island Based Design Enabling Pre-bond Testability in Die-Stacked Microprocessors. In *Proceedings of the International Test Conference*, October 2007.

[11] D. L. Lewis and H.-H. S. Lee. Testing Circuit-Partitioned 3D IC Designs. In *IEEE Computer Society Annual Symposium on VLSI*, 2009.

[12] S. K. Lim. *Practical Problems in VLSI Physical Design Automation*. Springer Verlag, 2008.

[13] G. H. Loh. 3D-stacked memory architectures for multi-core processors. In *Proceedings of the International Symposium on Computer Architecture*, pages 453–464, June 2008.

[14] E. J. Marinissen, C.-C. Chi, J. Verbree, and M. Konijnenburg. 3D DFT architecture for pre-bond and post-bond testing. In *Proceedings of the International 3D System Integration Conference*, November 2010.

[15] E. J. Marinissen, S. K. Goel, and M. Lousberg. Wrapper design for embedded core test. In *Proceedings of the International Test Conference*, pages 911–920, 2000.

[16] E. J. Marinissen, J. Verbree, and M. Konijnenburg. A structured and scalable test access architecture for TSV-based 3D stacked ICs. In *Proceedings of the VLSI Test Symposium*, pages 269–274, April 2010.

[17] B. Noia, K. Chakrabarty, and Y. Xie. Test-wrapper optimization for embedded cores in TSV-based three-dimensional SOCs. In *Proceedings of the International Conference on Computer-Aided Design*, 2009.

[18] OpenCores. http://opencores.org. January 2011.

[19] B. Patti. Keynote address: Testing in a new dimension. In *IEEE International Workshop on Testing Three-Dimensional Stacked Integrated Circuits*, November 2010.

[20] K. Puttaswamy. *Designing High-Performance Microprocessors in 3-dimensional Integration Technology*. PhD thesis, School of Electrical and Computer Engineering, Georgia Institute of Technology, 2007.

[21] K. Puttaswamy and G. H. Loh. Thermal Herding: Microarchitecture techniques for controlling hotspots in high-performance 3D-integrated processors. In *Proceedings of the International Symposium on High-Performance Computer Architecture*, 2007.

[22] M. S. Quasem and S. Gupta. Designing multiple scan chains for systems-on-chip. In *Proceedings of the Asian Test Symposium*, pages 424–427, 2003.

[23] M. S. Quasem and S. Gupta. Designing Reconfigurable Multiple Scan Chains for Systems-on-Chip. In *Proc. of VLSI Test Symposium*, 2004.

[24] D. H. Woo, N. H. Seong, D. L. Lewis, and H.-H. S. Lee. An Optimized 3D-Stacked Memory Architecture by Exploring Excessive, High-Density TSV Bandwidth. In *Proc. of the International Symposium on High-Performance Computer Architecture*, 2010.

[25] X. Wu, Y. Chen, K. Chakrabarty, and Y. Xie. Test-access mechanism optimization for core-based three-dimensional SOCs. In *Proceedings of the International Conference on Computer-Aided Design*, 2009.

[26] T. Yoneda, M. Imanishi, and H. Fujiwara. An SOC test scheduling algorithm using reconfigurable union wrappers. In *Proceedings of the Design, Automation, and Test in Europe Conference*, 2007.