At-speed testing of inter-die connections of 3D-SICs in the presence of shore logic

At-Speed Testing of Inter-Die Connections of 3D-SICs in the Presence of Shore Logic*

Konstantin Shibin[1,2,3] Vivek Chickermane[1] Brion Keller[1] Christos Papameletis[1] Erik Jan Marinissen[2]

[1] Cadence Design Systems
1701 North Street, Building 257-1
Endicott, NY 13760
United States of America
{vivekc,kellerbl,christos}@cadence.com

[2] IMEC
Kapeldreef 75
B-3001 Leuven
Belgium
erik.jan.marinissen@imec.be

[3] Tallinn University of Technology
Department of Computer Engineering
Akadeemia tee 15a, 12618, Tallinn
Estonia
konstantin.shibin@ati.ttu.ee

Abstract

Inter-die connections in 2.5D- and 3D-stacked ICs require at-speed testing as their dynamic performance is crucial to the performance of the stack as a whole. In order to test at mission-mode speed and benefit from the already existing clock distribution network, our at-speed test approach for inter-die connections targets the entire register-to-register path that includes the interconnect. This forces the launching and capturing wrapper cells to be shared with functional flip-flops. In some designs, this unavoidably leads to some 'shore logic': a, typically small, amount of combinational logic outside the die's wrapper boundary register. This paper describes how we have adapted a previously developed 3D-DfT architecture and corresponding EDA tool flows to support at-speed interconnect testing, also in the presence of such 'shore logic'. The adaptations affect the DfT insertion of wrapper cells, the boundary model extraction, and the interconnect test pattern generation.

1 Introduction

Three-dimensional stacking is an approach for vertical integration of multiple dies into a single chip product. There are two main flavors of this approach: stacking active dies side-by-side on a passive interposer base die (a.k.a. 2.5D) and stacking active-on-active dies (a.k.a. true 3D). 3D stacking is a promising way to extend the momentum of Moore's Law, enabling high-bandwidth, low-latency connections between homogeneous or heterogeneous dies: memory, logic, analog, etc. 3D stacking evolves around the usage of many cheap, fast, low-power inter-die connections. These 3D interconnects consist of (1) a micro-bump pair, (2) often a through-silicon via (TSV) that provides an electrical connection between the front- and back-side of a silicon substrate [1–3], and (3) some intra-die wiring.

IMEC and Cadence have jointly developed a 3D-DfT architecture based on DfT die wrappers, that serves both 2.5D- and 3D-SICs [4–6]. Originally targeting stacks of monolithic logic-only dies [7], over time this architecture has been extended to include (1) memory-on-logic stacks [8, 9], (2) core-based SOCs [10, 11], and (3) multi-tower stacks [10–12]. We have implemented a full automation flow with Tcl scripts on top of Cadence' RTL Compiler (RC) and Encounter Test (ET) [11, 13].

Our test covering the inter-die connections containing TSVs and micro-bumps, until recently focused only on static defects caused by hard opens and shorts. However, these interconnects might also be affected by resistive ('weak') short and open defects; especially the latter typically manifest themselves as delay faults. Hence we have extended our 3D-DfT architecture and associated EDA tool flow to support at-speed transition-based delay-fault testing of these interconnects.

Instead of testing the inter-die connection as a stand-alone object, our at-speed test approach targets entire register-to-register paths that include the inter-die connections, in order to test at mission-mode speed and benefit from the already existing clock distribution network. This requires the wrapper cells of the die's 3D-DfT wrapper to be shared with functional flip-flops; even if these flip-flops reside not at the die boundary, but a little 'inland', i.e., separated from the die boundary by combinational ('shore') logic. Also, output wrapper cells need to be extended with transition-launch capability. Fortunately, RC's powerful functions for embedded-core wrapping provide us with the means to automate this.

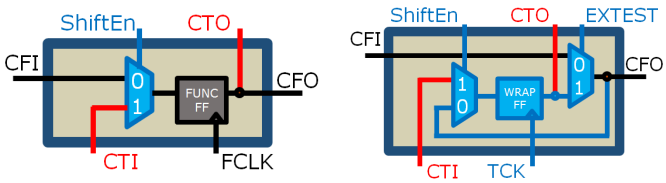
In our tool flow, ET generates a 'boundary model' of the wrapped die: a boundary DfT netlist which black-boxes the die's design IP, and is used by the stack integrator for stack interconnect test generation and test migration [14] from die to stack level. Inland wrapping requires this boundary model to be extended to include the shore logic. ET supports two ways of generating interconnect tests: (1) conventional Logic-ATPG, covering shore logic and interconnects, and (2) dedicated Interconnect-ATPG which only targets the interconnects, but is able to propagate through the shore logic.

The rest of this paper is organized as follows. Section 2 describes the related prior work, Section 3 and 4 present the core of our method with the required modifications to the EDA tool flow. Section 5 defines the required clocks for at-speed testing, Section 6 describes the chosen at-speed transition launch method and Section 7 describes the experimental results. Section 8 concludes this paper.

* Part of this work is supported by IT Academy of Estonia.

2 Related Prior Work

The main component of the IMEC/Cadence 3D-DfT architecture is a die-level DfT wrapper, which is based on a 3D-enhanced version of IEEE Std 1500 [4–7]. The 3D wrapper provides facilities to transport test stimuli from stack input pins up into the stack to the die-under-test (DUT), while the DUT's test responses are transported down through the stack to stack output pins. The architecture supports a modular test approach, in which dies and their inter-die interconnects can be tested as stand-alone units (as INTEST resp. EXTEST). This allows for (1) targeted test pattern generation and reuse, thereby enabling tractable ATPG compute times (even for stacks consisting of many large dies), (2) reduction of the test data volume [15], (3) protection of the die's design IP, and (4) first-order fault diagnosis and yield attribution. Modular testing is enabled by a wrapper boundary register (WBR); i.e., a concatenated series of wrapper cells which provide scan-based controllability and observability at all functional die I/Os. Typically, most die I/Os are equipped with a functional register, which can be reused and turned into a *shared* wrapper cell (see Figure 1(a)) clocked on the functional clock `FCLK`. For shared wrapper cells, there is zero area and performance impact due to the WBR. In case a die I/O is not registered, it can be equipped with a *dedicated* wrapper cell (see Figure 1(b)) clocked on the test clock `TCK`; this brings with it a small increase of silicon area, performance impact (due to the propagation delay of the multiplexer), and test application time (the extra flip-flop might increase the longest scan chain length).



(a) Shared wrapper cell.

(b) Dedicated wrapper cell.

Figure 1: Shared (a) and dedicated (b) wrapper cells.

As the WBR resides at the die boundary, the EXTEST circuitry in between the WBRs exclusively consists of components which for ATPG-purposes are modeled as wires, i.e., point-to-point connections without combinational nor sequential logic: micro-bump pairs, TSVs, and/or interposer wires. This means that the EXTEST test pattern generation can be handled by dedicated Interconnect-ATPG based on the effective and efficient True/Complement Counting Sequence [16]. This algorithm writes out a test consisting of only $2 \cdot \log_2(n)$ test patterns (for n simultaneously tested interconnects) that covers *all* hard opens and shorts.

We have implemented two automation flows with Tcl scripts on top of RC and ET [11, 13]: one for die makers and another one for stack makers. These flows consist of the following main steps.

- Die maker flow:
  1. RC: Die-level DfT insertion.
  2. ET: Die-level INTEST test pattern generation with Logic-ATPG.
  3. ET: Die-level boundary model extraction: a boundary DfT netlist which black-boxes the die's internal design IP.

- Stack maker flow:
  Manual: Creation of a stack netlist containing the inter-die connections and the boundary models of the stacked dies.
  1. ET: For each die: test migration of the die-level INTEST through the stack netlist to the stack-level I/Os.
  2. ET: EXTEST test pattern generation with Interconnect-ATPG at the stack netlist.

Prior work on at-speed 3D interconnect testing includes approaches based on the interconnect delay measurement using ring oscillators with input sensitivity [17, 18] and output thresholding [19] analysis. Another paper [20] uses transition-time monitors to detect slow signal transitions at the receiver side of the interconnect. While these approaches provide good diagnostics capabilities, they need special synchronized modifications in all involved dies, put an extra load on the functional signals, and require dedicated hardware (ring oscillator counters, delay lines). Another paper [21] proposes to modify the dedicated output wrapper cells to include the capability of at-speed transition launch and to use the functional flip-flops to capture the transition. Aforementioned methods require considerable amount of extra hardware and concentrate on testing the interconnect in isolation. In addition, these papers do not address the at-speed testing of the shore logic. Our approach tries to overcome these shortcomings by using functional flip-flops for 3D interconnect at-speed testing for both transition launch and capture and taking into account the shore logic.

3 From Boundary to Shared Wrappers

For static interconnect testing, we assumed wrapper cells at the die boundary; shared wrapper cells if possible, and dedicated wrapper cells if necessary. This implied that interconnects were 'wires' only, and hence that test pattern generation could be handled by dedicated Interconnect-ATPG alone. For dynamic at-speed interconnect testing through full register-to-register paths, these assumptions are no longer attractive, as is explained in this section.

Consider the generic inter-die interconnect as depicted in Figure 2. Interconnect 1 shows the functional design, prior to any DfT insertion. This interconnect is not 'registered', i.e., there is combinational logic between the driving flip-flop and the interconnect and/or between the interconnect and the receiving flip-flop. The functional designer is responsible for synchronizing `FCLK1` and `FCLK2` in Die 1 and Die 2 respectively. Let us assume that the mission-mode frequency of this design is 1 GHz, corresponding to a clock period of 1 ns. Let us further assume that the 1 ns clock period is divided into 0.2 ns for the actual interconnect and twice 0.4 ns for the combinational logic on either side of the interconnect.

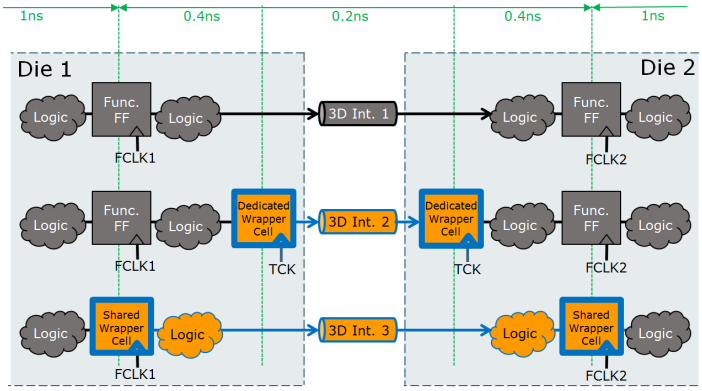


Figure 2: Two options to test a 3D interconnect (1) at-speed: test the interconnect itself through boundary wrapping (2) or test the entire register-to-register path that includes the interconnect through shared wrapper cells (3).

Interconnect 2 in Figure 2 depicts what happens if we insert dedicated wrapper cells around our interconnect. These wrapper cells run on the `TCK` clock. Given the 0.2 ns nominal propagation delay for the interconnect, an at-speed interconnect test needs to run at a 5 GHz `TCK` clock. In many designs, `TCK` is a rather slow clock signal, but at 5 GHz, `TCK` needs to be routed as a high-speed clock signal; due to the design effort involved, this is unattractive. Also unattractive is that the at-speed testing of the entire design requires multiple frequencies: 1 GHz for the internal logic, 2.5 GHz for the logic at the die boundary, and 5 GHz for the inter-die connections. This hardship has its root cause in the fact that the 1 GHz full register-to-register path now is partitioned into multiple partial paths each of which have to be tested at their own, elevated frequency.

Way more attractive is to test the interconnect as part of the functional register-to-register path at its regular mission-mode frequency (1 GHz in our example above), shown in Figure 2 as Interconnect 3. This means that we exclusively work with shared wrapper cells that operate on functional clocks and dedicated wrapper cells are not allowed. In that way, we avoid hard-to-meet timing constraints on `TCK`, and rely on and reuse the existing clock distribution network (`FCLK1` and `FCLK2`). As a side benefit, shared wrapper cells have less impact on silicon area and performance than dedicated wrapper cells. The price we pay for our approach is that we have to be able to cope with 'shore logic', i.e., some combinational logic that now becomes part of the EXTEST circuitry. Most interconnects are registered at the die boundary (as this provides a cleaner timing interface); for those interconnects the shore logic is non-existent and boundary wrapping actually equals shared wrapping. But in order to be able to provide a comprehensive solution for all functional designs, our tool flow must be able to handle the situations in which (some) shore logic does exist.

4 Tool Flow Adaptations for Shore Logic

This section addresses the consequences and necessary adaptations to our EDA tool flows for die and stack makers due to the choice to move to shared-only wrapper cells and in the possible presence of shore logic.

4.1 DfT Insertion

With conventional boundary wrapping, there is always exactly one wrapper cell per die I/O. In case the I/O is registered, the boundary flip-flop is marked as a shared wrapper cell and no DfT on top of the regular scan design needs to be inserted. If the I/O in question is not registered, a dedicated wrapper cell is inserted on the functional path to/from the I/O. Afterwards, all shared and dedicated wrapper cells need to be stitched up into one or more scan chains that are enabled in both EXTEST and INTEST modes.

In the case of shared wrapping, nothing changes for the large group of registered I/Os; their scan flip-flops are still marked as shared wrapper cells. New is the situation for non-registered I/Os. The DfT insertion tool needs to identify the (inland) scan flip-flops on the driving (for die outputs) or receiving (for die inputs) side of the logic and mark them as shared wrapper cells. The shore logic typically constitutes a fan-in (resp. fan-out) cone with multiple inputs (resp. outputs) (see Figure 3); all these scan flip-flops need to be marked as shared wrapper cells.

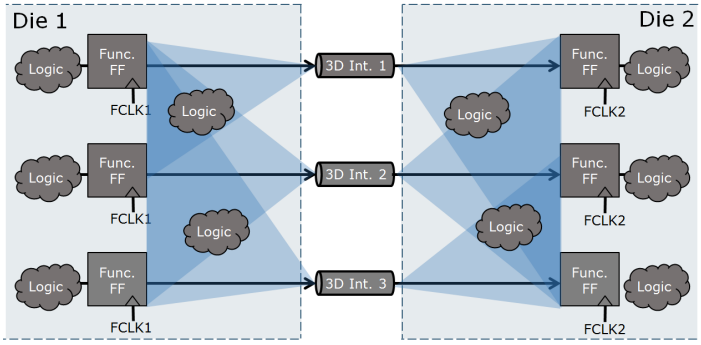


Figure 3: Fan-in/out cones of the 3D interconnect I/Os with shore logic.

A large fan-in/out cone has several negative consequences.

1. The boundary model includes the shore logic and hence grows in size.
2. The design IP of the shore logic is not protected by the black-box feature of the boundary model and will be released to the stack maker.
3. More scan flip-flops will be marked as shared wrapper cells, which will contribute to a growth in length of the EXTEST scan chain(s).
4. Depending on the exact configuration of the logic cone, test pattern generation for the interconnect wires and the shore logic itself may become harder for the ATPG due to higher risk of signal value conflicts in the shore logic. This might result in a drop in fault coverage and/or an increase in the number of test patterns due to less interconnects being tested simultaneously.

In order to limit the size of the fan-in/out cone, RC supports a user-defined integer threshold for the number of shared wrapper cells associated to a single die I/O. If the shore logic fans in/out to more flip-flops than this threshold, the tool issues a warning

and inserts a dedicated wrapper cell. The user should then either increase the threshold value or modify the IC design to avoid insertion of a dedicated wrapper cell.

4.2 Boundary Model Extraction

With conventional boundary wrapping, the extracted boundary model exclusively consists of interconnect 'wires' and EXTEST scan chains.

In case of shared wrapping, the boundary model extraction software also needs to include the shore logic. ET's boundary model extraction engine is capable of doing this. The boundary model format is essentially a gate-level netlist and hence has no problem to include the combinational gates of the shore logic.

The boundary model will grow in size with the inclusion of the shore logic. As the boundary model is meant to be shared between die maker and stack maker, the design IP protection which the boundary model provides to the black-boxed INTEST logic does not extend to the shore logic. As the shore logic is typically very small compared to the overall die size, we consider this a surmountable drawback.

4.3 ATPG for EXTEST

With conventional boundary wrapping, the EXTEST circuitry exclusively consists of 'wires' and its test pattern generation can be handled by dedicated Interconnect-ATPG algorithms based on counting sequences.

In case of shared wrapping, the test pattern has to cope with the likely presence of shore logic as part of the EXTEST circuitry and cover both the combinational shore logic as well as the wiring interconnects. This calls for an approach in which two separate ATPG engines within ET are invoked.

1. *Logic-ATPG*. It targets the shore logic between the two stacked dies. Inter-die connections are considered as regular wires between the combinational gates of the shore logic. Logic-ATPG can be used in combination with all common fault models, including stuck-at and transition fault models.

2. *Interconnect-ATPG*. It explicitly targets the inter-die connections between the two stacked dies. The combinational shore logic is *not* explicitly targeted, but ET's Interconnect-ATPG is capable of propagating stimuli and responses through this logic, thereby generating some incidental fault coverage. Interconnect-ATPG creates patterns for static and dynamic Stuck Driver Test (SDT) at the interconnect drivers and receivers, as well as Shorted Net Test (SNT) to cover possible shorts between the inter-die interconnects.

5 At-Speed Clocking During Test

At-speed testing in our method is performed by applying two at-speed clock cycles: for launch and capture. In order to generate exactly two cycles, we rely on the same mechanisms that are used for at-speed INTEST. It is usually implemented using a pulse generator which creates the required clock pulse pattern out of a continuous series of clock pulses generated by an on-chip PLL. In case of the Cadence tool flow, this pulse generator is the On-Product Clock Generator (OPCG) module [22]. Respective waveforms are shown in Figure 4. The `PLLCLK` signal corresponds to the output of the PLL. The `ShiftEn` signal switches the clock between the slow scan and at-speed test modes. The `TRIGGER` signal instructs the pulse generator module to issue the required pattern which appears on `FCLK`. The corresponding clock control and distribution scheme is shown in Figure 5. OPCG supports inter-clock-domain test; modules in several dies can be inserted and connected such that synchronized inter-die dynamic test can be performed.

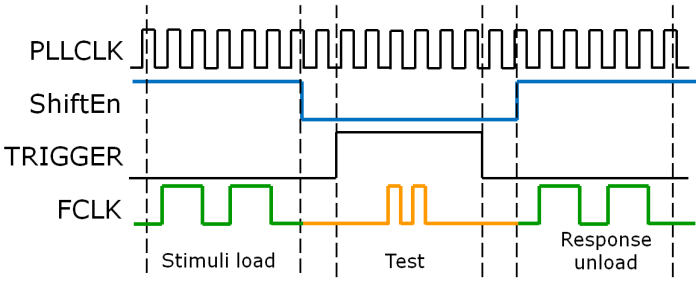


Figure 4: Waveforms for at-speed test.

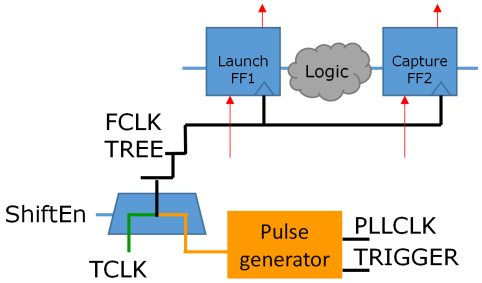


Figure 5: Clock control and distribution.

6 At-Speed Transition Launch/Capture

In our work, we use the register-to-register transition launch and capture method to test the 3D interconnect for at-speed timing specifications. This method is widely used in the industry for the same purpose in the internal circuitry of the die. The idea of the method is to generate a data transition from one flip-flop, launch it across the path-under-test, and after a functional clock period capture the resulting transition effect in a flip-flop. For this, two at-speed clock cycles are used: launch and capture. In order to generate a transition, first the initialization value is loaded (using scan and a slow, not-at-speed clock) into the launching flip-flop.

Then, at the first at-speed test clock cycle (launch), the initialization value is changed to its opposite (transition) value. At the second clock cycle, the resulting propagated transition effect is captured by the receiving flip-flop. In case of a delay fault on the path-under-test, the resulting transition does not reach the capture flip-flop in time, and hence the delay fault is detected.

The industrial practice for at-speed test of internal die circuits is to generate the transition by making the launch flip-flop to capture the opposite value generated by the logic in the fan-in cone of its functional input (`CFI`, see Figure 6). This method is called Launch-on-Capture (LoC) [23]. However, this method is not preferred for our approach for 3D interconnect testing: the opposite value is to be captured from the functional input path, and this would involve the functional logic further inside the die, which might lead to ATPG conflicts and would make the boundary model of the die larger, disclosing more of the die internal design IP.

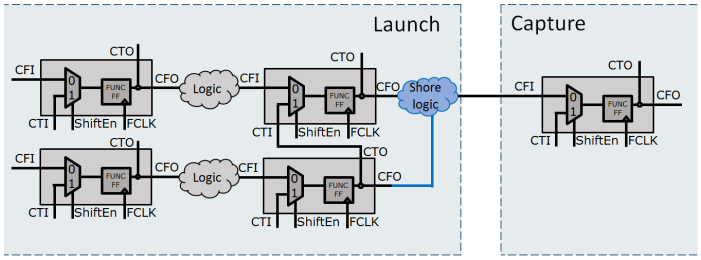


Figure 6: Paths involved in transition launch and capture.

In another method called Launch-on-Shift (LoS) [24], the transition value is taken from the test data input (`CTI`). Although this method removes the need to include more functional logic in the boundary model, it has two drawbacks: the value at the `CTI` input may create conflicts in the shore logic (see Figure 6) and the `ShiftEn` signal must switch at-speed between the two clock cycles due to the conflict of its required values at the launch (value "1" required to select the CTI input) and capture (value "0" required to select the CFI input) sides. For these reasons, LoS is typically not used for at-speed InTest.

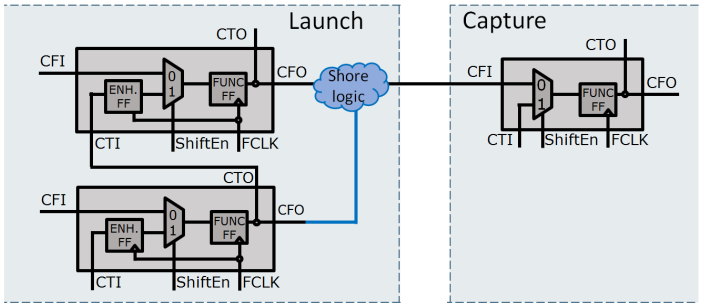


Figure 7: Enhanced Launch on Shift (ELoS).

In our work, we use the Enhanced-Launch-on-Shift (ELoS, see Figure 7) method [25], also referred to as Enhanced Scan. The idea is to insert at the `CTI` input of the shared wrapper cell a dedicated flip-flop used to launch the transition. This flip-flop can be set to the transition value by scanning without the risk of creating ATPG conflicts in the shore logic. However, this method still suffers from the need to switch the `ShiftEn` signal at-speed. To address this issue, we create separate `ShiftEn` signals for the launch and capture side. At the launch side, we override it with the `EXTEST` signal from the wrapper controller block using an OR gate. Now, the `ShiftEn` signal can be de-asserted (slowly) before launch/capture so that the receiving side captures the value of the interconnect while the driving side where the `ShiftEn` is overriden by `EXTEST` continues to shift, creating transitions.

A drawback of the ELoS method is the area overhead of inserting the additional flip-flop. However, since the amount of functional flip-flops that need to be enhanced is not very large (around the same number as 3D interconnects under test, i.e., thousands) compared to the overall number of the flip-flops in a typical large-scale design (millions), we consider this overhead acceptable.

7 Case Study: 'Vesuvius-3D' Stack

This section presents experimental simulation results on the 3D-DfT Demonstrator circuit in a 3D-SIC test chip named 'Vesuvius-3D'. This test chip was manufactured by GlobalFoundries and IMEC; a bare stack, prior to packaging, is shown in Figure 8 [7]. The 3D-DfT Demonstrator circuit in 'Vesuvius-3D' consists of two small ISCAS'89 benchmark circuits [26]: s5378 as bottom die and s1423 as top die. Functionally, this stack probably does not make sense; we use it only as a vehicle for structural testing with our 3D-DfT architecture. In this paper, of particular interest are the numbers of die I/Os. The *primary* port (= bottom side) of the bottom die counts 30 functional inputs and 32 functional outputs, the *secondary* port (= top side) of the bottom die connects to the *primary* port (= bottom side) of the top die with 17 functional outputs and 5 functional inputs implemented as TSVs and micro-bump pairs [7].

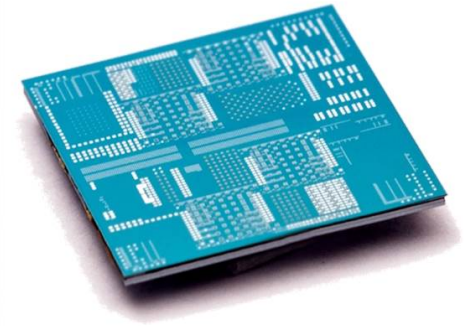


Figure 8: Photo of a bare 'Vesuvius-3D' stack.

For our experiments, we created two modified simulation versions of both dies: (1) completely registered (referred to as s1423R and s5378R) and (2) partially registered (referred to as s1423P and s5378P). In the s1423P circuit 1 out of total 22 I/Os is unregistered. In the s5378P circuit the ratio is 7 out of 84 I/Os, where 5 of the unregistered I/Os are in the secondary port. We consider two stacks: s1423R-on-s5378R and s1423P-on-s5378P. The wrapped version of the $*$R stack is guaranteed free of shore logic, while the wrapped version of the $*$P stack does contain some shore logic if only shared wrapper cells are used.

First, we analyze the impact of the threshold for the number of functional flip-flops per I/O fanin/fanout (Section 4.1) on the num-

ber of inserted shared vs. dedicated wrapper cells. We selected three values for the threshold: 1, 5, and 10. The experiment results are listed in Table 1. As expected, the fully registered (∗R) versions of both dies have each of their I/Os wrapped with a single shared wrapper cell (SWC). However, for the partially registered (∗P) versions of the dies, some I/Os require more than one functional flip-flop to become a SWC. For example, one of the I/Os of s1423P required 9 SWCs, while other I/Os required just one. From the s5378P design with 84 I/Os, only 77 of them were registered, while the remaining 7 required 35 SWCs. On average, shared wrapping requires 1.36 wrapper cells per I/O for s1423P and 1.33 wrapper cells per I/O for s5378P. Considering that the ELoS approach adds one flip-flop per SWC, shared wrapping is affordable in case of circuits that include shore logic.

Fan-in/out threshold	s1423R	s1423P		s5378R	s5378P	
	SWC	SWC	DWC	SWC	SWC	DWC
1	22	21	1	84	77	7
5	22	21	1	84	98	2
10	22	30	0	84	112	0

Table 1: Impact of the fan-in/out threshold on SWC/DWC numbers.

Table 2 lists the boundary-model sizes as fraction of the original full netlist (both expressed in gate counts) for all four circuits s1423R, s1423P, s5378R, and s5378P. Note that due to the small size of the circuits in our case study, the relative size of the boundary model is rather large; for industrial SOCs it typically amounts to 1–2%. Table 2 clearly demonstrates that the boundary model grows in size due to the inclusion of the shore logic for the ∗P circuits.

Circuit type	s1423 (Top)	s5378 (Bottom)
Fully registered (∗R)	18.4%	24.6%
Partially registered (∗P)	23.5%	33.7%

Table 2: Boundary-model size as fraction of the full netlist.

In Table 3 the pattern set sizes generated by two ATPG algorithms for the two stack variants are shown. It can be seen that the shore logic does not significantly increase the pattern count: five extra patterns for Logic-ATPG algorithms and two extra patterns for Interconnect-ATPG. Considering the total pattern set size, the increase is negligible.

Pattern set	Algorithm	∗R	∗P
INTEST s1423∗	Logic-ATPG	66	67
INTEST s5378∗	Logic-ATPG	120	124
EXTEST	Logic-ATPG	n.a.	12
	Interconnect-ATPG	2	4
Total		188	207

Table 3: Pattern set sizes.

8 Conclusion

At-speed testing of the interconnect in 2.5D- and 3D-SICs is necessary to guarantee the dynamic performance of the connections between the dies. In this work, we propose a method for this purpose that is based on testing the full register-to-register paths that include the interconnect lines. Our method covers the case when besides the interconnect there is some combinational shore logic at the die boundary. The dies are inland-wrapped and a boundary model is extracted for INTEST pattern migration, EXTEST pattern generation, and IP protection. A case study with small benchmark designs was performed to show that the method is effective and efficient in the presence of shore logic. The DfT hardware overhead and the pattern set size increases due to shore logic are shown to be small.

Acknowledgments

The authors thank Patrick Haspel and Anton Klotz of Cadence Design Systems, Artur Jutman of Testonica Lab and Tallinn University of Technology and Tobias Burgherr of FHNW, Switzerland for their support in this project.

References

[1] Robert S. Patti. Three-Dimensional Integrated Circuits and the Future of System-on-Chip Designs. *Proceedings of the IEEE*, 94(6):1214–1224, June 2006. doi:10.1109/JPROC.2006.873612.

[2] Eric Beyne and Bart Swinnen. 3D System Integration Technologies. In *Proceedings of IEEE International Conference on Integrated Circuit Design and Technology (ICICDT)*, pages 1–3, June 2007. doi:10.1109/ICICDT.2007.4299568.

[3] Philip Garrou, Christopher Bower, and Peter Ramm, editors. *Handbook of 3D Integration – Technology and Applications of 3D Integrated Circuits*. Wiley-VCH, Weinheim, Germany, August 2008. ISBN 978-3-527-33265-6.

[4] Erik Jan Marinissen, Jouke Verbree, and Mario Konijnenburg. A Structured and Scalable Test Access Architecture for TSV-Based 3D Stacked ICs. In *Proceedings IEEE VLSI Test Symposium (VTS)*, pages 269–274, April 2010. doi:10.1109/VTS.2010.5469556.

[5] Erik Jan Marinissen et al. 3D DfT Architecture for Pre-Bond and Post-Bond Testing. In *Proceedings IEEE International Conference on 3D System Integration (3DIC)*, November 2010. doi:10.1109/3DIC.2010.5751450.

[6] Erik Jan Marinissen et al. A DfT Architecture for 3D-SICs Based on a Standardizable Die Wrapper. *Journal of Electronic Testing: Theory and Applications*, 28(1):73–92, February 2012. doi:10.1007/s10836-011-5269-9.

[7] Erik Jan Marinissen et al. Vesuvius-3D: A 3D-DfT Demonstrator. In *Proceedings IEEE International Test Conference (ITC)*, October 2014. doi:10.1109/TEST.2014.7035332.

[8] Sergej Deutsch et al. DfT Architecture and ATPG for Interconnect Tests of JEDEC Wide-I/O Memory-on-Logic Die Stacks. In *Proceedings IEEE International Test Conference (ITC)*, pages 1–10, November 2012. doi:10.1109/TEST.2012.6401569.

[9] Sandeep K. Goel et al. Test and Debug Strategy for TSMC CoWoS Stacking Process Based Heterogeneous 3D IC: A Silicon Case Study. In *Proceedings IEEE International Test Conference (ITC)*, pages 1–10, September 2013. doi:10.1109/TEST.2013.6651893.

[10] Christos Papameletis et al. Automated DfT Insertion and Test Generation for 3D-SICs with Embedded Cores and Multiple Towers. In *Proceedings IEEE European Test Symposium (ETS)*, pages 15–20, May 2013. doi:10.1109/ETS.2013.6569350.

[11] Christos Papameletis et al. A DfT Architecture and Tool Flow for 3D-SICs with Test Data Compression, Embedded Cores, and Multiple Towers. *IEEE Design & Test of Computers*, 32(4):40–48, July/August 2015. doi:10.1109/MDAT.2015.2424422.

[12] Chun-Chuan Chi et al. DfT Architecture for 3D-SICs with Multiple Towers. In *Proceedings IEEE European Test Symposium (ETS)*, pages 51–56, May 2011. doi:10.1109/ETS.2011.52.

[13] Sergej Deutsch et al. Automation of 3D-DfT Insertion. In *Proceedings IEEE Asian Test Symposium (ATS)*, pages 395–400, November 2011. doi:10.1109/ATS.2011.58.

[14] Brion Keller et al. Efficient testing of hierarchical core-based SOCs. In *Proceedings IEEE International Test Conference (ITC)*, pages 1–10, October 2014. doi:10.1109/TEST.2014.7035292.

[15] Ozgur Sinanoglu et al. Test Data Volume Comparison of Monolithic Testing vs. Modular SOC Testing. *IEEE Design & Test of Computers*, 26(3):25–37, May/June 2009. doi:10.1109/MDT.2009.65.

[16] Paul Wagner. Interconnect Testing with Boundary Scan. In *Proceedings IEEE International Test Conference (ITC)*, pages 52–57, October 1987.

[17] Jhih-Wei You et al. Performance Characterization of TSV in 3D IC via Sensitivity Analysis. In *Proceedings IEEE Asian Test Symposium (ATS)*, pages 389–394, December 2010. doi:10.1109/ATS.2010.73.

[18] Jhih-Wei You et al. In-Situ Method for TSV Delay Testing and Characterization Using Input Sensitivity Analysis. *IEEE Transactions on VLSI Systems*, pages 443–453, March 2013. doi:10.1109/TVLSI.2012.2187543.

[19] Yu-Hsiang Lin et al. Parametric Delay Test of Post-Bond Through-Silicon Vias in 3-D ICs via Variable Output Thresholding Analysis. *IEEE Transactions on Computer-Aided Design*, pages 737–747, May 2013. doi:10.1109/TCAD.2012.2236837.

[20] Shi-Yu Huang et al. On-Line Transition-Time Monitoring for Die-to-Die Interconnects in 3D ICs. In *Proceedings IEEE Asian Test Symposium (ATS)*, pages 162–167, November 2014. doi:10.1109/ATS.2014.39.

[21] Shreepad Panth and Sung Kyu Lim. Transition delay fault testing of 3D ICs with IR-drop study. In *Proceedings IEEE VLSI Test Symposium (VTS)*, pages 270–275, April 2012. doi:10.1109/VTS.2012.6231065.

[22] Anis Uzzaman et al. Automated Handling of Programmable On-Product Clock Generation (OPCG) Circuitry for Delay Test Vector Generation. In *Proceedings IEEE International Test Conference (ITC)*, pages 1–10, October 2007. doi:10.1109/TEST.2007.4437610.

[23] Jacob Savir and Srinivas Patil. On Broad-Side Delay Test. *IEEE Transactions on VLSI Systems*, pages 368–372, September 1994. doi:10.1109/92.311647.

[24] Jacob Savir. Skewed-Load Transition Test: Part I, Calculus. In *Proceedings IEEE International Test Conference (ITC)*, pages 705–, September 1992. doi:10.1109/TEST.1992.527892.

[25] Bulent I. Dervisoglu and Gayvin E. Stong. Design for Testability Using Scanpath Techniques for Path-Delay Test and Measurement. In *Proceedings IEEE International Test Conference (ITC)*, pages 365–, October 1991. doi:10.1109/TEST.1991.519696.

[26] Franc Brglez, David Bryan and Krzysztof Kózmiński. Combinational Profiles of Sequential Benchmark Circuits. In *Proceedings International Symposium on Circuits and Systems (ISCAS)*, pages 1924–1934, May 1989. doi:10.1109/ISCAS.1989.100747.