# Timing Aware Wrapper Cells Reduction for Pre-bond Testing in 3D-ICs

Pei-An Ho

Yen-Hao Chen

Allen C.-H. Wu

TingTing Hwang

james552214@gmail.com yhchen@cs.nthu.edu.tw allenwuuw@gmail.com tingting@cs.nthu.edu.tw Department of Computer Science, National Tsing Hua University, Taiwan

# ABSTRACT

Three dimensional integrated circuits (3D-ICs) have being developed to improve existing 2D designs by providing smaller chip areas, higher performance and lower power consumption. With the short and dense through-silicon-vias (TSVs), multiple dies can be integrated to overcome the barrier of interconnection. However, before 3D-ICs become a viable technology, the understanding of 3D testing issues is still insufficient and there are still many unresolved testing challenges. To ensure the stack yield of future adopting of 3D-ICs, pre-bond testing is needed to provide the known good die (KGD). Since the TSVs are not fully accessible prior to bonding, testing the combinational logic between the scan flip-flops and TSV becomes a complex issue. In order to overcome the limitation of TSV, additional wrapper cells were inserted at the two ends of TSVs to provide controllability and observability [1], [2]. Even though it is a major breakthrough solution for pre-bond testing, the wrapper cells used by the TSVs lead to significant area overhead. Further, to reduce area overhead, the existing primary scan flip-flops were reused to achieve high testability [3], [4]. However, practical timing considerations were overlooked and the number of inserted wrapper cells was still high. In this paper, we present an enhanced method to not only generate more reused scan flip-flops but also not incur any timing violation by using an accurate timing model. Furthermore, testability constraints are also considered and can be traded-off between area overhead and testability. The experimental results on ITC99 benchmark circuits [5] have shown that our method can reduce 0.92%-6.01% wrapper cells with competitive testability compared to the previous work.

#### I. INTRODUCTION

As integrated circuit technology continues to scale to a smaller feature size, the performance of integrated circuit designs has improved [6]. However, the global connections do not scale accordingly with technology and have become the critical bottleneck of chip performance [7]. New technology solutions are needed to overcome the limitation of current process in order to improve the interconnection delays and power consumption.

Three dimensional integrated circuit (3D-IC) has been proposed to overcome the barriers in interconnection scaling. The 3D-ICs partition designs into multiple silicon dies and bond (or stack) dies vertically using through-silicon-vias (TSVs) as shown in Figure 1, where the circuit is partitioned into *die0* and *die1* and connected through TSVs. The TSVs are high-density and low-capacitance vertical wires connecting dies and offer lower parasitic losses and higher I/O density [8]. Thus, 3D-ICs utilizing TSVs may achieve power-consumption reduction and system performance improvement [9], [10]. However, manufacturing defects of 3D-ICs often occur during fabrication processes [11] including micro-bump formation, bonding of the support substrate, wafer thinning, TSV formation, debonding of the support wafer, dicing of the thinned wafer, chip stacking, etc. Many 3D-IC testing methodologies for defects are proposed to achieve high-yield

fabrication. Among these, pre-bond testing detects TSV defects, e.g. impurities and voids, and provides known good dies (KGDs) while post-bond testing ensures that stacked chips are not mechanically or thermally damaged, i.e. bonding failures. The pre-bond testing avoids bonding defective dies and is particularly important to the fabrication yield rate of 3D-ICs [11].



Fig. 1. A 3D-IC partitioning example where the circuit is partitioned into two 3D-IC dies.

The scan chain technique uses scan flip-flops to provide testability and has been widely adopted in conventional 2D-IC designs. Also, the automatic test pattern generation (ATPG) can generate a set of test patterns to detect defects [12]. A good design for testability (DFT) may cover as many manufacturing defects as possible, i.e. fault coverage. In general, more test patterns can detect more defects and provide a high fault coverage. However, the number of test patterns directly affects the testing time and cost. Thus, a design with high fault coverage and small number of test patterns is more desirable. Apart from 2D-ICs, TSVs of 3D-ICs are left floating and not accessible before the bonding (or stacking) step. Thus, TSVs can neither be controlled nor observed directly by conventional scan chains of 2D-ICs, and the testability of 3D-IC dies is decreased. In order to detect defects in pre-bond testing, appropriate design for testability (DFT) solutions are required to provide controllability and observability of TSVs.

Die level wrapper cells have been proposed [13] to increase the testability of pre-bond testing as shown in Figure 2. In this figure, TSV input signals which are outputs of a die (defined as *outbound* TSVs) cannot be observed, and TSV output signals which are inputs of a die (defined as *inbound* TSVs) cannot be controlled because the TSV has not been connected at pre-bond testing. Thus, two wrapper cells are added at two ends of TSVs to provide controllability and



Fig. 2. Two wrapper cells are inserted at both ends of a TSV to provide testability. [1], [2]

observability [1], [2]. However, due to large number of TSVs on 3D-ICs, inserting wrapper cells at both ends of all TSVs may lead to a significant area overhead. Instead of inserting additional wrapper cells, J. Li et al. [3] proposed to reuse the existing scan flip-flops as some wrapper cells of TSVs to reduce the area overhead. In this approach, additional wrapper cells are inserted only when no existing scan flip-flops can be reused as wrapper cells. However, this approach does not provide insights on the minimum number of wrapper cells. M. Agrawal et al. [4] proposed a wrapper cell minimization problem (WCM), which belongs to NP-hard problems, and used a heuristic method to solve the problem. In this paper, we propose an enhanced heuristic method to find the minimum number of wrapper cells with an accurate timing model on scan flip-flops to provide testability. In our method, testability constraints are defined to provide a tradeoff between testability and area overhead.

The rest of this paper is organized as follows. Section II reviews previous work and gives our motivation. The problem formulation is presented in Section III. In Section IV, we illustrate our graph construction method and heuristic algorithm. Experimental results are presented in Section V. Finally, Section VI concludes the paper.

### II. PREVIOUS WORK AND MOTIVATION

In pre-bond testing, the TSV output signals are not controllable, and the TSV input signals are not observable. In this paper, we refer a TSV output signal as *inbound* TSV which drives logic gates and a TSV input signal as *outbound* TSV which is driven by other logic gate as shown in Figure 2. To provide testability, two wrapper cells are added at both *inbound* and *outbound* TSVs at pre-bond testing of 3D-ICs [13]. However, inserting wrapper cells leads to significant die area overhead and increases timing delay on functional paths.

J. Li et al. proposed to reuse existing scan flip-flops to reduce additional wrapper cells [3]. A scan flip-flop can be reused as a wrapper cell of an *inbound* TSV by adding a multiplexer as shown in Figure 3 (a). Similarly, a scan flip-flop can be reused as a wrapper cell of an *outbound* TSV by adding a multiplexer and an XOR gate as shown in Figure 3 (b). For each scan flip-flop, their method carefully selects a pair of scan flip-flop and *inbound/outbound* TSV without overlapped fan-in/fan-out cones to avoid testability degradation as shown in Figure 4 (a). However, the method proposed by J. Li et al. [3] reuses each scan flip-flop only once rather than multiple times.



Fig. 3. Reusing existing scan flip-flops as wrapper cells of TSVs. (a) Reusing a scan flip-flop as a wrapper cell of an *inbound* TSV. (b) Reusing a scan flip-flop as a wrapper cell of an *outbound* TSV.

M. Agrawal et al. tried [4] to reuse the scan flip-flop multiple times and formulated a wrapper cell minimization problem (WCM) to further reduce additional wrapper cells. They also proved that the WCM belongs to NP-hard problems, and hence proposed a method to translate the problem into a minimal clique-partitioning problem. Then, they use a heuristic algorithm to solve the problem. However,



Fig. 4. Apart from previous works, our method allows overlapped fan-in/fanout cones with testability constrains. (a) A scan flip-flop can be reused as a wrapper cell of an *inbound* TSV safely without testability lost if the fan-out cones are non-overlapped. (b) Even with overlapped fan-out cones, sharing a scan flip-flop as a wrapper cell may not have testability lost.

Agrawal's method [4] does not consider which TSV set, i.e. inbound or outbound TSV sets, should be processed first. Since the number of usable scan flip-flops will affect the testability of TSVs, the order of processing inbound and outbound TSVs matters. Also, it considers only the capacity load without wire delay of scan flip-flops. When reusing scan flip-flops as wrapper cells, long distance wires may increase the capacity load of inbound TSVs and affect timing slack of outbound TSVs. As a result, the 3D-IC design may result in timing violation after inserting the wrapper cells. Furthermore, Agrawal's method [4] prevents testability degradation by not sharing a scan flipflop to a TSV with overlapped fan-in or fan-out cones as shown in Figure 4 (a). However, the testability, e.g. fault coverage and test pattern count, may be not affected when sharing the scan flip-flops as wrapper cells with overlapped fan-in or fan-out cones. Take Figure 4 (b) as an example, the stuck-at-0 fault can still be observed with overlapped fan-out cones of the scan flip-flop and TVS. Thus, sharing a scan flip-flop with overlapped fan-in/fan-out cones may not cause serious testability degradation.

In this paper, we target the problem of minimizing the number of wrapper cells (WCM). We are going to use a method that maps the problem into a minimal clique-partitioning problem and propose a heuristic clique-partitioning algorithm to solve the problem. We enhance the flow of DFT insertion with the step of deciding the *inbound* TSVs or *outbound* TSVs to be processed. Furthermore, an accurate timing model is constructed with the scan flip-flop capacity load information and detailed wire delay information. Also, apart from the previous works, our method reuses scan flip-flops to TSVs with overlapped fan-in and fan-out cones with testability constraints, i.e. fault coverage and number of test patterns.

# III. THE WRAPPER CELL MINIMIZATION PROBLEM (WCM) [4]

M. Agrawal et al. [4] have proven that the wrapper cell minimizing problem (WCM) belongs to NP-hard problems and translate the WCM problem to the minimal clique-partitioning problem. The minimal clique-partitioning problem is: given a graph G(N, E), partition the graph into several cliques such that the total number of cliques is minimal, where a clique is a subgraph in which all nodes are connected to each other, i.e. a complete graph.

Given a circuit die, the wrapper cell minimizing problem (WCM) can be transformed into a minimal clique-partitioning problem of graph G(N, E). Let the scan flip-flop set be  $scan_flipflops$ , the *inbound* TSV set *inbound\_TSVs*, the *outbound* TSV set *outbound\_TSVs*. The TSV set TSVs is the union of the *inbound* TSV set and *outbound\_TSVs* uset, i.e.  $TSVs = inbound_TSVs \cup$ 

 $outbound\_TSVs$ . The node set N of graph G is the union of the scan flip-flop set and TSV set, i.e.  $N = scan\_flipflops \cup TSVs$ . A node in  $scan\_flipflops$  has an edge to any node in TSVs if and only if the scan flip-flop can be used as a wrapper cell of the TSV. Any two nodes in TSVs have an edge connection if and only if the two TSVs can share a single wrapper cell (either a reused scan flip-flop or an additional wrapper cell). Apart from the previous works, two TSVs may share a single wrapper cell as long as the testability constraint can be satisfied even with overlapped fan-in or fan-out cones.

An optimal clique-partitioning solution of graph G is an optimal wrapper cell minimizing problem (WCM) solution. For all cliques with a node in  $scan_flipflops$ , the TSVs in the same clique can share the scan flip-flop as their wrapper cell to achieve testability. For the other cliques without a node in  $scan_flipflops$ , an additional wrapper cell is inserted for each clique. Thus, the number of additional wrapper cells of the WCM solution is the number of cliques without a node in  $scan_flipflops$  of the minimal clique-partitioning solution. Also, the number of cliques with a node in  $scan_flipflops$  equals to a constant value, i.e. the number of scan flip-flops. On the other words, the number of additional wrapper cells equals to the number of cliques subtracting to a constant value. Hence, the optimal solution of clique-partitioning problem is the optimal solution of WCM.

Figure 5 shows an example of the wrapper cell minimization problem (WCM) and the minimal clique-partitioning problem. By the above formulation, there is no edge between any two scan flip-flop nodes. Also, an optimal solution of the clique-partitioning problem is an optimal solution of the wrapper cell minimization problem. In this case, there are five cliques which indicate five wrapper cells. Also, TSVs in the same clique can share a wrapper cell with each other. From the figure, two cliques have a scan flip-flop which means that they can reuse the scan flip-flops as their wrapper cells. In contrast, the other three cliques do not have a scan flip-flop which means that three additional wrapper cells are inserted.



Fig. 5. A minimal clique-partitioning problem.

## IV. PROPOSED METHOD

Figure 6 shows the design flow of our proposed method. The steps in shade are new steps in our global flow. Firstly, the netlist is synthesized. Then, the TSV sets, i.e. *inbound* and *outbound* TSVs, are analyzed. After that, the graph is constructed regarding to the scan flip-flops, *inbound* TSVs, and *outbound* TSVs. In this step, the problem is transformed into a minimal clique-partitioning problem. We then apply our proposed graph construction flow with an accurate timing model and testability constraints. Next, we propose a heuristic clique-partitioning algorithm to solve the problem. After that, a testable netlist with reused and additional wrapper cells is generated. Also, a commercial automatic test pattern generation (ATPG) tool is adopted to further examine the fault coverage to check whether the testability requirement of all TSVs is satisfied. Finally, a 3D physical design flow and static timing analysis is performed.



Fig. 6. The global design flow.

## A. TSV Analysis

We observe that the ordering of *inbound* and *outbound* TSV sets may affect the solution quality. As shown in Table I, fault coverage is improved with less additional wrapper cells if starting from the larger TSV set. Hence, in the following experiments, the procedure starts from the larger TSV set.

#### B. Graph Construction

Algorithm 1 shows the graph construction steps which consists of two parts, namely node construction part and edge construction part. In the node construction part, a node n is added if (1) it is a scan flip-flop, (2) it is an inbound TSV and the capacity load is smaller than a predefined threshold  $cap_th$  (from cell library), or (3) it is an outbound TSV and the slack is larger than an user defined threshold s\_th. In the edge construction part, any two nodes where at least one of them in TSVs have an edge if they can share a wrapper cell. In the algorithm, an edge is constructed only if distance of two nodes is less than an user defined threshold d th to prevent the long wire delay and routing congestion. Next, a scan flip-flop can be shared without losing fault coverage if the fan-in or fan-out cones are not overlapped. Also, our method also tries to share scan flip-flops with overlapped fan-in or fan-out cones (with the help of a commercial automatic test pattern generation (ATPG) tool) if (1) the decreased fault coverage  $fault\_coverage(n_1, n_2)$  is lower than an user defined threshold  $cov_th$  and (2) the increased test pattern count  $\#test\_patterns(n_1, n_2)$  is less than an user defined threshold p\_th. The proposed method gives a trade-off between area overhead, fault coverage  $(cov_th)$ , and the number of test patterns  $(p_th)$ . After the graph construction, we then apply the heuristic clique-partitioning algorithm to solve the problem.

## C. The Heuristic to Solve Clique-partitioning Algorithm

In this section, we present our heuristic clique-partitioning algorithm. Given a graph with nodes and edges, we want to find a set of cliques in which each node belongs to a clique such that the total number of cliques is minimized. At the beginning, all nodes are in separated cliques with size 1, i.e. all TSVs use additional wrapper cells (i.e. the upper bound initial solution). Then, the capacity load of two connecting nodes *cap* with minimal non-zero edge degrees,  $n_1$ and  $n_2$ , are checked. If the capacity load is less than the predefined value *cap\_th*, then  $n_1$  and  $n_2$  are merged into the same clique, and a new node n' is added. Also, new edges are added to n' for all common neighbors of  $n_1$  and  $n_2$ . Then, the capacity load information is updated correspondingly. Finally, the nodes  $n_1$  and  $n_2$  are deleted. Otherwise, the edge between  $n_1$  and  $n_2$  is deleted. The above steps are iterated until all nodes have zero degree, i.e. all edges are removed. TABLE I

THE FAULT COVERAGE COMPARISONS ON STARTING FROM inbound OR outbound TSVS USING THE METHOD BY M. AGRAWAL ET AL. [4].

		#inbound	#outbound	Start from in	bound TSVs	Start from outbound TSVs		
		TSV TSV		Fault coverage	#wrapper cells	Fault coverage	#wrapper cells	
	Die0	22	28	99.14%	26	99.34%	23	
b12	Die1	41	41	98.80%	23	98.90%	23	
012	Die2	23	42	99.11%	0	99.43%	0	
	Die3	25	5	99.93%	7	99.89%	9	

# Algorithm 1 Graph construction

	-
1:	/* Node construction */
2:	for each $n \in scan_flip flops$ do
3:	Add a new node n
4:	end for
5:	for each $n \in inbound\_TSVs$ do
6:	if $capacity\_load(n) < cap\_th$ then
7:	Add a new node n
8:	end if
9:	end for
10:	for each $n \in outbound\_TSVs$ do
11:	if $slack(n) > s_th$ then
12:	Add a new node n
13:	end if
14:	end for
15:	
16:	/* Edge construction */
17:	for each node pair $(n_1, n_2)$ where $n_1 \in TSVs$ or $n_2 \in TSVs$ do
18:	if $distance(n_1, n_2) < d_{th}$ then
19:	if fan-in/fan-out cones of $n_1, n_2$ are not overlapped then
20:	Add a new edge $(n_1, n_2)$
21:	else if $fault\_coverage(n_1, n_2) > cov\_th$
22:	and $\#test_patterns(n_1, n_2) < p_th$ then
23:	Add a new edge $(n_1, n_2)$
24:	end if
25:	end if
26:	end for

Algorithm 2 The heuristic clique-partitioning algorithm
1: Let all nodes to be in separated cliques with size 1
2: while $\exists$ non-zero degree node do
3: Let $n_1$ be the smallest non-zero degree node
4: Let $n_2$ be a neighbor node of $n_1$ with smallest degree
5: if $cap + 1 < cap_th$ then
6: Merge $n_1$ and $n_2$ into the same clique
7: Add a new node $n'$
8: for each common neighbor $n_c$ of $n_1$ and $n_2$ do
9: Add new edge $(n', n_c)$
10: end for
<ol> <li>Update capacity load information</li> </ol>
12: Delete nodes $n_1$ and $n_2$
13: else
14: Delete edge $(n_1, n_2)$
15: end if
16: end while

# V. EXPERIMENT RESULTS

Our method is evaluated using C++ programming language on a 3.30GHz Linux system with 128GB memory. Six benchmark circuits from ITC'99 benchmark [5] are selected, namely *b*11, *b*12, *b*18, *b*20, *b*21, and *b*22. These benchmark circuits were synthesized to gate level netlists by Design Compiler [12] with 45 *nm* technology library. Then, a 3D physical design flow package 3D-Craft [14] is adopted to extract the physical information of scan flip-flops and TSVs. Finally, static timing analysis by a commercial tool PrimeTime [12] is performed to collect the timing information. Table II provides the information of the benchmark circuits including the number of scan flip-flops, gates, TSVs, *inbound* TSVs, and *outbound* TSVs.

 TABLE II

 The characteristics of ITC'99 benchmark circuits [5].

		#scan flip-flops	#gates	#TSVs	#inbound TSVs	#outbound TSVs
	Die0	14	120	30	14	16
1.1.1	Die1	15	234	70	27	43
DII	Die2	3	229	76	38	38
	Die3	9	148	34	23	11
	Die0	7	304	50	23	27
L12	Die1	18	397	82	41	41
012	Die2	45	344	65	23	42
	Die3	51	317	30	25	5
	Die0	515	22934	1505	772	733
L10	Die1	1033	26698	3436	1561	1875
018	Die2	833	23575	3529	1732	1797
	Die3	641	20825	1581	810	771
	Die0	180	6937	614	251	363
h20	Die1	49	8603	1500	720	780
620	Die2	118	8101	1518	740	778
	Die3	83	7325	643	408	235
	Die0	196	6200	592	264	328
h21	Die1	113	9172	1611	836	775
021	Die2	69	9093	1732	837	895
	Die3	52	6402	711	368	343
	Die0	225	9427	982	499	483
h22	Die1	201	12726	2071	1006	1065
022	Die2	181	13075	2095	1031	1064
	Die3	6	11358	992	511	481
Ave	erage	194.04	8522.67	1064.54	523.33	541.21

### A. Reductions on Additional Wrapper Cells

To reduce the area overhead, our method tries to reuse more scan flip-flops and insert less additional wrapper cells. However, different timing constraints may affect how scan flip-flops can be reused as wrapper cells. In our first experiment, two scenarios of different timing constraints are considered. The first scenario is the extremely loose timing constraint, i.e. no timing constraint at all, as an areaoptimized scenario. On the other hand, the timing constraint of the second scenario is tuned to a very tight value as a performanceoptimized scenario. Also, the Agrawal's method in the area-optimized scenario is defined as baseline for comparisons. In the area-optimized scenario, our method on average reuses 3.48% more scan flip-flops and inserts 6.01% less additional wrapper cells compared to the Agrawal's method (baseline). In the performance-optimized scenario, the Agrawal's method on average reuses 6.67% less scan flip-flops and inserts 7.81% more wrapper cells compared to the baseline. Even Agrawal's method tries to use more hardware resources to meet the rigid timing requirements, 20 out of 24 circuits still violate the timing requirements. In contrast, our method manages to reuse 0.98% more scan flip-flops and insert 0.92% less additional wrapper cells on average compared to the baseline without any timing violation across all benchmarks.

 TABLE III

 The number of reused scan flip-flops and wrapper cells comparisons under optimizing for area (no timing) and performance (tight timing).

	Agrawal's (no timing) [4]		no timing) [4]	Our (no timing)		Agrawal's (tight timing) [4]			Our (tight timing)		
		#reused scan	#additional	#reused scan	#additional	#reused scan	#additional	Timing	#reused scan	#additional	Timing
		flip-flops	wrapper cells	flip-flops	wrapper cells	flip-flops	wrapper cells	violation	flip-flops	wrapper cells	violation
	Die0	7	2	8	1	6	3	X	8	2	
b11	Die1	16	1	17	0	16	2		17	0	
011	Die2	14	0	14	0	13	1	X	14	0	
	Die3	11	0	11	0	10	2		10	1	
	Die0	16	3	17	2	15	4		16	3	
b12	Die1	31	0	31	0	30	0	X	31	0	
012	Die2	24	4	26	1	24	5	X	24	2	
	Die3	4	1	4	1	3	2	X	3	2	
	Die0	275	125	275	125	265	140	X	262	142	
b18	Die1	801	146	835	119	782	159	X	825	125	
010	Die2	709	4	712	0	702	8	X	708	5	
	Die3	328	64	330	61	320	77	X	326	70	
	Die0	115	130	128	110	110	139	X	122	5 70 112 131	
b20	Die1	82	139	92	135	75	141	X	90	131	
020	Die2	115	131	120	135	100	156	X	118	142	
	Die3	110	5	110	5	108	7		106	9	
	Die0	159	75	165	69	151	83	X	160	75	
b21	Die1	144	196	142	200	138	210	X	140	203	
021	Die2	104	160	105	158	104	160	X	85	180	
	Die3	97	60	97	60	96	61	X	96	61	
	Die0	168	170	166	175	166	172	X	164	179	
h22	Die1	159	231	205	190	14	252	X	200	194	
022	Die2	172	175	182	158	164	184	X	175	161	
	Die3	100	125	100	125	98	131	X	98	130	
Ave	erage	156.71	81.13	162.17	76.25	146.25	87.46	20/24	158.25	80.38	0/24
(	%)	(100.00%)	(100.00%)	(103.48%)	(93.99%)	(93.33%)	(107.81%)	20/24	(100.98%)	(99.08%)	0/24

## B. The Fault Coverage Analysis

The main purpose for inserting additional wrapper cells is to provide the testability of TSVs at pre-bond testing. Our method aggressively tries to share the scan flip-flops with overlapped fanin/fan-out cones and sets testability constraints, i.e. the decreased fault coverage threshold  $cov_th = 0.5\%$  and the increased test pattern count threshold  $p_th = 10$ . In the area-optimized scenario, our experimental results show a similar fault coverage and test pattern count compared to the Agrawal's methods, which is omitted to save space. Table IV shows the fault coverage and the number of test patterns under the performance-optimized scenario (fault coverage, #test patterns). The results show that our method on average achieves the same fault coverage and 4.71 and 2.5 less test patterns on stuck-at and transition faults, respectively, compared to the Agrawal's method.

# C. Solution Space Expansion Analysis with Overlapped Fan-in/fanout Cones

Next, we demonstrate the performance effect of sharing scan flip-flops with overlapped fan-in/fan-out cones. In this experiment, the performance-optimized scenario is considered. Figure 7 shows that, by allowing overlapped fan-in/fan-out cones, the number of edges of constructed graph is increased by 2.83% on average, which means that solution space is expended. As the result, our method on average reuses 0.90% more scan flip-flops and achieves 2.01% less additional wrapper cell insertion compared to the method with no overlapped fan-in/fan-out cones as shown in Table V. Table V also illustrates the fault coverage and test pattern count of stuck-at fault and transition fault (fault coverage, #test patterns). The results show that our method provides a similar testability compared to the method with no overlapped fan-in/fan-out cones, in which our method on average

TABLE IV THE FAULT COVERAGE AND PATTERN COUNT COMPARISONS ON STUCK-AT FAULT AND TRANSITION FAULT.

		Agraw	/al's [4]	Our			
		Stuck-at fault	Transition fault	Stuck-at fault	Transition fault		
	Die0	(99.52%, 58)	(99.32%, 108)	(99.52%, 61)	(99.34%, 92)		
1.1.1	Die1	(99.60%, 78)	(99.51%, 173)	(99.70%, 81)	(99.51%, 178)		
011	Die2	(98.40%, 81)	(98.10%, 179)	(98.20%, 78)	(98.10%, 191)		
	Die3	(99.83%, 82)	(99.61%, 155)	(99.84%, 80)	(99.61%, 145)		
	Die0	(99.38%, 102)	(99.11%, 228)	(99.12%, 98)	(99.10%, 225)		
h12	Die1	(99.65%, 161)	(99.55%, 351)	(99.65%, 165)	(99.55%, 345)		
012	Die2	(99.43%, 131)	(99.21%, 257)	(99.42%, 121)	(99.22%, 260)		
	Die3	(99.93%, 96)	(99.88%, 205)	(99.93%, 95)	(99.89%, 210)		
	Die0	(99.71%, 1448)	(99.66%, 3208)	(99.73%, 1455)	(99.68%, 3182)		
b19	Die1	(99.85%, 1901)	(99.81%, 3991)	(99.85%, 1981)	(99.81%, 4102)		
010	Die2	(99.73%, 2011)	(99.90%, 4152)	(99.73%, 1852)	(99.91%, 4181)		
	Die3	(99.79%, 1476)	(99.52%, 2974)	(99.79%, 1452)	(99.52%, 2901)		
	Die0	(99.42%, 818)	(99.16%, 1399)	(99.42%, 821)	(99.16%, 1442)		
b20	Die1	(99.21%, 820)	(99.72%, 1620)	(99.21%, 814)	(99.71%, 1631)		
020	Die2	(99.70%, 1022)	(99.11%, 2008)	(99.77%, 981)	(99.10%, 1982)		
	Die3	(99.86%, 1368)	(98.90%, 2348)	(99.88%, 1388)	(98.79%, 2401)		
	Die0	(99.55%, 829)	(99.56%, 1381)	(99.55%, 877)	(99.56%, 1341)		
h21	Die1	(99.87%, 961)	(98.70%, 1778)	(99.99%, 937)	(98.69%, 1781)		
021	Die2	(99.71%, 985)	(99.78%, 1881)	(99.71%, 1071)	(99.91%, 1852)		
	Die3	(99.86%, 801)	(98.72%, 1411)	(99.90%, 784)	(98.72%, 1381)		
	Die0	(99.78%, 779)	(99.11%, 1428)	(99.80%, 758)	(99.11%, 1377)		
h22	Die1	(99.88%, 1878)	(99.03%, 3308)	(99.96%, 1870)	(99.02%, 3218)		
022	Die2	(99.65%, 1611)	(98.92%, 3368)	(99.74%, 1597)	(98.99%, 3407)		
	Die3	(99.99%, 764)	(99.05%, 1462)	(99.99%, 731)	(99.05%, 1488)		
Ave	erage	(99.64%, 844.21)	(99.29%, 1640.54)	(99.64%, 839.50)	(99.29%, 1638.04)		

has only 0.23% and 0.15% fault coverage lost but achieves 8.92 and 10 less test patterns on stuck-at and transition faults, respectively.

# VI. CONCLUSIONS

Due to un-controllable and un-observable characteristics of TSVs, the 3D-ICs lack of testability at pre-bond testing. By inserting the wrapper cells, the TSVs can regain the controllability and observability. Since the insertion of the wrapper cells may lead to significant area overhead, minimizing the number of wrapper cells is a prime consideration. We have studied the problem of minimizing the wrapper cells count in 3D-ICs using a formal approach based on

TABLE V The fault coverage and pattern count comparisons on stuck-at fault and transition fault with/without allowing overlapped fan-in/fan-out cones.

			No overlapp	ed fan-in/fan-out cones	3	Allow overlapped fan-in/fan-out cones				
		#reused scan	#additional	Stuck at fault	Transition fault	#reused scan	#additional	Stuck at fault	Transition fault	
		flip-flops	wrapper cells	Stuck-at Tault		flip-flops	wrapper cells	Stuck-at fault		
	Die0	122	112	(99.42%, 821)	(99.11%, 1441)	124	110	(99.01%, 815)	(99.16%, 1442)	
b20	Die1	90	131	(99.21%, 814)	(99.71%, 1631)	91	129	(99.11%, 813)	(99.52%, 1624)	
020	Die2	118	142	(99.77%, 981)	(99.10%, 1982)	120	138	(99.51%, 975)	(98.99%, 1977)	
	Die3	106	9	(99.88%, 1388)	(98.79%, 2401)	110	4	(99.88%, 1388)	(98.79%, 2401)	
	Die0	160	83	(99.55%, 877)	(99.56%, 1341)	162	80	(99.37%, 866)	(99.46%, 1331)	
b21	Die1	140	203	(99.99%, 937)	(98.69%, 1781)	143	198	(99.79%, 924)	(98.51%, 1766)	
021	Die2	85	180	(99.71%, 1071)	(99.91%, 1852)	85	180	(99.58%, 1055)	(99.78%, 1844)	
	Die3	96	61	(99.90%, 784)	(98.72%, 1381)	97	59	(99.78%, 780)	(98.69%, 1373)	
	Die0	164	179	(99.80%, 758)	(99.11%, 1377)	164	178	(99.62%, 744)	(98.87%, 1371)	
b22	Die1	200	194	(99.96%, 1870)	(99.02%, 3218)	201	194	(99.51%, 1861)	(98.72%, 3211)	
022	Die2	175	161	(99.74%, 1597)	(98.99%, 3407)	171	159	(99.31%, 1581)	(98.69%, 3379)	
	Die3	98	130	(99.99%, 731)	(99.05%, 1488)	100	124	(99.59%, 720)	(98.85%, 1461)	
Average (%)		129.50 (100.00%)	132.08 (100.00%)	(99.74%, 1052.42)	(99.15%, 1941.67)	130.67 (100.90%)	129.42 (97.98%)	(99.51%, 1043.50)	(99.00%, 1931.67)	



Fig. 7. The solution space is expended due to allowing overlapped fan-in/fanout cones.

graph theory. By consulting the theorem proposed by M. Agrawal et al. [4], we adopt the minimal clique-partitioning problem to address the minimizing issue. Our method also considered the timing impact on reused scan flip-flops by utilizing capacity load and wire delay information. Moreover, for the purpose of increasing the re-usability of scan flip-flops, we adopt the commercial ATPG tool to analyze the testability impact after reusing flip-flops. The experimental results have demonstrated that our method can truly solve the testability issues at pre-bond testing with the optimized number of additional wrapper cells. Moreover, with the accurate timing model, experimental results also show that no timing violation occurred in our method. Also, the testability parameters are provided by our method which giving a trade-off between area overhead and testability. The experimental results show that our method provides a competitive testability with inserting 0.92%-6.01% less wrapper cells compared to the Agrawal's method [4].

#### References

- [1] IEEE Standard for Test Access Port and Boundary-Scan Architecture, May 2013.
- [2] IEEE Standard Testability Method for Embedded Core-based Integrated Circuits, Aug 2005.
- [3] J. Li and D. Xiang, "Dft optimization for pre-bond testing of 3d-sics containing tsvs," in 2010 IEEE International Conference on Computer Design, pp. 474–479, Oct 2010.
- [4] M. Agrawal, K. Chakrabarty, and R. Widialaksono, "Reuse-based optimization for prebond and post-bond testing of 3-d-stacked ics," *IEEE*

Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 34, pp. 122–135, Jan 2015.

- S. Davidson, "Itc'99 benchmark circuits preliminary results," in *International Test Conference 1999. Proceedings (IEEE Cat. No.99CH37034)*, pp. 1125–1125, Sep. 1999.
- [6] G. E. Moore, "Cramming more components onto integrated circuits, reprinted from electronics, volume 38, number 8, april 19, 1965, pp.114 ff.," *IEEE Solid-State Circuits Society Newsletter*, vol. 11, pp. 33–35, Sep. 2006.
- [7] J. Lu, "3-d hyperintegration and packaging technologies for micro-nano systems," *Proceedings of the IEEE*, vol. 97, pp. 18–30, Jan 2009.
- [8] J. Van Olmen, A. Mercha, G. Katti, C. Huyghebaert, J. Van Aelst, E. Seppala, Z. Chao, S. Armini, J. Vaes, R. C. Teixeira, M. Van Cauwenberghe, P. Verdonck, K. Verhemeldonck, A. Jourdain, W. Ruythooren, M. de Potter de ten Brocck, A. Opdebeeck, T. Chiarella, B. Parvais, I. Debusschere, T. Y. Hoffmann, B. De Wachter, W. Dehaene, M. Stucchi, M. Rakowski, P. Soussan, R. Cartuyvels, E. Beyne, S. Biesemans, and B. Swinnen, "3d stacked ic demonstration using a through silicon via first approach," in 2008 IEEE International Electron Devices Meeting, pp. 1–4, Dec 2008.
- [9] C. C. Liu, I. Ganusov, and M. B. and, "Bridging the processor-memory performance gap with 3d ic technology," *IEEE Design Test of Computers*, vol. 22, pp. 556–564, Nov 2005.
- [10] B. Black, D. W. Nelson, C. Webb, and N. Samra, "3d processing technology and its impact on ia32 microprocessors," in *IEEE International Conference on Computer Design: VLSI in Computers and Processors*, 2004. ICCD 2004. Proceedings., pp. 316–318, Oct 2004.
- [11] N. Watanabe, M. Suzuki, M. Aoyagi, M. Eto, and K. Kawano, "Development of a chip prober for pre-bond testing of a 3d-ic," in 2013 3rd IEEE CPMT Symposium Japan, pp. 1–4, Nov 2013.
- [12] H. Bhatnagar, Advanced ASIC Chip Synthesis: Using Synopsys Design Compiler Physical Compiler and PrimeTime. Norwell, MA, USA: Kluwer Academic Publishers, 2nd ed., 2002.
- [13] E. J. Marinissen and Y. Zorian, "Testing 3d chips containing throughsilicon vias," in 2009 International Test Conference, pp. 1–11, Nov 2009.
- [14] J. Cong and G. Luo, "A 3d physical design flow based on open access," in 2009 International Conference on Communications, Circuits and Systems, pp. 1103–1107, July 2009.