

A DfT Architecture for 3D-SICs Based on a Standardizable Die Wrapper

Erik Jan Marinissen · Chun-Chuan Chi ·
Mario Konijnenburg · Jouke Verbree

Received: 17 January 2011 / Accepted: 8 November 2011 / Published online: 13 December 2011
© The Author(s) 2011. This article is published with open access at Springerlink.com

Abstract Process technology developments enable the creation of three-dimensional stacked ICs (3D-SICs) interconnected by means of Through-Silicon Vias (TSVs). This paper presents a 3D Design-for-Test (DfT) architecture for such 3D-SICs that allows pre-bond die testing as well as mid-bond and post-bond stack testing. The architecture enables a modular test approach, in which the various dies, their embedded IP cores, the inter-die TSV-based interconnects, and the external I/Os can be tested as separate units, which allows flexible optimization of the 3D-SIC test flow and provides yield monitoring and first-order fault diagnosis. The architecture builds on and reuses existing DfT hardware at the core, die, and product level. Its main new component is a die-level wrapper, which can be

based on either IEEE Std 1149.1 or IEEE Std 1500. The paper presents a conceptual overview of the architecture, as well as implementation aspects. Experimental results show that the implementation costs are negligible for medium to large dies.

Keywords 3D stacked ICs · Through-silicon via · Manufacturing test · Design-for-test · Wrapper · Test access mechanism · Standardization · IEEE 1149.1 · IEEE 1500

1 Introduction

Vertical stacking of multiple integrated circuits has benefits in terms of combining heterogeneous technologies and achieving a small footprint. The semiconductor industry is preparing itself to make a major step forward in stacking in the third (vertical) dimension, now that the technology of Through-Silicon Vias (TSVs) is becoming available [2, 6, 29]. TSVs are conducting nails which extend out of the back-side of a thinned-down die, enabling the vertical interconnect to another die [27, 32]. TSVs are high-density, low-capacitance interconnects compared to traditional wire-bonds, and hence allow for many more interconnections between stacked dies, while operating at higher speeds and consuming less power [1]. TSV-based three-dimensional technologies enable the creation of a new generation of ‘super chips’ by opening up new architectural opportunities [20, 38]. These so-called 3D Stacked ICs (3D-SICs) combine a smaller form factor and lower overall manufacturing costs [35] with many other compelling benefits, and hence their technology is quickly gaining ground.

Responsible Editor: Y. Zorian

E. J. Marinissen (✉) · C.-C. Chi · J. Verbree
IMEC vzw, Kapeldreef 75, 3001 Leuven, Belgium
e-mail: erik.jan.marinissen@imec.be

C.-C. Chi
Department of Electrical Engineering,
National Tsing-Hua University, Hsinchu,
30013 Taiwan, Republic of China
e-mail: ccchi@larc.ee.nthu.edu.tw

M. Konijnenburg
Holst Centre/IMEC, High Tech Campus 31,
5656AE Eindhoven, The Netherlands
e-mail: mario.konijnenburg@imec-nl.nl

J. Verbree
Department of Computer Engineering,
Delft University of Technology, Mekelweg 4,
2628 CD Delft, The Netherlands
e-mail: j.verbree@student.tudelft.nl

Like all micro-electronic products, 3D-SICs need to be tested for manufacturing defects incurred during their many, high-precision, and hence defect-prone manufacturing steps. Next to all basic and most advanced test technology issues, 3D-SICs have some unique new test challenges of their own [16, 21, 23], pertaining to (1) test flow, (2) test contents, and (3) test access. Regarding (1), a 3D manufacturing flow allows for many more natural test moments than a conventional 2D flow, and dedicated cost modeling is required to assess at which moments to test (or re-test) what in order to keep the tests both effective and cost-efficient [33]. Regarding (2), new fault models and corresponding tests for TSV-based interconnects need to be developed and although not convincingly identified yet, we also should stay alert for 3D-induced intra-die defects that cause new faults that might escape the conventional test sets. Regarding (3), test access deals with transporting test stimuli in and test responses out of the die- or stack-under-test. Test access challenges exist for wafer probing, where one needs to probe on small and numerous micro-bumps and/or TSV tips and pads under stringent damage requirements [31], and handle and probe non-planar wafers with thinned-die stacks. Test access challenges also exist *within* the dies and stacks, where DfT architectures that span across multiple dies need to be designed, partitioned, and optimized.

This paper describes a 3D DfT architecture that services the test needs of die makers, stack makers, and stack users alike. The architecture is based on a die-level test wrapper that should be included by the various die makers in the designs of the respective dies that together make up the stack. Our 3D DfT architecture supports (1) pre-bond die testing, (2) mid-bond testing of partial stacks, (3) post-bond testing of complete stacks, (4) board-level interconnect testing, as well as (5) (low-bandwidth) in-field test and debug. Our DfT architecture enables a modular test approach [22], in which the various dies, their embedded IP cores, the inter-die TSV-based interconnects, and the external I/Os can be tested as separate units. This modular test approach provides yield monitoring and first-order fault diagnosis, and allows for flexible inclusion (or exclusion) and scheduling of (re-)tests at the various product stages, for example depending on the maturity of the manufacturing process.

The remainder of this paper is organized as follows. Section 2 describes related prior work on test access architectures for 3D-SICs. Section 3 provides an overview of test access architecture standards for PCBs and SOCs, which, like 3D-SICs, are also built from interconnected smaller components. Section 4

describes the assumptions and requirements that form the foundation of our 3D DfT architecture. The architecture itself is presented in Section 5; this section also describes the two alternative variants, based on IEEE Std 1149.1 [9, 28] and IEEE Std 1500 [5, 11]. Section 6 details various implementation aspects of the die-level wrapper, and in Section 7 we present experimental results. Section 8 concludes this paper.

2 Related Prior Work

Early papers addressing the testability issues of 3D-SICs are by Lewis and Lee [17, 18]. They focus on pre-bond die testing to increase the compound stack yield and propose a “scan island” approach, which is essentially the wrapper technique from IEEE Stds 1149.1 [9, 28] and 1500 [5, 11] under a different name.

Subsequent papers on 3D-SIC testing implicitly propose a test access architecture, while focusing on optimizing the design parameters of that architecture to minimize the resulting test length and/or the associated wire length [13, 14, 40, 41]. Wu et al. [40] propose three scan chain optimization algorithms, taking the length of TSV-based interconnects into account. Implicitly, this paper assumes that a single logic test unit is partitioned over multiple tiers, which seems rather unrealistic. Therefore, in [41], Wu et al. propose a core-based design and test approach (as is common for 2D-SOCs) in which each core resides on a single tier. The paper proposes a Test Access Mechanism (TAM) optimization approach based on Integer Linear Programming (ILP), which tries to minimize the resulting test length under a constraint for the number of additional ‘test TSVs’. Both papers [40, 41] focus exclusively on post-bond stack testing, and ignore the requirements for pre-bond die testing.

Jiang et al. [13] describe a TAM optimization approach based on simulated annealing that minimizes test length and TAM wire length with a user-defined cost weight factor. They assume a modular core-based 3D-SIC test approach and take both pre-bond and post-bond test lengths into account. The paper lacks realistic constraints on wafer and packaged stack test access, due to which it unrealistically allows TAMs to start and end at any stack tier. Successor paper [14] remedies this partly, by working with pre-bond tests that are applied through dedicated probe pads at the die in question, for which a maximum count is assumed. The paper proposes heuristics that determine a post-bond stack test architecture, from which segments are reused as much as possible to build additional die-level test architectures for the pre-bond tests, while meeting the

maximum probe pad count constraint and minimizing test length and TAM wire length.

Lo et al. [19] proposes a test architecture for 3D-SICs, considering pre-bond, post-bond, as well as TSV-based interconnect testing. The proposed architecture reuses the test wrapper of cores embedded in the various dies to support modular testing in 3D-SICs, achieving a small area cost. This approach works fine, under the assumption that there is no circuitry within the dies in between the wrapped embedded cores. Unfortunately, this is often not the case; in [8], Goel et al. describe a typical industrial SOC for which, despite its large embedded TriMedia cores, most of the on-chip circuitry is situated in between the embedded cores.

In contrast to the prior work by others, our paper starts out by identifying realistic constraints and requirements set forward by, among others, wafer probe technology and test flow set-ups. Subsequently, we focus on the design of a generic and structured test access architecture. The architecture is scalable in the sense that its design parameters can be optimized for varying core, die, and stack parameters, although the focus of our paper is *not* on those optimization procedures. The prior work has focused on testing the cores in the various dies constituting the 3D-SIC, but has ignored testing the circuitry within a die in between the cores, as well as it has ignored testing the (TSV-based) inter-die interconnects. The prior work also did not identify how existing DfT standards and test access architectures can be leveraged. Finally, test control and instructions were ignored in the prior work. We address all the above issues.

3 Related Test Access Standards

Two successful test access standards for systems built out of pre-defined components are IEEE Std 1149.1 [9, 28] for chips on Printed Circuit Boards (PCBs) and IEEE Std 1500 [5, 11, 22] for embedded cores in

System-on-Chips (SOCs). In this section, we briefly describe the similarities and differences of both standards, that serve as a starting point for our proposed 3D-SIC test access architecture.

3.1 Test Access Architecture for PCBs

The commonly-used test access architecture for PCBs is based on IEEE Std 1149.1, Boundary Scan (a.k.a. ‘JTAG’) [9, 28]. In order for chips to be compliant to IEEE 1149.1, a small hardware wrapper is added to them. IEEE 1149.1 works through a narrow single-bit interface, as every JTAG terminal requires an additional chip pin and these are considered expensive. Fortunately, the prime focus of IEEE 1149.1 is PCB interconnect testing, and that requires only a small number of test patterns [25]. The single-bit interface pins are called TDI and TDO, and they transport both instructions and test data. The control interface consists of the pins TCK, TMS (and optionally TRSTN).

For an example PCB containing three chips, a common JTAG-based test access architecture is depicted in Fig. 1. The control signals are broadcast to all chips, while the TDI-TDO pins are concatenated through the chips. The broadcast control signals can configure the TAP Controller finite state machine in a mode in which it is able to receive instructions, which are subsequently scanned into the Instruction Register (IR) via the daisychained TDI-TDO interface. Note that this allows for different instructions for different chips; for example, Chip B can be configured in INTEST mode, while Chips A and C are configured in BYPASS mode. Then, the chips are brought into their instructed test modes via the broadcast control signals and test data is scanned in and out again via the daisychained TDI-TDO interface. The selected test data register (e.g., the bypass register, a Boundary Scan Register (BSR), or a chip-internal scan chain) depends on the instruction, and can be different for different chips; in any case, it is a single shift register, as shown in Fig. 1.

Fig. 1 Board-level test access architecture for chips based on IEEE 1149.1

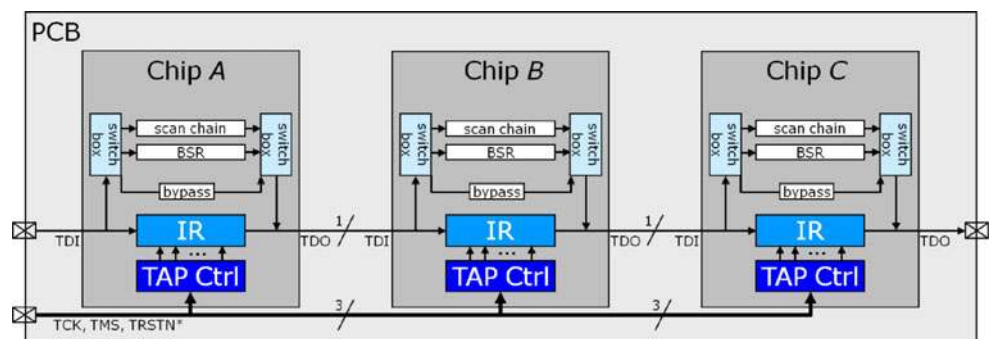
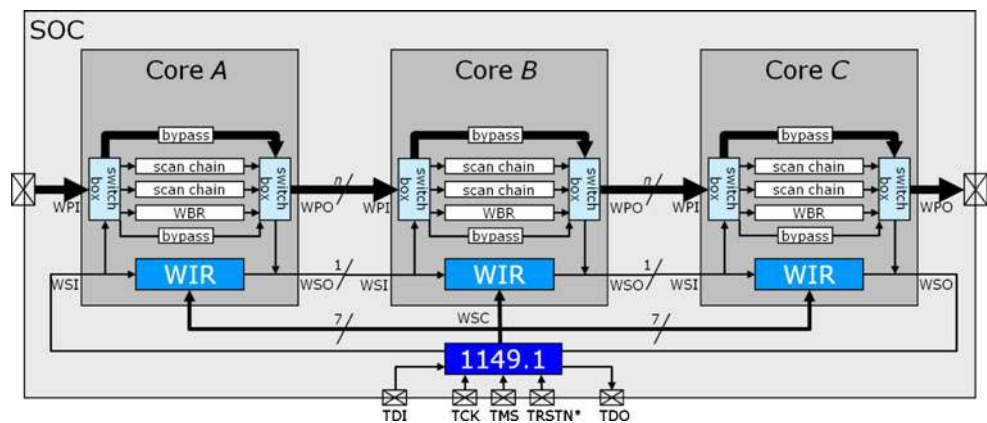


Fig. 2 SOC-level test access architecture for cores based on IEEE 1500



3.2 Test Access Architecture for 2D-SOCs

The commonly-used test access architecture for (two-dimensional) SOC containing embedded IP cores is based on IEEE Std 1500 [5, 11, 22]. Like IEEE 1149.1, IEEE 1500 adds a small hardware wrapper around the module-under-test, which in this case is an embedded core. As shown in Fig. 2, the test access architecture for an IEEE 1500-based SOC shows similarities to IEEE 1149.1-based PCBs. Control signals are broadcast to all cores. Once configured in the appropriate mode, instructions are shifted into the Wrapper Instruction Register (WIR) via the daisy-chained wsi-wso interface. That same instruction interface also doubles as a single-bit test data interface. However, next to the similarities, there are also significant differences between IEEE 1149.1- and IEEE 1500-based test access architectures. Below, we list the most important ones.

- Unlike IEEE 1149.1, the focus of IEEE 1500 is not (only) on testing wiring interconnects between cores. First of all, the interconnect circuitry in between IP cores typically does not consist of only wires, but is often formed by deep sequential logic [8]. In addition, IEEE 1500 is meant to also support the testing of the cores themselves, and IP cores are often significantly-sized and complex design entities. Therefore, the test data volumes involved are typically quite large, and as a result, a single-bit test data interface would not suffice. Hence, IEEE 1500 has an optional n -bit (“parallel”) test data interface (named wpi and wpo), where n can be scaled by the user to match the test data volume needs of the IP core in question.
- Adding wider interfaces to embedded IP cores does not add chip pins as in IEEE 1149.1, but only core terminals; and they are considered to be significantly less expensive than chip pins.

- IEEE 1149.1 has two (or three) standardized control pins, which are expanded into multiple control signals within the chip by the TAP Controller. IEEE 1500 has no TAP Controller, but receives its control signals directly. These are six (or seven) signals: wrck, wrstn, selectwir, shiftwr, capturewr, updatewr (and optionally transferdr) [5, 11, 22].

Figure 2 also features a parallel wrapper bypass. This bypass is not mandated by IEEE 1500, but often implemented to shorten the test access path to other cores in the same TAM [7]. It is the task of the *switch boxes* in Fig. 2 to make an effective mapping between the active WIR instruction mode and the TAM-to-chain connections.

IEEE 1500 only standardizes the core-level test wrapper, and *not* the SOC-level test access architecture of the optional parallel TAMs. At the SOC-level, optimizations can be made with respect to TAM type [24, 34], TAM architecture [7], and corresponding test schedule. In a typical implementation, as shown in Fig. 2, the SOC itself is equipped with an IEEE 1149.1 wrapper to facilitate board-level testing. The IEEE 1500 serial interface (wsc, wsi, and wso) is multiplexed onto the IEEE 1149.1 Test Access Port [5] to save otherwise additional test pins. The IEEE 1500 parallel interface (wpi and wpo) can be multiplexed onto the functional external pins, as is also common for regular scan chains; this saves otherwise additional test pins.

4 Assumptions and Requirements

In this paper, we consider 3D-SICs for which all interdie connections are implemented by means of TSVs and for which all external connections (*pins*) of the

stack are located on one side of one of the extreme tiers, i.e., top or bottom. To simplify our descriptions, we assume in the remainder of this paper that all pins are in the *bottom* die; note that this assumption is without loss of generality, as we can always swap the references to top and bottom die. We furthermore assume that on top of a die *b*, one or multiple dies can be stacked; we refer to *b* as the ‘base’ die, on which one or multiple ‘towers’ are stacked. Die *b* can be a stacked die itself, allowing the possibility of ‘sub-towers’. Figure 3 shows three example 3D-SICs, each consisting of three stack layers: (a) wire-bond from the bottom die, (b) wire-bond from the top die (which therefore is referred to as ‘bottom die’), and (c) flip-chip connections from the bottom die with two ‘towers’ in the third layer.

A 3D DfT architecture should service the test needs of die maker(s), stack maker, and stack users alike. The die maker(s) might execute *pre-bond* tests, covering the intra-die circuitry and possibly also the TSVs [23]. The stack maker might execute *mid-bond* and/or *post-bond* tests on not-yet-packaged and/or packaged die stacks; these tests might cover intra-die circuitry (possibly as re-test), as well as the inter-die TSV-based connections [23]. It is assumed that it is a requirement from the stack user that the overall stack product is IEEE 1149.1-compliant [9, 28] on its pins, in order to facilitate board-level interconnect testing.

We assume a 3D-SIC of which the constituting dies are *scan testable*; for example, this can include scan-tested digital logic, BIST-ed embedded memories, or even scan-enabled analog cores. To minimize silicon area, we want to re-use the existing intra-die DfT infrastructure as much as possible: internal scan chains, test control, test data compression circuitry, built-in self-test, etc. We assume that additional external test pins beyond what is required functionally and for IEEE 1149.1 are expensive and hence should be avoided. In contrast, we assume that some additional TSV-based interconnects between tiers for the purpose of test are relatively affordable; e.g., IMEC’s *via-middle* TSVs are made at a 10 μm minimum pitch [27, 32].

Today’s probe technology is insufficiently precise and damage-free to provide probe access on small

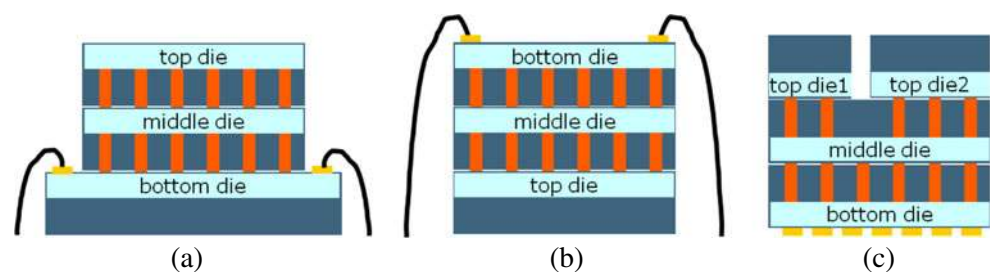
micro-bumps, TSV tips, nor TSV landing pads [23]. As long as that is the case, it is a requirement to provide dedicated *probe pads* for pre-bond wafer test access [14, 17, 23] for all dies in the stack, apart from the bottom die.

For the mid-bond and post-bond stack tests, test access is only possible via the external I/Os of the bottom die. This implies that signals for test control and test data exclusively come from and go to the bottom die, and hence have to make a ‘u-turn’; we refer to these as *TestTurns*. Also, in order to reach dies higher up in the stack, the underlying dies need to cooperate in a dedicated mode which requires additional DfT and TSVs which we refer to as *TestElevators*.

We require the 3D DfT architecture to be *scalable* in multiple ways. We will equip it with both a fixed one-bit (‘*serial*’), as well as a scalable multi-bit (‘*parallel*’) test access mechanism. The focus of the serial mechanism is on debug and diagnosis; it provides a low-cost, low-bandwidth mechanism for test configuration instructions and test data, which can be used even if the stack product is soldered onto a printed circuit board. The focus of the scalable parallel mechanism is on high-volume production testing; it provides a trade-off between implementation costs and test access bandwidth. In addition, the architecture should be scalable in the sense that it works for an undetermined number of stack tiers. Also, the architecture should not predestine a middle die to a certain tier level, such that dies that adhere to the architecture can function at any level in the stack hierarchy. The bottom and top dies are obviously exempt from this requirement, as they play a special role in the stack.

A final requirement is that the 3D DfT architecture should support a *modular* test approach [7, 22], as opposed to an approach in which the entire stack is tested as one monolithic entity. A modular test considers the various dies and TSV-based interconnect layers as separate test units; for complex dies, it is very well possible that they are further sub-divided into multiple finer-grain test modules, e.g., embedded cores. A modular test approach allows to optimize for circuit-specific fault models, enables flexible test flow

Fig. 3 Three examples 3D-SIC: **a** wire-bond from *bottom* die, **b** wire-bond from *top* die (which therefore is referred to as ‘*bottom die*’), and **c** flip-chip from *bottom* die with two ‘*towers*’ on *top*



optimization, and provides yield monitoring and first-order fault diagnosis.

5 3D DfT Architecture

5.1 Architecture Overview

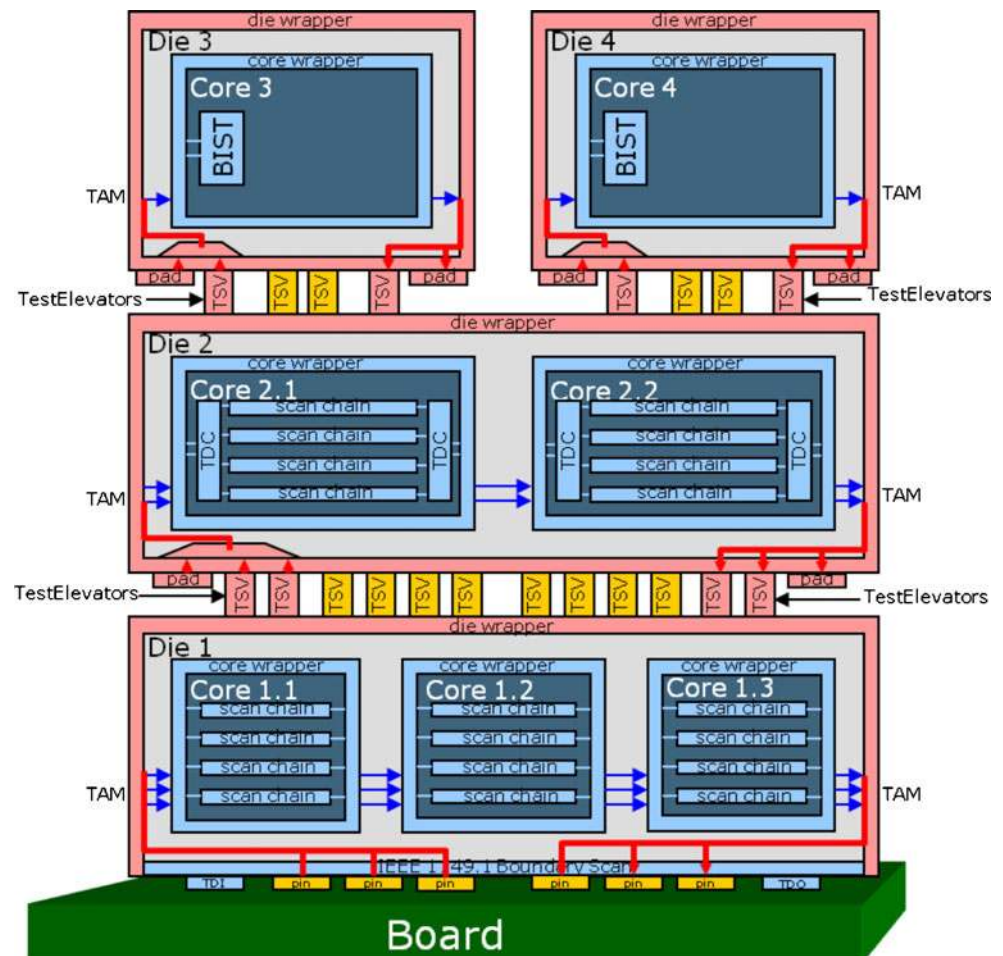
The 3D DfT architecture consists of a set of cooperating die-level test wrappers, one for each die in the stack. A conceptual overview of the architecture is depicted in Fig. 4. The figure shows an example stack consisting of four dies; Dies 3 and 4 are side-by-side stacked on top of Die 2, which in turn is stacked on top of Die 1. The functional I/Os of the four dies are shown in yellow. At the bottom of bottom Die 1 are the external functional I/Os ('pins'). The dies are interconnected by means of functional TSVs. The figure shows in light-blue the conventional, already existing DfT infrastructure. The external I/Os of the stack, all located in the bottom die, are wrapped by IEEE 1149.1 Boundary Scan; this

requires a limited number of additional pins, of which two (TDI and TDO) are shown. Furthermore, the dies have existing intra-die DfT, exemplified by internal scan chains, Test Data Compression (TDC), Built-In Self Test (BIST), IEEE 1500-compliant core wrappers, and Test Access Mechanisms (TAMs). Shown in light-red is the new 3D DfT, comprised of test wrappers around each die in the stack.

The main features of the die-level wrapper are the following: (1) a serial interface for wrapper instructions and low-bandwidth test data and a scalable, parallel interface for higher-bandwidth test data, (2) *TestTurns* in every die that feed test data back toward the pins of the bottom die, (3) TestElevators that propagate test signals up and down through the stack, (4) optionally, a scalable number of dedicated probe pads on all non-bottom dies to enable pre-bond die testing, and (5) an optional hierarchical inclusion/exclusion mechanism for embedded IP cores, if any, and dies higher up in the stack, that prevents unbridled growth of test lengths.

Our two proposed 3D die wrappers are based on either one of the existing DfT standards IEEE 1149.1

Fig. 4 Conceptual overview of our 3D DfT architecture



and IEEE 1500. In the subsequent sub-sections, we describe both alternative architectures in more detail.

5.2 Die-level Wrapper Based on IEEE 1149.1

Stacked dies in a 3D-SIC can be considered similar to chips on a PCB. Consequently, the IEEE 1149.1 chip wrapper can be used and enhanced to form a die-level wrapper for 3D-SICs. Figure 5 shows such a 3D-enhanced die wrapper based on IEEE 1149.1, for the cases where a single ‘tower’ (Fig. 5a) or two ‘towers’ (Fig. 5b) will be stacked on top. The 3D enhancements are highlighted in orange and comprise the following five items.

1. *Parallel Test Port:* In order to support efficient high-volume testing of the die’s circuitry, a parallel, scalable test port of user-defined width n is provisioned. We refer to the inputs and outputs of this port as resp. TPI and TPO.
2. *TestTurns:* The extended IEEE 1149.1 interface, consisting of TCK, TMS, TRSTN* (optional), TDI-TDO, and TPI-TPO, is located at the bottom side of the die. In the output paths toward TDO and TPO, we insert pipeline registers for a clean timing interface (especially important if many dies are stacked).
3. *TestElevators:* The extended IEEE 1149.1 interface is copied at the top side of the die, toward higher-up dies. We give these I/Os the same names, post-fixed with the letter “s” (for “stack”) and a sequence number in case multiple such test ports exist (Fig. 5b).
4. *Probe Pads:* As long as probe technology does not provide us with solutions to safely probe micro-

bumps and/or TSV tips and landing pads, all non-bottom dies are equipped with additional probe pads. If implemented, these probe pads are mandatory on the serial interface (TCK, TMS, TRSTN* (optional), TDI, and TDO), and optional and scalable on the parallel interface (TPI-TPO). If the parallel TPI-TPO interface coming from the bottom is n bits wide (with $n \geq 0$), the corresponding probe pad interface can be m bits wide, where typically $0 \leq m \leq n$.

5. *Hierarchical Test Mechanism:* Optionally, we equip the die-level Instruction Register (IR) with one or more bits that control in- or exclusion of the test control and test data mechanisms of higher-level dies and/or embedded IP cores, if any. The purpose of this hierarchical mechanism is to prevent an unbridled growth of the length of the various IR and TAM chains. The die-level IR can be equipped with one in-/exclusion control bit per ‘tower’ above it and for its embedded IP cores. The control bits work in a way similar to the Segment Insertion Bit (SIB) of IEEE P1687 [42]. If set, the corresponding die/core is included. By default, the corresponding die/core is excluded and its IR is placed in a safe reset state.

Figure 6 shows the 3D DfT architecture with IEEE 1149.1-based die wrappers for a stack of four dies. The control signals TCK, TMS, and TRSTN* (optional) are broadcast to all dies. The serial and parallel test access mechanisms are daisychained throughout the stack. The middle die has a wrapper as described above. The die wrappers for the top and bottom dies are slightly different. The top dies have no die above them, and hence do not implement TestElevators. The bottom die contains all external I/Os. The parallel interface

Fig. 5 3D-enhanced IEEE 1149.1 die wrapper with **a** one and **b** two test ports at its top side

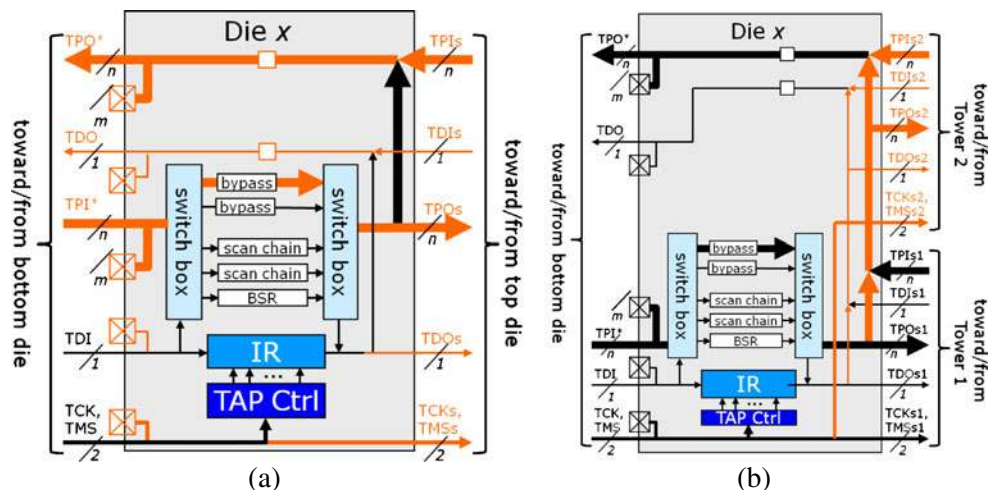
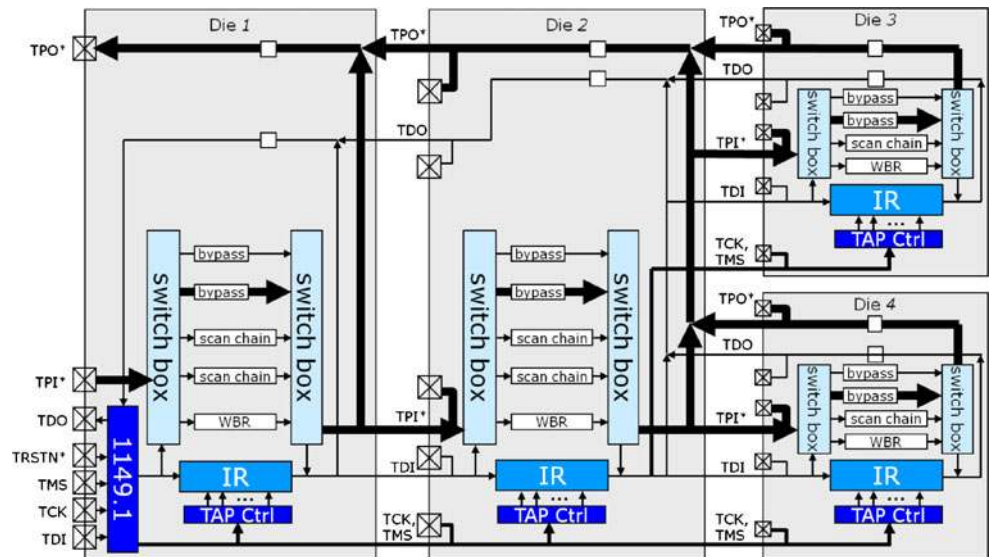


Fig. 6 3D-SIC DfT architecture based on IEEE 1149.1



TPI-TPO can be multiplexed onto existing functional pins. Consequently, the overall 3D DfT architecture does not incur additional stack pins beyond the standard four/five pins interface of IEEE 1149.1.

There exist many alternative uses of IEEE 1149.1 beyond board-level interconnect testing for purposes like silicon and software debug, emulation, in-circuit programming, etc. [10, 15, 30, 36, 39]. These applications have a large hardware and software infrastructure, which relies on the presence of the IEEE 1149.1 features. A potential benefit of basing 3D die-level wrappers on IEEE 1149.1, as described in this section, is that this infrastructure remains operational, also for 3D-SICs.

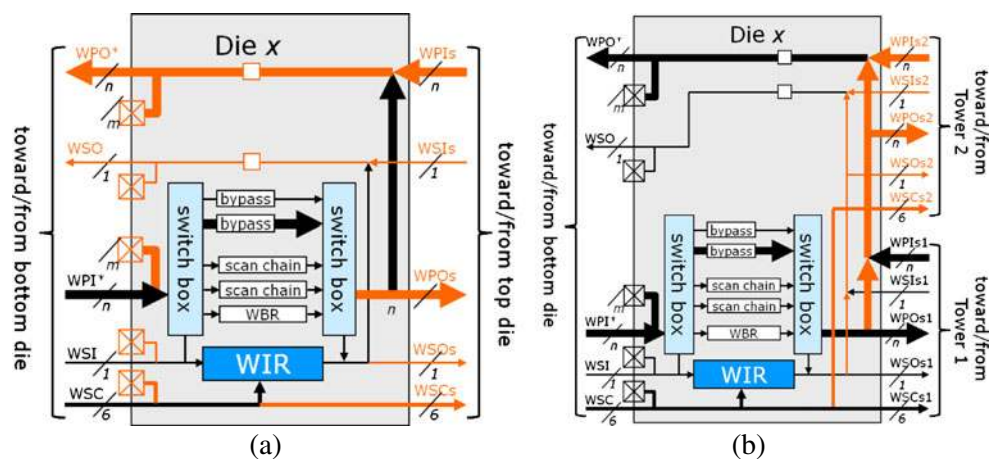
5.3 Die-level Wrapper Based on IEEE 1500

Stacked dies in a 3D-SIC can be considered similar to embedded cores in a System-on-Chip (SOC).

Consequently, the IEEE 1500 core wrapper can be used and enhanced to form a die-level wrapper for 3D-SICs [26]. Figure 7 shows such the 3D-enhanced die wrapper based on IEEE 1500, for the cases where a single ‘tower’ (Fig. 7a) or two ‘towers’ (Fig. 7b) will be stacked on top. The 3D enhancements are highlighted in orange and comprise the following five items.

1. *Parallel Test Port*: The conventional (2D) IEEE 1500 already contains an optional and scalable parallel test port.
2. *TestTurns*: The standard IEEE 1500 interface, consisting of wsc, wsi-wso, and wpi-wpo is located at the bottom side of the die. In the output paths toward wso and wpo, we insert pipeline registers for a clean timing interface (especially important if many dies are stacked).
3. *TestElevators*: The extended IEEE 1500 interface is copied at the top side of the die, toward higher-

Fig. 7 3D-enhanced IEEE 1500 die wrapper with a one and b two test ports at its top side



up dies. We give these I/Os the same names, post-fixed with the letter “s” (for “stack”) and a sequence number in case multiple such test ports exist (Fig. 7b).

4. *Probe Pads*: As long as probe technology does not provide us with solutions to safely probe micro-bumps and/or TSV tips and landing pads, all non-bottom dies are equipped with additional probe pads. If implemented, these probe pads are mandatory on the serial interface (wsc, wsi-wso), and optional and scalable on the parallel interface (wpi-wpo). If the parallel wpi-wpo interface coming from the bottom is n bits wide (with $n \geq 0$), the corresponding probe pad interface can be m bits wide (with $m \geq 0$).
5. *Hierarchical Test Mechanism*: Optionally, we equip the die-level Wrapper Instruction Register (WIR) with one or more bits that control in- or exclusion of the test control and test data mechanisms of higher-level dies and/or embedded IP cores, if any.

Figure 8 shows the 3D DfT architecture with IEEE 1500-based die wrappers for a stack of four dies. The wsc control signals are broadcast to all dies. The serial and parallel test access mechanisms are daisy-chained throughout the stack. The middle die has a wrapper as described above. The die wrappers for the top and bottom dies are slightly different. The top dies have no die above them, and hence do not implement TestElevators. The bottom die contains all external I/Os. Hence, it implements IEEE 1149.1 for board-level interconnect testing. Its serial interface, consisting of wsc and wsi-wso, is connected to its IEEE 1149.1 TAP controller, as is common in conventional SOCs [5], in order to save dedicated pins. The parallel interface

wpi-wpo is multiplexed onto existing functional pins. Consequently, the overall 3D DfT architecture does not incur additional stack pins beyond the standard four/five pins interface of IEEE 1149.1.

The 1500-based architecture in Fig. 8 has large similarities to the one based on IEEE 1149.1 in Fig. 6. In fact, the only major difference is in the number and function of the broadcast control signals (six/seven-bit wsc vs. two/three-bit TCK/TMS/TRSTN*) and the absence or presence of a TAP Controller.

5.4 Operating Modes

3D DfT architectures as described in Sections 5.2 and 5.3 support a number of test modes. The following selections can be made.

- *Serial/Parallel*: using the *Serial* (1-bit) or *Parallel* (n -bit in pre-bond or m -bit in mid- and post-bond) test access mechanism.
- *Prebond/Postbond*: test access via dedicated probe pads (*Prebond*) or TSV-based interconnects from the die below (*Postbond*).
- *Intest/Extest/Bypass*: the test access chain includes the wrapper cells and internal scan chains (*Intest*), or only the wrapper cells (*Extest*), or none of the above and only travels through a bypass register (*Bypass*).
- *Exclude/Include*: the test access chain excludes or includes the corresponding embedded IP cores or dies [4]. This option exists for the embedded cores of this die (*Exccore/Inccore*) and for all k towers (*Extwrk/Inctwrk*) above this die (with $k \geq 0$).
- *Turn/Elevator*: the test access mechanism turns downwards from this die (*Turn*) or it does go up

Fig. 8 3D-SIC DfT architecture based on IEEE 1500

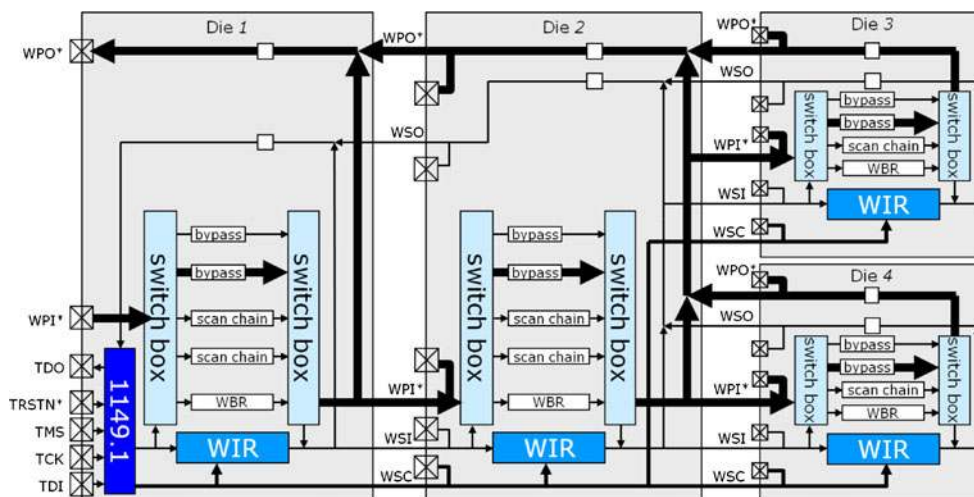


Table 1 Multiplexer control signals for all operating modes

Mode	m1	m2	m3	m4	m5	m6	m7	m8	m9	m10	m11
SerialPrebondBypassTurn	x	0	0	x	x	x	1	0	0	1 AND $\overline{\text{selectWir}}$	1
SerialPrebondIntestTurn	SHIFTWR	1	0	1	1	01	0	0	0	1 AND $\overline{\text{selectWir}}$	1
ParallelPrebondBypassTurn	x	0	0	x	x	x	0	0	0	1 AND $\overline{\text{selectWir}}$	1
ParallelPrebondIntestTurn	SHIFTWR	1	0	0	1	10	1	0	0	1 AND $\overline{\text{selectWir}}$	1
SerialPostbondBypassTurn*	x	0	0	x	x	x	1	0	1	1 AND $\overline{\text{selectWir}}$	1
SerialPostbondIntestTurn	SHIFTWR	1	0	1	1	01	0	0	1	1 AND $\overline{\text{selectWir}}$	1
SerialPostbondExtestTurn	SHIFTWR	0	1	1	0	01	0	0	1	1 AND $\overline{\text{selectWir}}$	1
SerialPostbondBypassElevator	x	0	0	x	x	x	1	0	1	0 AND $\overline{\text{selectWir}}$	0
SerialPostbondIntestElevator	SHIFTWR	1	0	1	1	01	0	0	1	0 AND $\overline{\text{selectWir}}$	0
SerialPostbondExtestElevator	SHIFTWR	0	1	1	0	01	0	0	1	0 AND $\overline{\text{selectWir}}$	0
ParallelPostbondBypassTurn	x	0	0	x	x	x	0	0	1	1 AND $\overline{\text{selectWir}}$	1
ParallelPostbondIntestTurn	SHIFTWR	1	0	0	1	10	1	0	1	1 AND $\overline{\text{selectWir}}$	1
ParallelPostbondExtestTurn	SHIFTWR	0	1	0	0	00	1	0	1	1 AND $\overline{\text{selectWir}}$	1
ParallelPostbondBypassElevator	x	0	0	x	x	x	0	0	1	0 AND $\overline{\text{selectWir}}$	0
ParallelPostbondIntestElevator	SHIFTWR	1	0	0	1	10	1	0	1	0 AND $\overline{\text{selectWir}}$	0
ParallelPostbondExtestElevator	SHIFTWR	0	1	0	0	00	1	0	1	0 AND $\overline{\text{selectWir}}$	0

*Indicates default reset mode

to the next-higher die (*Elevator*). This option exists for all k towers (*Turnk/Elevatork*) above this die (with $k \geq 0$).

The above set-up allows for a large flexibility in test mode configuration, as most instruction options can be combined. Instruction options that cannot be combined are the following.

- In the *Prebond* mode, *Extest* and *Inctwr* options do not make sense, as there are no stack neighbors yet.
- If *Exctwr* is asserted, *Elevator* for the same tower will not work, as we cannot elevate data into a disabled tower of dies.
- *Intest* requires the wrappers of the embedded cores to be included and hence *Intest* cannot be combined with *Exccore*.

For a generic flat design with k towers for which the exclude/include bits are not used, there are in total $4 + 6 \cdot 2^k$ test modes. An exhaustive list of test modes for such a design with $k = 1$ is provided in Table 1. The number of test modes grows to

$6 + 10 \cdot 3^k$ test modes for a hierarchical SOC with embedded cores for which the exclude/include option is implemented for all towers. For such an example with $k = 2$, Fig. 9 shows which combinations of wrapper settings can be made by traversing this so-called ‘railroad diagram’ from left to right. The figure shows the mandatory instruction parts in blue and the optional instruction parts in gray. Some examples of operating modes are *SerialPrebondIntestInccoreExctwr1Turn1Exctwr2Turn2*, *ParallelPostbondIntestInccoreExctwr1Turn1Inctwr2Elevator2*, *SerialPostbondBypassExccoreInctwr1Elevator1Inctwr2Turn2*, and *ParallelPostbondExtestExccoreInctwr1Elevator1Inctwr2Elevator2*. In the example of Fig. 9, in total 96 test modes are possible: six in the pre-bond case, and 90 in the post-bond case.

Combining instructions for the various dies in a stack allows to test one, multiple, or all dies simultaneously, as well as test one, multiple, or all layers of TSV-based interconnects simultaneously. Hence, the proposed architecture allows flexible scheduling during test execution. This can for example be exploited in an

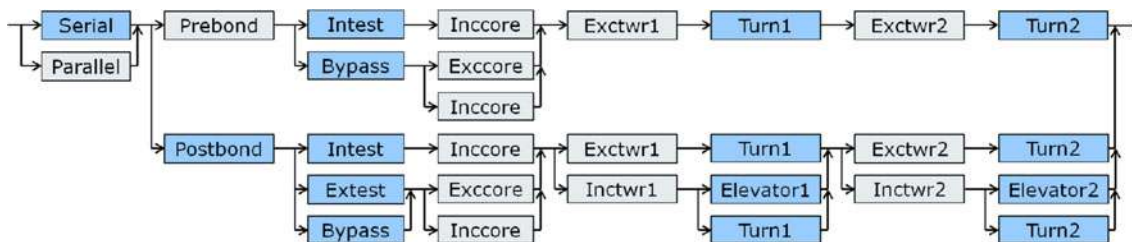
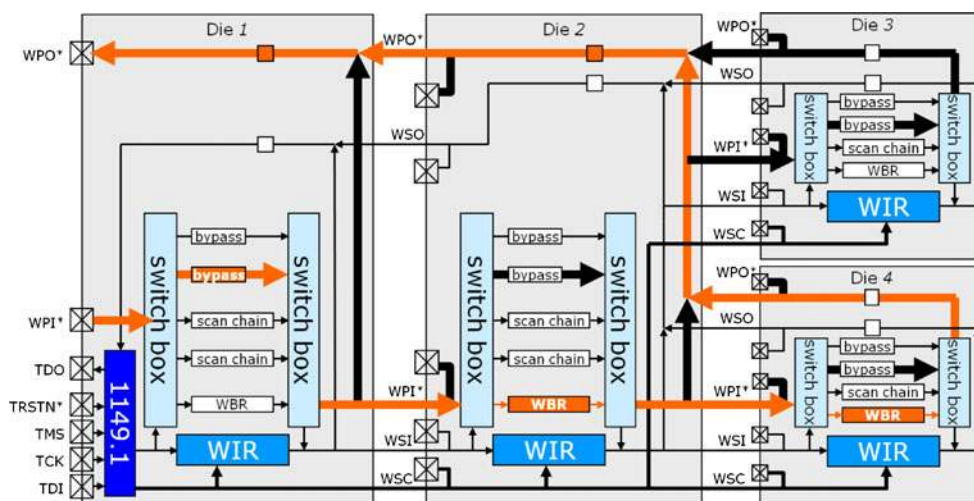


Fig. 9 ‘Railroad diagram’ for operating mode set-up for a hierarchical SOC with embedded cores and $k = 2$ towers

Fig. 10 Example of mode setting for an IEEE 1500-based 3D DfT architecture in which the TSV-based interconnect between Dies 2 and 4 is tested



Abort-on-Fail set-up to (re-)schedule short and/or likely-to-fail tests first and thus reduce the average test time [12].

Figure 10 shows the four-die example stack of Fig. 8, in which the TSV-based interconnects between Dies 2 and 4 are tested through the high-bandwidth parallel port. The orange lines in the figure highlight the activated test access path. Die 1 is in its *ParallelPostbondBypassElevator* mode; it does not actively participate in the test, but only passes the test data on to and from the dies above it. Die 2 is in its *ParallelPostbondExttestElevator4Turn3* mode; it participates in an *Exttest* and also includes Die 4 in the test data path. Die 4 is in its *ParallelPostbondExttestTurn* mode. The test mode of Die 3 is not relevant, as it does not get any test data from Die 2 below it; Die 3 could for example be in its *ParallelPostbondBypassTurn* mode.

The exact bit-level encoding of the various wrapper instructions can be different per die. It is required that the essential test instructions are implemented and that the bit-level encoding of the instruction codes is documented for the user of the dies and stack. Unused instruction codes, if any, can be mapped on the functional (non-test) mode.

We should prevent the situation in which an instruction register is in an undefined state and hence leaves its corresponding die or embedded core in an undefined mode. Upon start-up, the user is required to issue an (asynchronous) reset to bring the instruction registers in their default functional (non-test) mode. Upon loading a new instruction, the user is required to load *all* registers in the current instruction register chain with a valid instruction. The instruction registers in excluded embedded IP cores or towers are kept in their

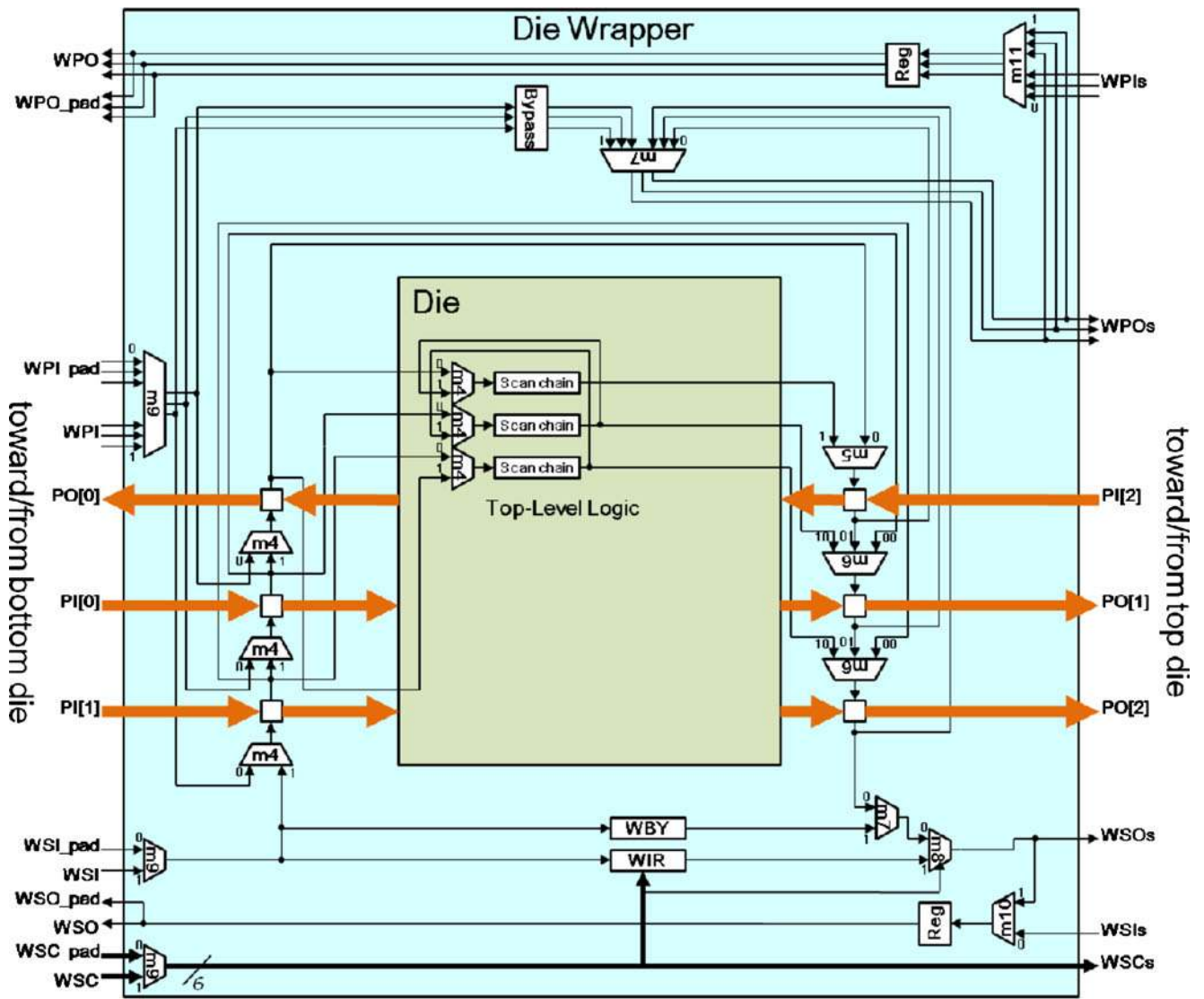
(safe) functional reset mode by means of hardware provisions [4].

6 Implementation Aspects

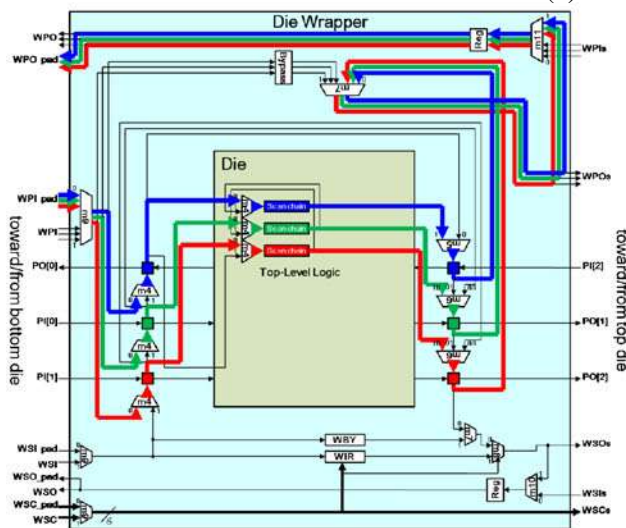
This section details several implementation aspects of our proposed 3D-enhanced die-level wrappers. We describe the 1500-based wrapper only, but the implementation aspects discussed are quite similar for the 1149.1-based wrapper. This section first considers a relatively simple case of a die which consists of one (*flat*) monolithic scan-testable logic design only and a wrapper for which the number of probe pads equals the number of TestElevator TSVs ($n = m$). Subsequently we address a more complex case, in which the die is an SOC with top-level logic and embedded cores, and a wrapper for which $n \neq m$. Both examples contain only one tower, i.e., $k = 1$.

6.1 3D Wrapper for a Flat Die

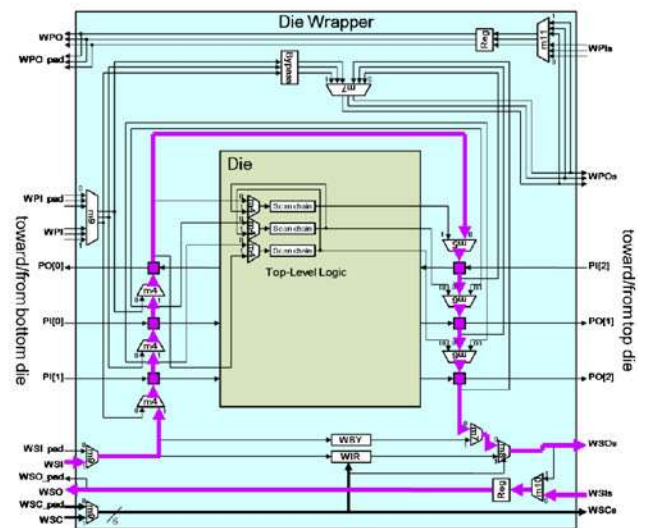
Figure 11 shows the implementation of a 3D-enhanced wrapper for a flat die. This (simplified) example die only contains flat top-level logic. It has three functional primary inputs ($pi[0..2]$) and three functional primary outputs ($po[0..2]$); some of these functional signals are (to be) connected to the die *below* this one (at the left-hand side of the figure), others are (to be) connected to the die *above* this one (at the right-hand side of the figure). In Fig. 11a, these functional I/Os are highlighted by bold orange arrows. The DfT implementation in the die consists of three internal scan chains.



(a) Functional mode



(b) ParallelPrebondIntestTurn mode



(c) SerialPostbondExtestElevator mode

Fig. 11 Implementation of a 3D-enhanced IEEE 1500 wrapper for a flat die

The 3D-enhanced die wrapper is drawn in light-blue, encapsulating the die. The wrapper contains all elements introduced in Section 5: WBR cells (shown in Fig. 11a as small white rectangles), wsc, WIR, serial port wsi-wso, serial bypass WBY, parallel port wpi-wpo, parallel bypass ('Bypass'), extra probe pads, TestElevators, and pipeline registers ('Reg'). In our example, we have chosen the parallel TestElevator and the parallel probe pad port to be of equal width, viz. $n = m = 3$.

The wrapper can be reconfigured in various operating modes, as described in Section 5.4. Each operating mode enables a different test access path through the wrapper. Two examples of such operating modes and their corresponding test access paths are shown in Fig. 11b and c. Figure 11b shows the *ParallelPrebond-IntestTurn* mode. This mode is intended for a time-efficient high-volume production test of the intra-die circuitry before stacking. The three-bit wide test access path is highlighted in the figure by means of bold red, green, and blue lines. Figure 11c shows the *SerialPostbondExtestElevator* mode. This mode is intended for a low-bandwidth test of the inter-die TSV-based connections after bonding. The single-bit test access path is highlighted in the figure by means of a bold violet line.

Reconfiguration of the wrapper into its various operating modes is done through multiplexers, which are controlled by the wsc control signals and the currently active WIR instruction. In this paper, we assign numbered names to the wrapper multiplexers: $m1, m2, m3, \dots$. Multiplexers with the same name are controlled by the same control signal.

Figure 12 shows commonly used IEEE 1500 WBR cells for respectively a (core or die) input and output [5, 22, 37]. The two wrapper cells are essentially equal, apart from their multiplexer control signals: for *Intest* and *Extest* modes, the $m2$ and $m3$ multiplexers need to be in opposite states.

The other multiplexer names are shown in Fig. 11. Multiplexers $m4 \dots m7$ select among the conventional

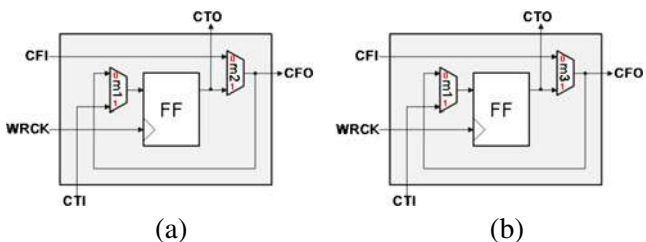


Fig. 12 A typical IEEE 1500 WBR cell: a for inputs and b for outputs

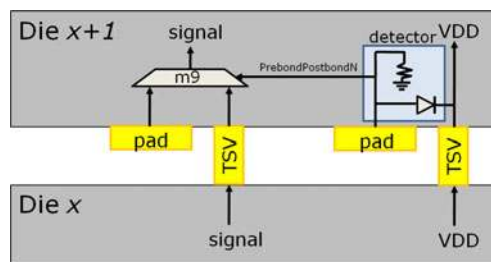


Fig. 13 Pre-bond detector circuit that generates the $m9$ multiplexer control signal

IEEE 1500 modes, including Serial/Parallel and Intest/Extest/Bypass. Multiplexer $m8$ is controlled by the selectwir signal from wsc and determines whether the serial port wsi-wso is used for loading a new instruction into the WIR or for loading test data into WBR or WBY.

New for the 3D-enhanced IEEE 1500 wrapper are multiplexers $m9, m10$, and $m11$. Multiplexers $m9$ select as I/Os between the extra probe pads on the die (*Prebond*) and the TestElevator TSVs from the die below (*Postbond*). The $m9$ control signal is the only wrapper multiplexer control signal which cannot be controlled by a WIR instruction, as also the WIR itself needs to distinguish between its pre-bond and post-bond input. Instead, it can be generated from a dedicated probe pad connected to a weak pull-down circuit. To assert the Prebond mode, the pad should be probed with value logic '1'; otherwise, it is considered to be in Postbond mode. Figure 13 shows an implementation in which the dedicated probe pad is combined with an existing power (V_{DD}) connection, in order to save the additional pad.

Multiplexers $m10$ and $m11$ select between the *Turn* and *Elevator* operating modes. Multiplexer $m10$ does this for the serial TAM, and $m11$ for the parallel TAM.

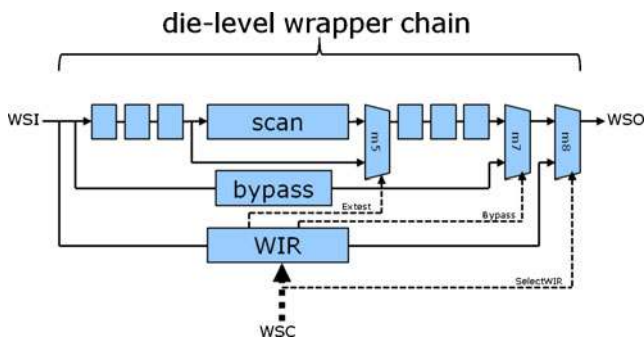
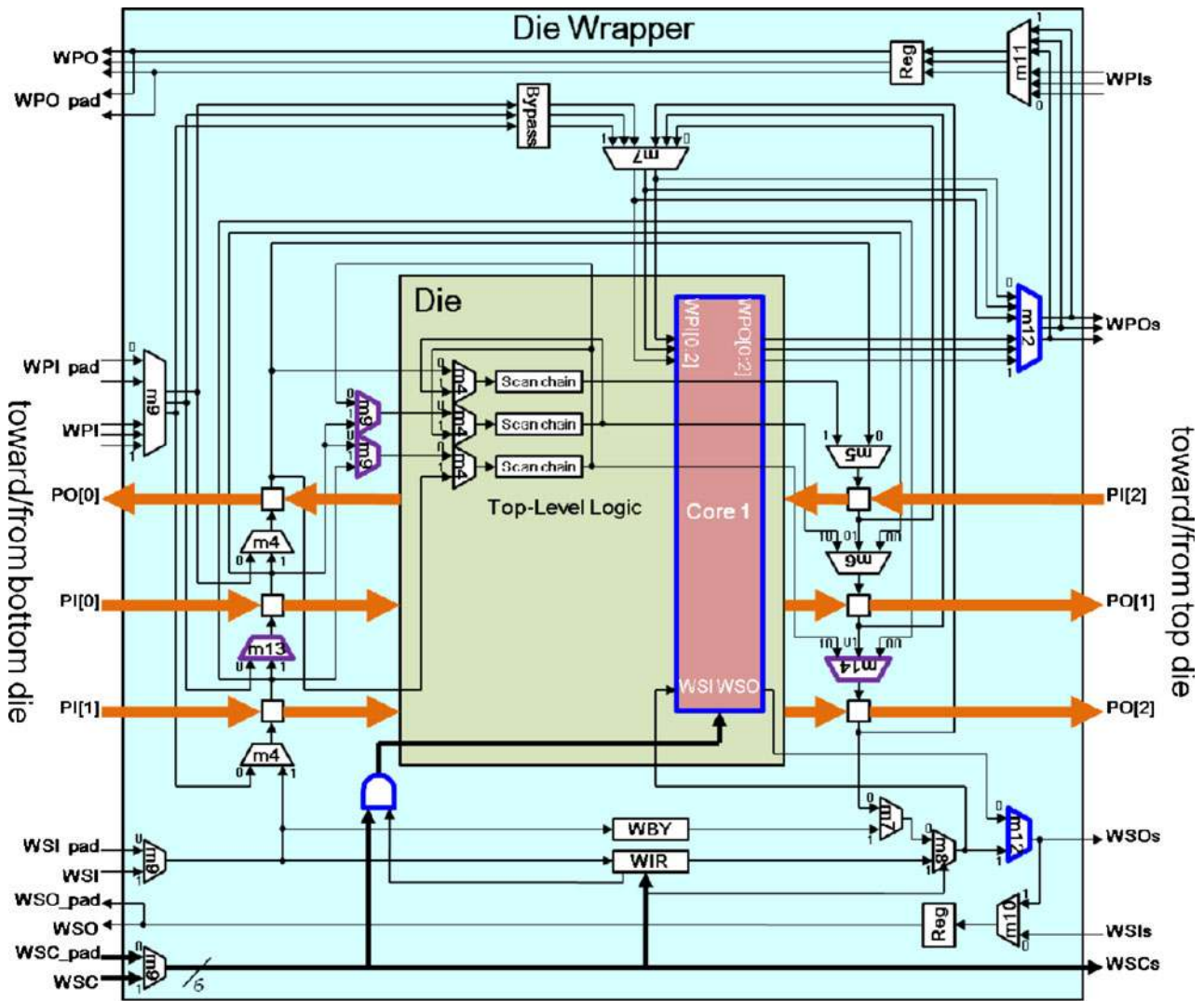
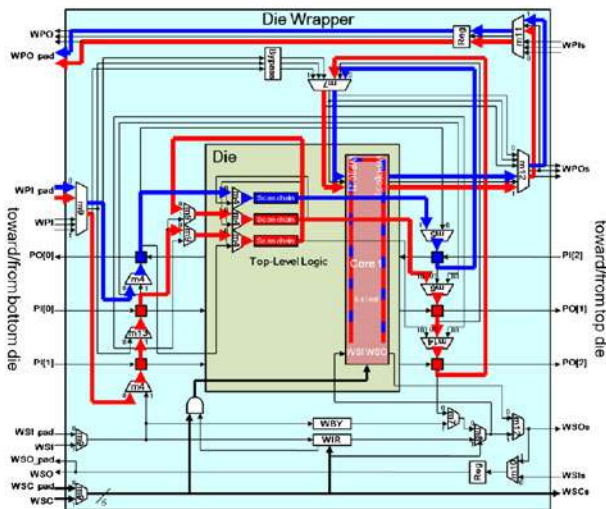


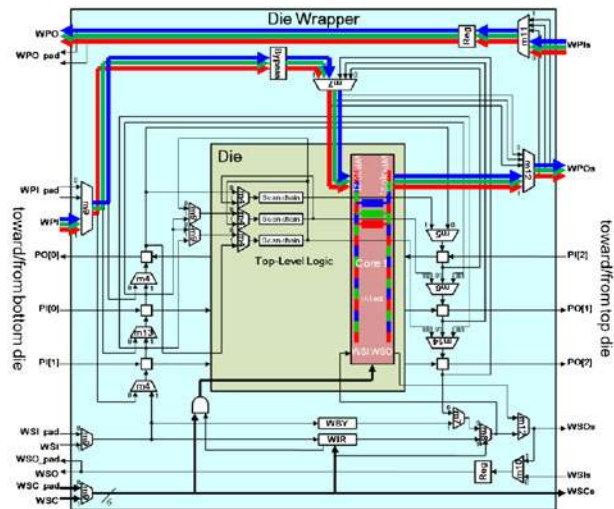
Fig. 14 Wrapper chain configurations between wsi and wso for a flat die



(a) Functional mode



(b) ParallelPrebondIntestInccoreTurn mode and Core 1 in ParallelExtest mode



(c) ParallelPostbondBypassInccoreElevator mode and Core 1 in ParallelIntest mode

Fig. 15 Implementation of a 3D-enhanced IEEE 1500 wrapper for a hierarchical die

As the serial TAM is also used for loading WIR instructions, the control input of multiplexer $m10$ is a logical AND between the WIR’s Turn/Elevator bit and the inverted $SELECTWIR$ input, such that if $SELECTWIR$ is asserted, multiplexer $m10$ is always in Elevator mode.

Figure 14 depicts the wrapper chain configuration for the serial TAM. While the parallel TAM is used for test data only, the serial TAM is used for both test data and test instructions. Instructions and data are separated by the wsc signal $SELECTWIR$; test instructions are fed into the WIR, while test data are meant for the test data access path. Bits in the test instructions determine whether the test data access path is in *Bypass* mode, *Extest* mode (test data is fed only to the wrapper boundary cells), or *Intest* mode (test data is fed to both wrapper boundary cells and die-internal scan chains). Similar test data access path reconfiguration options exist for the parallel TAM (not shown).

Table 1 shows the assignment of all multiplexer control signals for the various operating modes of the wrapper. This table is essentially the output specification of the WIR. The input specification of the WIR is given by the user-defined instruction codes for each of the operating modes.

6.2 3D Wrapper for a Hierarchical Die

In this section, we consider the implementation details for a slightly more complex case, in which (1) the wrapper has different widths for parallel probe pad ports and parallel TestElevator ports (i.e., $n \neq m$), and (2) the die is a core-based SOC with top-level logic and embedded cores. Figure 15 shows the implementation of a 3D-enhanced wrapper for this case. The figure is in the same style as Fig. 11; the differences required to support (1) and (2) are highlighted by means of purple and blue outlines, respectively.

In this example, our 3D-enhanced wrapper has different pre-bond and post-bond parallel port widths, viz. $n = 3$ and $m = 2$. As shown in Fig. 15a by means of purple outlines, this requires two extra $m9$ multiplexers as well as two new multiplexers $m12$ and $m13$ to switch between pre-bond parallel test modes (with $m = 2$) and post-bond parallel test modes (with $n = 3$).

The example die has one embedded core, named Core 1; in our simplified example, the single Core 1 actually represents a possibly larger number of embedded cores. Core 1 is wrapped with a conventional IEEE 1500 wrapper (not shown) with a parallel port WPI-WPO of three bits wide. The example TAM architecture in our example SOC is a Daisychain Architecture [7, 37].

The serial and parallel TAMs of the embedded core(s) are included at the tail end of the die-level wrapper chains. Multiplexers $m11$, controlled by the *Coreen/Coredis* bit of the WIR instructions, determine whether or not the core-level TAMs are bypassed. Figure 16 shows the wrapper chain configuration for the serial TAM; a similar configuration exists for the optional parallel TAM. The figure shows in blue the wrapper chain configuration logic in the die-level wrapper and in green the wrapper chain configuration logic in the core-level wrapper(s). Note that this design setup requires access from the die wrapper to the head and tail ends of the core-level TAM(s). The *Coredis* control signal allows to bypass all embedded cores of this die. In order to guarantee that the core-level WIR(s) are in a well-defined safe state, two things are required: (1) each test starts with a reset on $WRSTN$, which should bring the WIR(s) in their (safe) functional start-up state [5], and (2) at the core-level, either the $WRCK$ or $WRSTN$ signals are AND-gated with the *Coredis* control signal, which keeps the core-level WIR(s) in their start-up state.

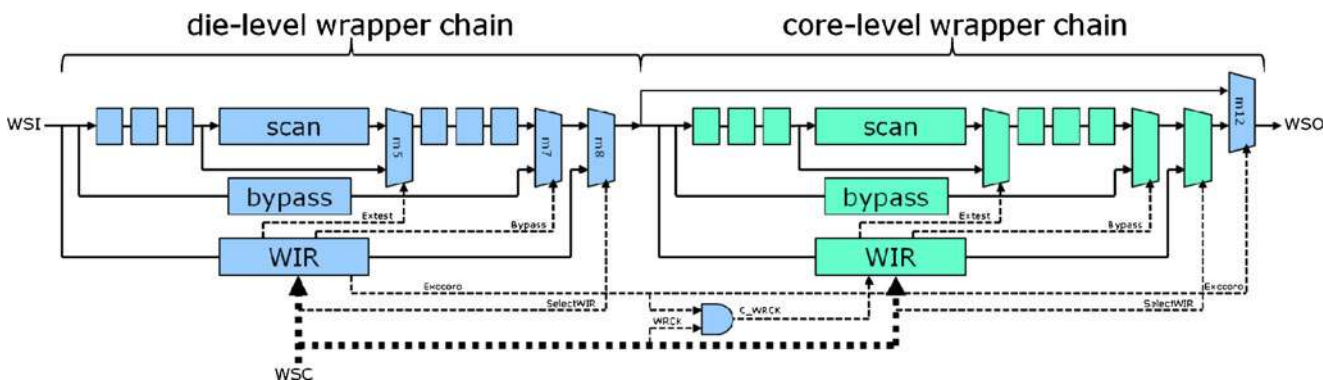
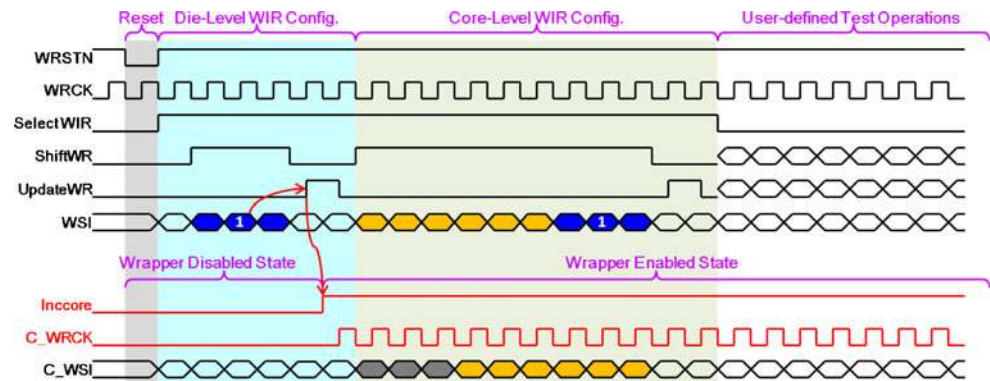


Fig. 16 Wrapper chain configurations between wsi and wso for a hierarchical SOC die containing an embedded core

Fig. 17 WIR instruction sequence: (1) reset, (2) die-level WIR configuration, (3) die- and core-level WIR configuration



In the hierarchical set-up with bypassable embedded cores, we distinguish three types of operations: a $WRSTN$ reset, followed by resp. zero, one, or two instructions loads, as shown in Fig. 17. A single $WRSTN$ reset is sufficient to jump-start all WIRs into their functional (non-test) mode [5]. The wrapper chain is reset to its shortest length through the dies only, i.e., bypassing any embedded cores. To enter a test mode, it is sufficient to subsequently load the appropriate instructions in all die-level WIRs. If we want to enable one or more cores, the corresponding die-level WIR instructions need to assert their *Coreen* signals. The hierarchical TAM will then be reconfigured to include the cores of the corresponding dies. Subsequently, the longer WIR chain will need to be reprogrammed with instructions for all die-level WIRs and the selected core-level WIRs. Note that one can flexibly re-order tests without explicitly keeping track of the WIR chain length in the previous test, provided all tests start with a reset pre-amble.

For this example die, Fig. 15b and c show two operating mode examples and their corresponding test access paths. Figure 15b shows a mode in which the die's top-level logic is tested. The die wrapper is in its *ParallelPrebondIntestCoreenTurn* mode. Note that this test requires the IEEE 1500 wrapper of embedded Core 1 to participate in its *ParallelExtest* (WP_EXTEST) mode, as the inputs and outputs of Core 1 actually are outputs resp. inputs of the die's top-level logic. Also note that, although the die and its embedded core support a test path width of three bits, in this pre-bond test mode only two input and output pads are provided ($m = 2$). Consequently, we are forced to assign the three internal test paths to two external pads, as highlighted in the figure by means of bold red and blue lines.

Figure 15c shows a mode in which Core 1 is tested. The die wrapper is in its *ParallelPostbondBy-*

passCoreenElevator mode. The die's top-level logic is bypassed, and Core 1 is in its *ParallelIntest* (WP_INTEST) mode. This example is a post-bond test mode ($n = 3$), and the test data paths are highlighted by means of bold red, green, and blue lines.

7 Experimental Results

The implementation costs for the 3D die wrapper are threefold: (1) additional TSVs, (2) additional probe pads, and (3) additional logic gates. For the IEEE 1500-based wrapper, the additional TSV count is $6 + 2 + 2n$ (with $n \geq 0$) for respectively the wsc, wsi-wso serial port, and wpi-wpo parallel port; for the IEEE 1149.1-based wrapper, this number changes to $2 + 2 + 2n$. For the IEEE 1500-based wrapper, the additional probe pad count is $6 + 2 + 2m$ (with $m \geq 0$) for respectively the wsc, wsi-wso serial port, and wpi-wpo parallel port; for the IEEE 1149.1-based wrapper, this number changes to $2 + 2 + 2m$. These numbers exclude TSVs and pads for infrastructure like power, ground, and clocks.

The area costs of the additional logic gates consist of five components.

- A *fixed* cost, f_c , which consists of WIR, WBY, and some of the configuration MUXes.
- A *variable* cost, which is the product of the number of functional I/Os i of the die and the area cost i_c per functional I/O. This category consists of the Wrapper Boundary Register (WBR) cells.
- A *variable* cost, which is the product of the die-internal TAM width n and the area cost n_c per TAM wire. This category consists of the MUXes for scan chain concatenation.
- A *variable* cost, which is the product of the number of towers k and the area cost k_c per tower. This

Table 2 Area costs of the proposed 3D die wrapper in Faraday/UMC 90nm technology

Circuit name	Die specification			Parameters			Wrapper Area A_w			
	#Inputs	#Outputs	Area A (μm^2)	i	n	k	Actual (μm^2)	Estimated (μm^2)	Overhead (A_w/A)	
Equation calibration										
s400	[3]	3	6	1,014	9	3	1	902	892	+88.955%
		3	6	1,014	9	3	2	960	962	+94.675%
		3	6	1,014	9	3	3	1,029	1,032	+104.479%
s1423	[3]	17	5	3,718	22	3	1	1,359	1,361	+36.552%
		17	5	3,718	22	3	2	1,429	1,431	+38.435%
		17	5	3,718	22	3	3	1,499	1,501	+40.317%
s5378	[3]	35	49	11,468	84	3	1	3,596	3,600	+31.357%
		35	49	11,468	84	3	2	3,666	3,669	+31.967%
		35	49	11,468	84	3	3	3,736	3,739	+32.578%
Results										
PNX8550	[8]	n.a.	n.a.	~50M	~280	140	1	n.a.	19,388	+0.039%
		n.a.	n.a.	~50M	~280	140	2	n.a.	20,431	+0.041%
		n.a.	n.a.	~50M	~280	140	3	n.a.	21,474	+0.043%
TestDesign1		n.a.	n.a.	80M	10,000	140	3	n.a.	372,366	+0.465%

category consists of the MUXes for the selection of each tower.

- A *variable* cost, which is the product of the die-internal TAM width n , the number of towers k , and the area cost nk_c per TAM wire per tower. This category consists of the configuration MUXes for the daisychain TAM.

Combining the above listed terms, the area cost A_w can be estimated by the following equation.

$$A_w = f_c + (i \times i_c) + (n \times n_c) + (k \times k_c) + (n \times k \times nk_c) \tag{7.1}$$

where f_c , i_c , n_c , k_c and nk_c are technology-dependent area costs, and i , n , and k are design-dependent parameters, representing the number of functional I/Os, die-internal TAM width, and number of towers respectively.

In order to verify our proposed 3D-enhanced wrapper design and assess its implementation costs, we have set up a prototype tool flow that adds a 3D wrapper to a die design. The tool flow starts with the gate-level netlist of a die design, including its conventional internal DfT features. Subsequently, we use a commercial EDA tool to add a conventional test wrapper to the die. We manually modify the 2D wrapper into a 3D-enhanced wrapper, as there is no commercial tool support for that available yet. Next, we are able to assess the impact on the design size by reporting the gate area costs. Finally, we verify our design by generating test patterns with a commercial ATPG tool and simulating the resulting test sets.

In order to calibrate Eq. 7.1, we have applied the tool flow described above using the Faraday/UMC 90nm CMOS standard cell library to three ISCAS'89 benchmark circuits s400, s1423, and s5378 [3], posing as to-be-wrapped dies; the area results are listed in Table 2. Column 1 lists the circuit names, and Columns 2 to 4 present key circuit specifications, including die area A . Columns 5 to 7 list design-specific parameters i , n , and k . Column 8 shows the wrapper area A_w as obtained in the actual gate-level implementation, while Column 10 shows the overhead ratio A_w/A . The wrapper area costs for these three benchmark circuits are rather high, since the dies considered are unrealistically small design, and hence we have grayed them out in Table 2.

By analyzing the wrapper implementations for the three ISCAS benchmark circuits, we extract the technology-dependent parameters in Eq. 7.1 as follows: $f_c = 327.7\mu\text{m}^2$, $i_c = 36.1\mu\text{m}^2$, $n_c = 56.5\mu\text{m}^2$, $k_c = 48.6\mu\text{m}^2$ and $nk_c = 7.1\mu\text{m}^2$. Column 9 of Table 2 shows the wrapper area estimated by Eq. 7.1 using these parameter values. The results demonstrate the accuracy of the equation, since the estimated wrapper area is very close to the actual one. Therefore, we can use Eq. 7.1 to estimate the area costs of the proposed multi-tower die-level wrapper on other, more complex designs, provided that the three design-dependent parameters i , n , and k are available.

We apply Eq. 7.1 to published data of industrial SOC PNX8550 [8]. The result shows that, with three towers on top of PNX8550, the die wrapper area overhead is only 0.043%, which is a negligible amount. TSVs hold the promise to offer much a larger number of inter-

die interconnects. Hence, we also apply Eq. 7.1 to a hypothetical design, *TestDesign1*, having 10,000 I/Os. The wrapper area overhead in such an I/O-rich design is 0.465%, which is still a small fraction. From the area results above, we see that the proposed multi-tower wrapper is low-cost under different design parameter settings.

8 Conclusion

In this paper, we presented a generic Design-for-Test architecture for TSV-based 3D-SICs. The main component of our 3D DfT architecture is a die-level wrapper. The paper describes two alternative wrappers, one based on an extended version of IEEE 1149.1, the other based on an extended version of IEEE 1500. Both wrappers have the following key features: (1) a serial (one-bit) and scalable parallel (n -bit) test access mechanism, (2) *TestTurns* from and to the stack's external I/Os (typically located in the bottom die), (3) *TestElevators* that carry test data up and down through the stack in post-bond testing, (4) optional additional probe pads for all non-bottom dies allowing for pre-bond testing, and (5) an optional hierarchical inclusion/exclusion mechanism for embedded IP cores, if any, and dies in higher-level towers, that prevents unbridled growth of test lengths. The main difference between the IEEE 1149.1- and IEEE 1500-based die wrappers is in the width of the broadcast control buses (two or three vs. six or seven wires), the on-die TAP Controller (present vs. absent), and the support for existing debug and emulation set-ups (present vs. absent).

The architecture leverages existing intra-die DfT features such as internal scan, test data compression, built-in self-test, and core-based wrappers and TAMs, as well as boundary scan at the 3D-SIC's PCB interface, and requires no additional product-level pins. The architecture services the test needs for die maker(s), stack maker, and stack user alike, by providing support for (1) pre-bond die testing, (2) mid-bond testing for partial stacks, (3) post-bond testing for complete stacks, (4) board-level interconnect testing, and (5) (low-bandwidth) in-field test and debug. The architecture supports a modular test approach, in which dies and their embedded cores, as well as inter-die interconnects, can be tested separately. The architecture provides maximum freedom with respect to inclusion or exclusion of certain tests at a particular stage of the test flow and allows for flexible (re-)scheduling of those tests, in order to optimize the test flow and minimize the associated test costs. We have shown that the imple-

mentation costs for medium and large industrial SOCs are negligible.

The proposed architecture is *structured*, as it provides a common DfT template that meets all 3D-SIC test access requirements. The proposed architecture is also *scalable*, in the sense that it works for all stack heights and multi-tower stacks, and provides user-defined test access bandwidth; the latter provides a trade-off opportunity between silicon area and test length. Consequently, the architecture is a great starting point for future standardization and automation in EDA tool flows for DfT insertion and test expansion.

Open Access This article is distributed under the terms of the Creative Commons Attribution Noncommercial License which permits any noncommercial use, distribution, and reproduction in any medium, provided the original author(s) and source are credited.

References

- Banerjee K et al (2001) 3-D ICs: a novel chip design for improving deep-submicrometer interconnect performance and systems-on-chip integration. *Proc IEEE* 89(5):602–633
- Beyne E, Swinnen B (2007) 3D system integration technologies. In: *Proceedings of IEEE International Conference on Integrated Circuit Design and Technology (ICICDT)*, pp 1–3
- Brglez F, Bryan D, Kozmiński K (1989) Combinational profiles of sequential benchmark circuits. In: *Proceedings International Symposium on Circuits and Systems (ISCAS)*, pp 1924–1934
- Chi C-C et al (2011) DfT architecture for 3D-SICs with multiple towers. In: *Proceedings IEEE European Test Symposium (ETS)*, pp 51–56
- da Silva F, McLaurin T, Waayers T (2006) *The core test wrapper handbook—rationale and application of IEEE Std. 1500™*, vol 35 of *frontiers in electronics testing*. Springer-Verlag, Boston
- Garrou P, Bower C, Ramm P (eds) (2008) *Handbook of 3D integration—technology and applications of 3D integrated circuits*. Wiley-VCH, Weinheim
- Goel SK, Marinissen EJ (2003) SOC test architecture design for efficient utilization of test bandwidth. *ACM Transact Des Automat Electron Syst* 8(4):399–429
- Goel SK et al (2004) Test infrastructure design for the Nexperia™ home platform PNX8550 system chip. In: *Proceedings Design, Automation, and Test in Europe (DATE) designers forum*, pp 108–113, Paris, France
- IEEE Computer Society (2001) *IEEE Std 1149.1™-2001, IEEE standard test access port and boundary-scan architecture*. IEEE, New York
- IEEE Computer Society (2003) *IEEE Std 1532™-2002, IEEE standard for in-system configuration of programmable devices*. IEEE, New York
- IEEE Computer Society (2005) *IEEE Std 1500™-2005, IEEE standard testability method for embedded core-based integrated circuits*. IEEE, New York

12. Ingelsson U et al (2005) Test scheduling for modular SOCs in an abort-on-fail environment. In: Proceedings IEEE European Test Symposium (ETS). Tallinn, Estonia, pp 8–13
13. Jiang L, Huang L, Xu Q (2009) Test architecture design and optimization for three-dimensional SoCs. In: Proceedings Design, Automation, and Test in Europe (DATE), pp 220–225
14. Jiang L et al (2009) Layout-driven test-architecture design and optimization for 3D SoCs under pre-bond test-pin-count constraint. In: Proceedings International Conference on Computer-Aided Design (ICCAD), pp 191–196
15. Jung D-Y, Kwak S-H, Lee M-K (2002) Reusable embedded debugger for 32-bit RISC processor using the JTAG boundary scan architecture. In: IEEE Asia-Pacific Conference on ASIC, pp 209–212
16. Lee H-HS, Chakrabarty K (2009) Test challenges for 3D integrated circuits. *IEEE Des Test Comput* 26(5):26–35
17. Lewis DL, Lee H-HS (2007) A scan-island based design enabling prebond testability in die-stacked microprocessors. In: Proceedings IEEE International Test Conference (ITC), Paper 21.2
18. Lewis DL, Lee H-HS (2009) Testing circuitpartitioned 3D IC designs. In: Proceedings of IEEE computer society annual symposium on VLSI (ISVLSI), pp 139–144
19. Lo C-Y et al (2010) SOC test architecture and method for 3-D ICs. *IEEE Trans Comput-Aided Des* 29(10):1645–1649
20. Loh GH, Xie Y, Black B (2007) Processor design in 3D die-stacking technologies. *IEEE Micro* 27(3):31–48
21. Marinissen EJ (2010) Challenges in testing TSV-based 3D stacked ICs: test flows, test contents, and test access. In: Proceedings IEEE Asia-Pacific conference on circuits and systems, pp 544–547
22. Marinissen EJ, Zorian Y (2009) IEEE 1500 enables modular SOC testing. *IEEE Des Test Comput* 26(1):8–16
23. Marinissen EJ, Zorian Y (2009) Testing 3D chips containing through-silicon vias. In: Proceedings IEEE International Test Conference (ITC), Paper ET1.1.
24. Marinissen EJ et al (1998) A structured and scalable mechanism for test access to embedded reusable cores. In: Proceedings IEEE International Test Conference (ITC), pp 284–293, Washington, DC
25. Marinissen EJ et al (2003) Minimizing pattern count for interconnect tests under a ground-bounce constraint. *IEEE Des Test Comput* 20(2):8–18
26. Marinissen EJ, Verbree J, Konijnenburg M (2010) A structured and scalable test access architecture for TSV-based 3D stacked ICs. In: Proceedings IEEE VLSI Test Symposium (VTS), pp 269–274
27. Olmen JV et al (2008) 3D stacked IC demonstration using a through silicon via first approach. In: Proceedings IEEE International Electron Devices Meeting (IEDM), pp 1–4
28. Parker KP (2003) *The boundary-scan handbook*, 3rd edn. Springer-Verlag
29. Patti RS (2006) Three-dimensional integrated circuits and the future of system-on-chip designs. *Proc IEEE* 94(6):1214–1224
30. Posse K et al (2006) IEEE P1687: toward standardized access of embedded instrumentation. In: Proceedings IEEE International Test Conference (ITC), pp 1–8
31. Smith K et al (2010) KGD probing of TSVs at 40 μ m array pitch. In: Digest of IEEE international workshop on testing three-dimensional stacked integrated circuits (3D-TEST), Paper 4.1
32. Swinnen B et al (2006) 3D integration by Cu-Cu thermo-compression bonding of extremely thinned bulk-si die containing 10 μ m pitch through-si vias. In: Proceedings IEEE International Electron Devices Meeting (IEDM), pp 1–4
33. Taouil M et al (2010) Test cost analysis for 3D die-to-wafer stacking. In: Proceedings IEEE Asian Test Symposium (ATS), pp 435–441
34. Varma P, Bhatia S (1998) A structured test re-use methodology for core-based system chips. In: Proceedings IEEE International Test Conference (ITC), pp 294–302, Washington, DC
35. Velenis D, Marinissen EJ, Beyne E (2010) Cost effectiveness of 3D integration options. In: Proceedings IEEE International Conference on 3D system integration (3DIC)
36. Vermeulen B et al (2007) Communication-centric SoC debug using transactions. In: Proceedings IEEE European Test Symposium (ETS), pp 69–76
37. Waayers T, Morren R, Grandi R (2005) Definition of a robust modular SOC test architecture; resurrection of the single TAM daisy-chain. In: Proceedings IEEE International Test Conference (ITC). Austin, TX
38. Weerasekera R et al (2007) Extending systems-on-chip to the third dimension: performance, cost and technological tradeoffs. In: Proceedings International Conference on Computer-Aided Design (ICCAD), pp 212–219
39. Winters M (1994) Using IEEE-1149.1 for in-circuit emulation. In: WESCON 'idea/microelectronics' conference, pp 525–528
40. Wu X, Falkenstern P, Xie Y (2007) Scan chain design for three-dimensional integrated circuits (3D ICs). In: Proceedings International Conference on Computer Design (ICCD), pp 208–214
41. Wu X et al (2008) Test-access mechanism optimization for core-based three-dimensional SOCs. In: Proceedings International Conference on Computer Design (ICCD), pp 212–218
42. Zadegan FG et al (2011) Design automation for IEEE P1687. In: Proceedings Design, Automation, and Test in Europe (DATE), pp 1412–1417

Erik Jan Marinissen is Principal Scientist at IMEC vzw in Leuven, Belgium. Previously, he worked at NXP Semiconductors and Philips Research, both in Eindhoven, The Netherlands. Marinissen holds an MSc degree in Computing Science (1990) and a PDEng degree in Software Technology (1992), both from Eindhoven University of Technology. Marinissen's research interests include all topics in the domain of test and debug of micro-electronics. He is co-author of over 160 journal and conference papers and co-inventor on nine granted US and EP patent families. Marinissen is recipient of the ITC 2008 and ITC 2010 Most Significant Paper Awards and Best Paper Awards at the Chrysler-Delco-Ford Automotive Electronics Reliability Workshop 1995 and the IEEE International Board Test Workshop 2002. He served as Editor-in-Chief of IEEE Std. 1500. He is a founder of workshops on 'Diagnostic Services in Network-on-Chips' (DSNOC), '3D Integration', and 'Testing of Three-Dimensional Stacked Integrated Circuits' (3D-TEST). He serves on numerous conference committees, including ATS, DATE, ETS, ITC, and VTS, and on the editorial boards of IEEE Design & Test of Computers, IET Computers and Digital Techniques, and Springer's Journal of Electronic Testing: Theory and Applications (JETTA). Marinissen is a Fellow of IEEE and Golden Core Member of Computer Society.

Chun-Chuan Chi is a PhD student at National Tsing-Hua University in Hsinchu, Taiwan. In 2010–2011, Chi spent nine-months as student intern at IMEC in Leuven, Belgium, working on 3D-DfT architectures. Chi holds a BSEE degree from National Central University in Jhongli, Taiwan. Chi's research interests include VLSI testing in general and 3D-SIC design and test in particular. Chi is co-author on six conference papers.

Mario Konijnenburg received the MSc degree in Electrical Engineering from Delft University of Technology in The Netherlands in 1993. The PhD degree was received from Delft University of Technology in The Netherlands in 1999 on Automatic Test Pattern Generation for Sequential Circuits. He joined Philips Research/NXP Semiconductors and worked on methodology- and tool- development to improve testability of designs and ease

Design-for-Test. Currently he is system architect and R&D engineer at Holst Centre/IMEC in Eindhoven, The Netherlands, where he works on development of ultra low-power (wireless sensor node) designs for mainly bio-medical applications. His current research interests are design for low-power and design-for-test in DSPs.

Jouke Verbree is a student at Delft University of Technology, Delft, The Netherlands. Verbree holds a Bachelor degree in Computer Science and a Master degree in Computer Engineering, both from Delft University of Technology. The master graduation project has been performed at IMEC in Leuven, Belgium. General research interests reside at computer chip design, and with 3D-SICs and chip testing in particular. Verbree is co-author on four conference papers.