EINDHOVEN UNIVERSITY OF TECHNOLOGY

IEEE Std P1838

Citation for published version (APA):
Marinissen, E. J., McLaurin, T., & Jiao, H. (2016). IEEE Std P1838: DfT standard-under-development for 2.5D-, 3D-, and 5.5D-SICs. In *Proceedings 2016 21st IEEE European Test Symposium, ETS 2016, 23-26 May 2016, Amsterdam, The Netherlands* (pp. 1-10). [7519330] Piscataway: Institute of Electrical and Electronics Engineers. https://doi.org/10.1109/ETS.2016.7519330

DOI:
10.1109/ETS.2016.7519330

Document status and date:
Published: 22/07/2016

Document Version:
Accepted manuscript including changes made at the peer-review stage

Please check the document version of this publication:

• A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
• The final author version and the galley proof are versions of the publication after peer review.
• The final published version features the final layout of the paper including the volume, issue and page numbers.

Link to publication

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

• Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
• You may not further distribute the material or use it for any profit-making activity or commercial gain
• You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

Take down policy
If you believe that this document breaches copyright please contact us at:

openaccess@tue.nl

providing details and we will investigate your claim.

Download date: 31. May. 2020

IEEE Std P1838: DfT Standard-under-Development for 2.5D-, 3D-, and 5.5D-SICs

Erik Jan Marinissen[1,3] Teresa McLaurin[2] Hailong Jiao[3,1]

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

[2] ARM
5707 Southwest Parkway #100
Austin, TX 78735
United States of America
teresa.mclaurin@arm.com

[3] Eindhoven University of Technology
Den Dolech 2
5612AZ Eindhoven
The Netherlands
h.jiao@tue.nl

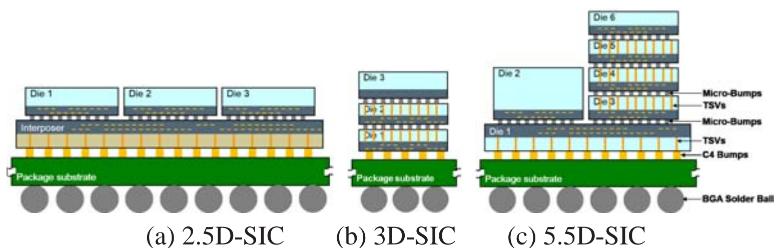
Abstract

For stacked integrated circuits, effective test access requires the design-for-test (DfT) features in the various dies to operate in a concerted way to transport test stimuli and responses from and to the external I/Os up and down through the stack. This 3D-DfT can be proprietary if all dies in the stack are made by a single company. However, in the likely case that the various dies in the stack originate from different companies, standardized 3D-DfT is required to guarantee inter-operability. IEEE Std P1838 is a standard-under-development that addresses exactly this issue. This paper presents a status report of P1838 and describes its three main hardware components: a serial control mechanism, a die wrapper register, and a flexible parallel port.

1 Introduction

Technology advances with respect to micro-bump inter-die interconnects, through-silicon vias and associated wafer thinning, and passive interposers enable a new generation of vertically-stacked ICs. We distinguish a number of popular die stack architectures.

- *2.5D-SIC*: Multiple active dies are placed side-by-side on top of and interconnected through an interposer base die [1, 2] (see Figure 1(a)).
- *3D-SIC*: Multiple active dies are placed on top of each other in a single tower [3] (see Figure 1(b)).
- *5.5D-SIC*: Multiple towers, each consisting of one or more stacked active dies, are placed side-by-side on top of and interconnected through an interposer base die [4] or an active die (see Figure 1(c)).



(a) 2.5D-SIC (b) 3D-SIC (c) 5.5D-SIC

Figure 1: Various commonly-encountered die-stack architectures.

These new IC product compositions offer various compelling benefits: (1) heterogeneous integration, i.e., the ability to use the most efficient technology node for each die in the stack, for example optimized for digital logic, memory, or analog circuitry; (2) inter-die communication with high bandwidth, low latency, and low power consumption; and (3) higher component yields (and hence: lower costs per component) in case an originally monolithic, large die is partitioned over multiple smaller dies that are stacked on top of each other – provided faulty dies can be adequately removed from the manufacturing process before stacking by means of a so-called *pre-bond* test.

Like all micro-electronic products, these die stacks need to be tested before they can be shipped with acceptable quality levels to their customers. We distinguish the following tests [5]: (1) *pre-bond tests* prior to stacking, (2) *mid-bond tests* on incomplete, partial stacks, (3) *post-bond tests* on complete yet still not packaged stacks, and (4) *final tests* on the final packaged product. The number of possible test flows grows quickly with the number of dies in the stack [6] and hence is subject of automated trade-off evaluation and optimization [7].

A well-architected design-for-test (DfT) test access infrastructure is indispensable for achieving a high-quality test. Not only do we need conventional ('2D') DfT structures (such as internal scan chains, test data compression circuitry, IEEE Std 1500 wrappers around embedded cores, and built-in self-test (BIST) engines) that provide test access within a single die. We also need novel '3D' DfT structures that provide modular test access from (and to) the external stack I/Os to (and from) the various dies and inter-die interconnect levels, thereby transporting test stimuli and responses up and down through other dies on the way. Several ad-hoc 3D-DfT architectures have been proposed, based on IEEE Std 1149.1 [8] [9], IEEE Std 1500 [10] [11–14], and on IEEE Std 1687 [15] [16]. These architectures all have their specific strong and weak points. However, their underlying 3D-DfT architectures do not inter-operate together. Hence, there is a need for a per-die 3D-DfT standard, such that if compliant dies are brought together in a die stack, a basic minimum of test access features are guaranteed to work across the stack. IEEE Std P1838 is such a standard; currently still under development.

978-1-4673-9659-2/16/$31.00 ©2016 IEEE

The rest of this paper describes IEEE Std P1838. Section 2 briefly describes the history and current status of the standard. Section 3 defines the scope and some key terms that are used in the draft standard. Subsequently, Sections 4, 5, and 6 describe the three main hardware components of IEEE Std P1838: its serial control mechanism, die wrapper register (DWR), and the flexible parallel port (FPP). Section 7 concludes this paper.

2 History and Status

During 2010, the IEEE, via its Test Technology Standards Committee (TTSC), sponsored a Study Group to investigate if the field of 3D testing provided opportunities for new IEEE standard(s). This group of over 60 interested persons, led by Erik Jan Marinissen, determined that a 3D-DfT standard was required. The Study Group formulated a Project Authorization Request (PAR) entitled *'Standard for Test Access Architecture for Three-Dimensional Stacked Integrated Circuits'* that was submitted to the IEEE Standards Association (IEEE-SA) on November 23, 2010.

On February 2, 2011, IEEE-SA approved the PAR with sequence number P1838, and from that moment on the Study Group transitioned into a Working Group. In the IEEE organization, the Working Group reports to the TTSC; TTSC in turn reports both to IEEE-SA as well as to the Test Technology Technical Council (TTTC) of IEEE's Computer Society. In the P1838 Working Group, Erik Jan Marinissen and Adam Cron were elected as Chair and Vice-Chair respectively, while Michael Wahl serves as Editor-in-Chief for the draft standard. Today, the Working Group has around 50 active members and an additional list of 50+ passive followers. The PAR, originally valid for five years, has recently been extended by IEEE-SA to the end of 2018; standardization with a large group of volunteers is inherently a time-consuming process! The Working Group has been meeting in first weekly and later biweekly one-hour meetings via conference call facilities kindly provided by Cisco Systems.

The first three years of the Working Group's existence were spent on education of the team and definition of the scope of the standard and associated terminology (see Section 3). Three hardware components were identified and corresponding sub-groups ('tiger teams') were started to work out the details in parallel.

1. A (one-bit) serial control mechanism, based on the IEEE Std 1149.1 Test Access Port (TAP) [8], for configuration of and low-bandwidth test data access to the DfT resources of this die and dies further up in the stack (see Section 4).

2. A die wrapper register (DWR), based on IEEE Std 1500 [10], consisting of wrapper cells at the die boundary that provide test controllability and observability and hence enable a modular test approach by supporting inward-facing (INTEST) and outward-facing (EXTEST) test modes (see Section 5).

3. A flexible parallel port (FPP), a new native P1838 development, which provides optional n-bit (with $n \geq 0$ user-defined) test access to the DfT resources of this die and dies further up in the stack (see Section 6).

These three tiger teams have largely defined their hardware architecture and are currently in the process of capturing their ideas in standards' language: Rules, Recommendations, and Permissions. The Working Group at large is reviewing the draft texts produced by the three tiger teams, and at the same time discussing the requirements for a formal language that can serve as a P1838 specification and implementation description language.

3 Scope and Terminology

The aim of IEEE Std P1838 is to define a standardized and scalable generic test access architecture to and between dies in a multi-die stack, especially stacks with TSV-based interconnects such as 2.5D-, 3D-, and 5.5D-SICs. The focus of the standard is on testing the intra-die circuitry as well as the inter-die interconnects in pre-bond, mid-bond, and post-bond cases in pre-packaging, post-packaging, and board-level situations.

The standard is die-centric, i.e., compliance to the standard pertains to a die (and *not* to a stack of dies). Standardized die-level DfT features comprise a stack-level test access architecture. In this way, the standard enables interoperability between die maker and stack maker. The standard does not address stack-level challenges and solutions. The most prominent example of this is that the standard does not address compliance of the stack to IEEE Std 1149.1 Boundary Scan [8] for board-level interconnect testing (although the standard should certainly not prohibit application thereof).

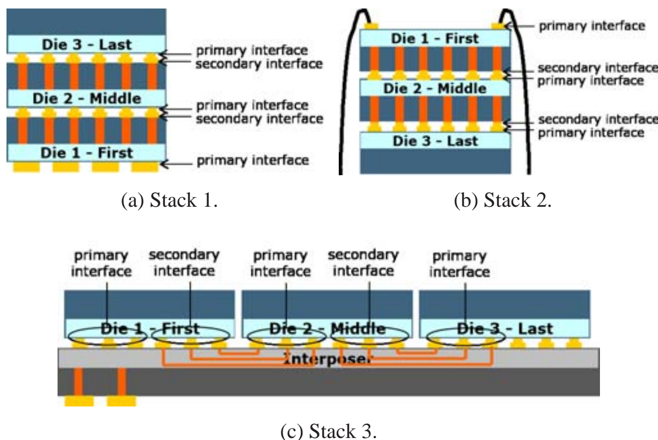
P1838 aims to standardize (1) mandatory and optional on-chip hardware components and (2) a formal language in which implementation choices could be specified and described. The on-chip (3D-)DfT hardware is based on and works with digital scan-based test access. The P1838 Working Group aims to leverage existing (2D-)DfT wherever applicable and appropriate, including test access ports (such as IEEE Std 1149.1 [8]), on-chip DfT such as internal scan chains and wrappers of embedded cores (as in IEEE Std 1500 [10]), and on-chip design-for-debug and embedded instruments (such as in IEEE Std 1687 [15]). The standard does not mandate specific defect or fault models, specific test generation methods, nor specific die-internal DfT features.

Stacking of dies requires that the vertical interconnects (microbumps and TSVs) are aligned with respect to footprint (i.e., matching x,y locations), mechanical properties (i.e., matching materials, diameter, height, etc.), and electrical properties (i.e., matching driver/receiver pairs). As a generic DfT-only standard, P1838 does not govern these items. Similar to IEEE Std 1149.1 [8] and IEEE Std 1500 [10], it only defines a DfT architecture:

- Number, name, type, and function of test I/Os
- On-chip DfT hardware and corresponding description
- Clock-cycle accurate test operation protocol

Consequently, off-the-shelf P1838-compliant dies are not guaranteed to 'plug-n-play' with each other without further alignment. The additional test I/Os as defined by P1838 require more parameters: footprint, and mechanical and electrical properties; but the same is true for the functional interfaces!

IEEE Std P1838 tries to abstract from the physical implementation of the stack. To that end, it avoids terms such as 'top', 'bottom', 'up', and 'down'. Instead, P1838 has adopted terminology based on the logical position of the interconnects relative to the external I/Os, which are assumed to be concentrated in a single die, referred to as the *first die*. Relative to a die, the adjacent die connected in the direction of the external I/Os is referred to as the *previous die*; the collection of signals going to the previous die is referred to as the *primary interface* of this die. P1838 assumes that every die has a single primary interface which connects to its previous die; an exception is the first die, for which its primary interface connects to the external world. Relative to a die, an adjacent die connected in the direction opposite to the external I/Os is referred to as a *next die*; the collection of signals going to a next die is referred to as a *secondary interface* of this die. The idea is that pairs of primary and secondary interfaces form the interconnect between two dies; the primary interface of one die plugs into the secondary interface of its previous die. A die can have zero or more secondary interfaces. A die with zero secondary interfaces is called a *last die*, whereas a die with one or more secondary interfaces that is not a first die is called a *middle die*.



(a) Stack 1.

(b) Stack 2.

(c) Stack 3.

Figure 2: Three physically different implementations of a logically equivalent stack.

Figure 2 depicts three different physical implementations of what is for IEEE P1838 logically the same stack. The stack consists of three active dies. Die 1 holds all external I/Os as its primary interface and hence is the first die. The secondary interface of Die 1 connects to the primary interface of Die 2 and the secondary interface of Die 2 connects to the primary interface of Die 3. Hence, Die 2 is a middle die. Die 3 has no secondary interfaces and therefore is the last die in the stack. The stacks in Figures 2(a) and 2(b) are both single-tower 3D stacks. In Figure 2(a), all dies are face-down and the external I/Os are implemented as area-array bumps at the bottom-side of the stack. Consequently, the primary interfaces are implemented at the bottom-side of each die and the secondary interfaces are implemented at the top-side of each die. In Figure 2(b), all dies are face-up and the external I/Os are implemented as wire-bonded pads at the top-side of the stack. As a result, 'up' and 'down' are reversed in comparison to Figure 2(a): the primary interfaces are implemented at the top-side of each die and the secondary interfaces are implemented at the bottom-side of each die. The stack in Figure 2(c) is a 2.5D-SIC, with the three active dies stacked side-by-side on top of and interconnected by a passive interposer die. In this example, the primary and secondary interfaces of a die are both located on the same bottom-side of that die.

4 Serial Control Mechanism

The main purpose of P1838's serial control mechanism is to configure the dies in a stack into one of their many test modes, while high-bandwidth test data access is handled by the (optional) FPP. In addition, the serial control mechanism also provides low-bandwidth test data access (at a rate of one bit per clock cycle) that remains accessible even when the die stack is soldered onto a printed circuit board and the FPP (which is typically multiplexed onto functional terminals) might no longer be directly accessible.

The P1838 Working Group has decided to base the P1838 serial control mechanism on IEEE Std 1149.1 [8], due to its maturity and the vast body of available hardware and software test solutions. The primary interface of a compliant die is equipped with an IEEE Std 1149.1-compliant *Test Access Port* (TAP), consisting of the four input terminals TDI (Test Data Input), TCK (Test Clock), TMS (Test Mode Select), and TRSTN (Test Reset Not), and the output terminal TDO (Test Data Output); in the context of P1838, we refer to this TAP as the *primary TAP*. Associated to this primary TAP is an IEEE 1149.1 *TAP Controller* consisting of the well-known 16-state finite state machine and corresponding decode logic [8].

A compliant die has each of its secondary interfaces equipped with a *secondary TAP*.[1] A secondary TAP is meant to plug into a primary TAP; consequently, it consists of the same five terminals, but with reversed direction. For secondary interface n (with $n \geq 1$), the secondary TAP input terminal is named TDI_Sn, while its four output terminals are named TDO_Sn, TCK_Sn, TMS_Sn, and TRSTN_Sn.[2]

Figure 3 shows the serial control mechanism for a middle die with two secondary interfaces. In this figure, the primary interface (including the primary TAP) resides on the bottom side of the die, while the two secondary interfaces (including their secondary TAPs, i.e., TAP_S1 and TAP_S2 respectively) reside on the top side of the die.

P1838 distinguishes the following types of registers (shown in Figure 3 – the color of configuration multiplexers corresponds to the color of the register that controls that multiplexer).

[1] A last die has no secondary interfaces, and therefore all its secondary interfaces are by definition equipped with a secondary TAP.

[2] In IEEE Std 1149.1, four TAP terminals are mandatory and TRSTN is optional. IEEE P1838 needs a primary TAP to plug into a secondary TAP, and hence cannot afford a variable number of TAP terminals. For maximum functionality, we have decided to settle on five terminals for all P1838 TAPs.

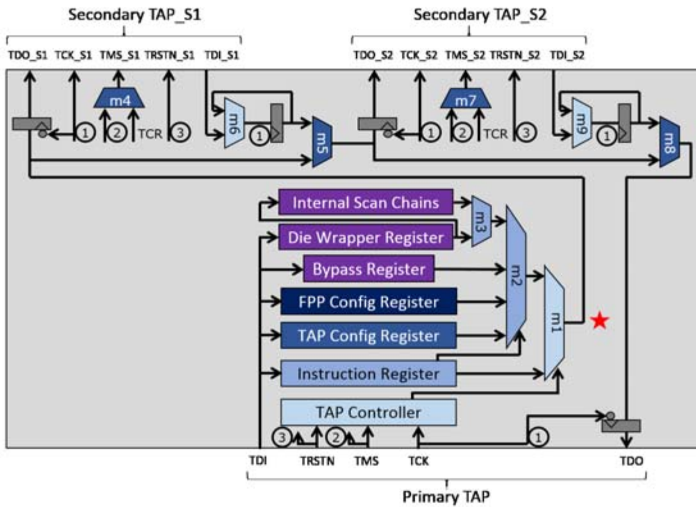


Figure 3: Serial control mechanism.

- Instruction Register

  1. *Instruction Register* (IR): the user programs IR to select which Test Data Register (TDR) is active between TDI and TDO and the test mode of this die (e.g., INTEST, EXTEST, or non-test mode); in Figure 3, IR controls multiplexers $m2$ and $m3$.

- Configuration Registers: dedicated TDRs that configure specific P1838 test resources.

  2. *TAP Configuration Register*: the TDR that determines which of this die's secondary interfaces are activated and have their serial control mechanisms included into the TDI–TDO path of this die. The TAP Configuration Register controls multiplexers $m4$, $m5$, $m7$, and $m8$.
  3. *FPP Configuration Register*: configures P1838's optional flexible parallel port – see Section 6.

- Regular Test Data Registers (shown in purple in Figure 3):

  4. *Bypass Register*: selected by the IR if this die is *not* tested (but perhaps another die in the die stack is). The Bypass Register consists of one flip-flop clocked by TCK.
  5. *Die Wrapper Register* (DWR): selected by the IR if this die is in INTEST or EXTEST mode. More details in Section 5.
  6. *Internal Scan Chains*: device registers selected in addition to the DWR to support die-internal tests, as required.

  There could be other TDRs, e.g., for electronic chip identification; these are not shown in Figure 3.

The serial control mechanism transports instructions, configuration data, and actual test data (i.e., test stimuli and responses) between TDI and TDO of its primary TAP. This reconfigurable scan path consists of two concatenated parts (in Figure 3 a red star indicates the concatenation location): (1) registers on *this* die, and (2) registers in the die towers that are stacked onto this die (or their bypass). Multiplexers $m5$ and $m8$ determine whether the serial scan paths from the die towers at secondary interfaces 1 and 2, respectively, will be included into this die's serial scan path; their control signals come from the user-programmable TAP Configuration Register. If a tower is activated and its corresponding secondary interface is selected, its TDI–TDO scan path is included in the serial scan path of our die, and TCK, TMS, and TRSTN of our die are passed on to the corresponding terminals of the selected secondary TAP. If a secondary TAP_Sn is deselected, its TMS_Sn output receives a user-programmable value from the TAP Configuration Register, switched in by multiplexers $m4$ and $m7$, respectively; these multiplexers are controlled by the same signals that also control $m5$ and $m8$, respectively.

Generally, scan chains are sensitive for data loss due to set-up/hold-time violations, as there is typically little to no propagation delay between subsequent flip-flops in the scan chain. In 3D die stacks, the die boundary crossings are prone to such timing violations, as the stacked dies typically originate from different design teams and/or fabrication facilities and consequently, it is quite likely they implement different clock-tree distribution approaches. To protect the serial control mechanism against loss of scan data due to set-up/hold time violations at die boundary crossings, P1838 has adopted rules (similar to IEEE Std 1149.1) that incoming TDI data shall be acquired on the rising edge of TCK, while TDO outputs shall change on the falling edge of TCK. These rules hold for the primary as well as secondary TAPs. In Figure 3, TDO, TDO_S1, and TDO_S2 are all equipped with a retiming element driven by the falling edge of TCK. TDI always drives the IR or one of the selected TDRs, which all run on the rising edge of TCK. Dedicated pipeline flip-flops clocked on the rising edge of TCK are added after TDI_S1 and TDI_S2, in order to assure that these TDI inputs acquire their incoming data on the rising edge of TCK. In Figure 3, these pipeline registers are fed by multiplexers $m6$ and $m9$ respectively, which receive SHIFT_IR OR SHIFT_DR from the TAP Controller as control input in order to be able to hold their data in the TAP Controller's Pause states.

5 Die Wrapper Register

IEEE Std P1838 mandates a die wrapper register (DWR) that provides controllability and observability to the inputs and outputs of a die. Currently most of the rules created for the DWR pertain solely to digital signals. Some digital signals, such as test signals, clocks and asynchronous signals are exempt from the DWR rules. There is still ongoing discussion on whether this standard will be extended to other signal types (e.g., analog) for control or observability of those signals during test.

The DWR provides isolation to enable internal test (INTEST) of the die. It also enables external testing (EXTEST) of the interconnect (micro-bumps, TSVs, and/or interposer wires) between dies without requiring access to the entire die. These DWR requirements are much the same as the IEEE 1500 Wrapper Boundary Register (WBR) requirements are on an embedded core [10, 17]. Utilizing only the DWR for interconnect testing en-

ables die providers to protect their design IP, as only the wrapper design must be delivered to enable the test. It also enables full testing of the interconnect with a small amount of logic which allows the CPU time and memory requirement during test creation to be much smaller than it would be if the complete design of both dies had to be utilized. Figure 4 shows the DWR in yellow.

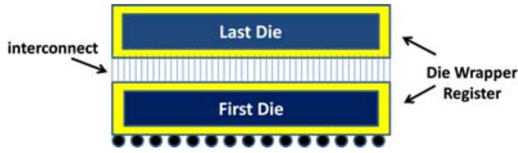


Figure 4: Die with IEEE P1838 DWRs.

There must be a DWR configuration with a single scan chain where the input to the scan chain is connected to the test input (TI) of the DWR and the output is connected to the test output (TO) of the DWR. During the various serial modes, the TI and TO of the DWR would be connected to the primary TAP's TDI and TDO, respectively.

There is currently an ongoing discussion around possible requirements with regard to a configuration for each interface type; for instance, mandating a DWR segment for the primary interface and one for the secondary interface. If there are multiple secondary interfaces a DWR segment for each of additional interfaces may be required to enable a more optimized mid-bond test access path of the interconnect as shown in Figure 5. Die 1 has one primary interface and two secondary interfaces. If only Die 2 has been stacked onto Die 1 and there is a desire to perform mid-bond testing, only the DWR segment from Secondary Interface 1 must be enabled.

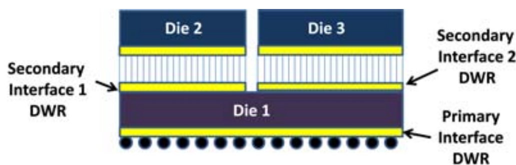


Figure 5: Separate DWR segments per interface.

Beyond the DWR segmentation per (primary/secondary) interface, the total number of segments in a DWR is at the discretion of the implementer. There may be a desire to add more DWR segments so that the scan chains can be shortened. The FPP is utilized to access multiple DWR segments in parallel when they are not concatenated into a single DWR chain (see Section 6).

5.1 Die Wrapper Register Cell

The DWR comprises DWR cells. A DWR cell can be a *shared* cell which reuses a functional storage element (e.g., flip-flop). This generally occurs on registered terminals (i.e., terminals that have a register directly connected to the terminal). The DWR cell can also be a *dedicated* wrapper cell, as shown in Figure 6. A dedicated wrapper cell uses one or more dedicated storage elements and must have a mode control (e.g., INTEST or EXTEST). Scan access enables the controllability and observability of each DWR cell.

There are two classes of DWR cells: *fully-provisioned* and *partially-provisioned*. The standard has rules that define the requirements of a fully-provisioned DWR cell, a partially-provisioned DWR cell, and when it is appropriate to use each type of DWR cell.

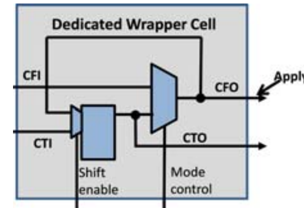


Figure 6: Dedicated wrapper cell example.

A fully-provisioned DWR cell must adhere to the following requirements that are illustrated in Figure 6.

1. At least one storage element connected between the cell functional input (CFI) and cell functional output (CFO).
2. At least one storage element connected between the cell test input (CTI) and cell test output (CTO). This storage element can be the same storage element as described in Requirement 1.
3. The capability to service the capture event.
4. The capability to service the shift event.
5. The capability to enable the apply event.

The *capture event* allows the value on the CFI or CFO to be captured into a storage element. The *apply event* (shown in Figure 6) is when the test data becomes active at the CFO output of the DWR cell as test stimuli. The *shift event* moves the value from the storage element to CTO and from CTI into the storage element. Figure 6 shows the functional and test terminals of a dedicated wrapper cell.

A fully-provisioned cell is required on all digital inputs and outputs, except for digital signals that cause data to be loaded into a sequential element (e.g., clock, asynchronous reset) or dedicated test signals specified by the standard.

A partially-provisioned cell must, at a minimum, be able to service the shift event and either the capture or apply event. If a signal is asynchronous and needs direct control from the tester, then there is no reason to have control capability (i.e., apply) in the DWR cell during scan test. However, if there is a desire to check the connectivity of the signal to the terminal of the die under test, an observe-only cell (e.g., shift and capture capability only) can be added to the DWR scan chain and connected to a terminal as shown in Figure 7.

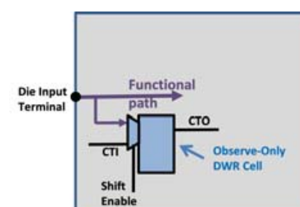
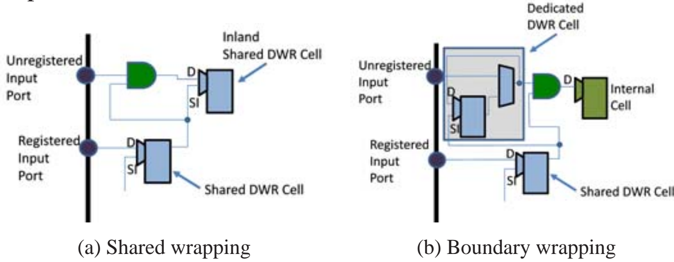


Figure 7: Example of an observe-only partially-provisioned DWR cell.

Beyond the basic requirements, there are many options permitted for a DWR cell, as discussed in Section 5.2.

As mentioned earlier, a shared DWR cell generally reuses the functional storage element of a registered input or output die terminal. One of the items under discussion in the IEEE Std P1838 Working Group is whether 'inland' wrapping should be allowed in this standard [18]. This is a methodology used often on 2D wrapping of embedded cores to reduce test logic and to get true functional timing during test from the external logic to the internal logic through a minimal amount of combinational ('shore') logic. An example of an inland shared wrapper cell is shown in Figure 8(a). The AND gate connected to the unregistered input terminal would be tested during EXTEST, rather than INTEST. Commercial ATPG tools know how to support inland wrapping automatically. The benefit of this methodology is that a dedicated wrapper cell (as shown in Figure 8(b)) is not required. This reduces area and prevents test logic from being added into what can be a timing-critical functional path. In addition, during EXTEST mode, it allows the true flop-to-flop functional path to be tested enabling a more accurate at-speed test.



(a) Shared wrapping (b) Boundary wrapping

Figure 8: Example of (a) shared wrapping with 'inland' wrapper cell and (b) boundary wrapping with dedicated wrapper cell.

Another place where a fully-provisioned DWR cell is required is the control signal of a three-state gate. It is also required that during shift, the state of the control signal of a three-state gate be persistent and a safe state should be maintained on the output of the three-state gate.

5.2 DWR Cell Naming Convention

The IEEE P1838 wrapper cell naming convention is inspired by IEEE Std 1500 and can be described by the following regular expression:
/DC(_S[DF]\d+)(_C([IO]([IO\d+]))|N)?(_U)?(_O)?(_G[01]?)?/

- The first character field, "(DC)", is mandatory and indicates that it is a die wrapper cell.
- The second character field, "(_S[DF]\d+)", describes the shift storage element (_S). If the wrapper cell is dedicated, it is described with a "D"; if it reuses a functional storage element (shared), it is described with an "F" ([DF]). The number of shift storage elements in the wrapper cell must also be described (\d+).
- The third character field, (_C([IO]([IO\d+]))|N), describes the capture storage element (_C). The functional input terminal from which the data is captured is described next ([IO]). If it is captured from the functional input of the DWR cell, the description character is "I". If it is captured from the functional output of the DWR cell the description character is "O". The next set of characters ([IO]|\d) describes into which storage element, with regard to the test input, data is captured. If there are multiple shift storage elements, the data can be captured into any of these storage elements. If the cell is closest to the test input or test output, then the "I" or "O" character, respectively, can be used. The \d+ must be used for any storage element that is not closest to the test input or test output. "N" is used when there is no capture capability in the DWR cell.
- The optional fourth character field (_U) describes if there is an update register.
- The optional fifth character field (_O) is used when the DWR cell only has observe capabilities.
- The optional sixth character field (_G[01]) is used to describe a gate that outputs either a 0 or 1 when enabled.

The description, using the DWR cell naming convention, for the dedicated DWR cell shown in Figure 6 is DC_SD1_COI. It is a dedicated DWR cell and there is only one shift element (SD1). Data is captured into an element that is next to functional output and closest to the test input (COI).

6 Flexible Parallel Port

State-of-the-art integrated circuits can have millions of flip-flops on chip. Having only a serial port would require millions of clock cycles only to shift in one test pattern, thereby leading to significantly long test times. The cost of ICs increases due to the elongated test time. Therefore, using a parallel port that can transport multiple stimulus and response bits simultaneously becomes highly desirable [12]. The on-chip scan chain is divided into multiple shorter scan chains that receive test stimuli from and transfer test responses to the parallel port. To avoid the overhead of having additional pins, the parallel port can share pins with existing functional pins through multiplexing. One particular focus of IEEE Std P1838 is to develop and standardize a flexible parallel port (FPP). The same as a conventional parallel port [12], this FPP is expected to be used for high-throughput testing; however, the FPP provides additional flexibility and configurability compared to the conventional parallel port. The FPP provides a flexible template that covers common, advanced, and even exotic test scenarios. The design and optimization of IEEE P1838's FPP are described in this section.

We distinguish two FPP components: a *registered lane* and a *non-registered lane*. Typical examples for a registered lane and a non-registered lane are a data lane and a clock lane, respectively. Data lanes are used to transport test stimuli and test responses. The data are registered by flip-flop(s) and a latch inside the lane. A clock lane is used to distribute a clock signal to data lanes. The clock signal is not registered inside the lane. The design of registered lanes and non-registered lanes is introduced in Sections 6.1 and 6.2, respectively. The principle of using multiple lanes on one die

is described in Section 6.3. The flexibility of the FPP is demonstrated in Section 6.4. A case study is presented in Section 6.5 to demonstrate how an existing test strategy can be implemented with the FPP.

6.1 FPP Registered Lane

IEEE Std P1838's FPP registered lane is shown in Figure 9. The FPP registered lane is used as a transportation switch box for test patterns. There are six terminals in the FPP registered lane: FPP_PRI, FPP_SEC, FPP_FROM_CORE, FPP_TO_CORE, FPP_FROM_SIDE, and FPP_TO_SIDE. FPP_FROM_CORE and FPP_FROM_SIDE are inputs. FPP_TO_CORE and FPP_TO_SIDE are outputs. FPP_PRI and FPP_SEC are bi-directional terminals that can be configured to serve only as input or output. Test patterns can be transferred from any input (*source* terminal) to any output (*sink* terminal) via a registered lane. A clock signal (FPP_CLK_IN) is connected to the clock inputs of all the pipeline flip-flops and the lock-up latches in the registered lane. Eight control signals (FPP_CORE_SEL, FPP_PRI_SEL, FPP_SIDE_SEL, FPP_REGPU_BYP, FPP_REGPD_BYP, FPP_REGN_BYP, FPP_PRI_EN, and FPP_SEC_EN) are used to control the multiplexers and tri-state buffers in the lane. These eight control signals are also the origin of the flexibility and configurability of the registered lane.

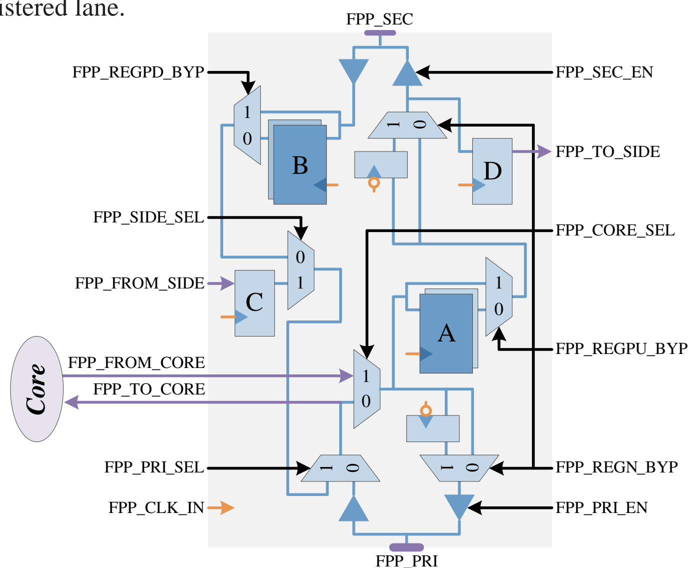


Figure 9: The generic template of IEEE P1838's FPP registered lane.

The flexibility and configurability of the FPP can be obtained either (1) during the FPP design phase or (2) during its use phase. During the design phase, components and ports in a registered lane are, for the most part, optional. Users can configure the registered lane by setting appropriate control signals such that unnecessary components and terminals can be removed. However, note that the registered lane is supposed to be a transportation switch box. Therefore, if one source terminal, such as FPP_FROM_CORE, is used in a registered lane, a sink terminal, such as FPP_TO_SIDE has to be utilized simultaneously. Alternatively, chip designers could include all or most of the components and control signals of the registered lane in their design and leave the flexibility and configurability to the user of the lane. Subsequently, during the use phase, users of the registered lane can implement different test scenarios by setting the lane's control signals to different values.

While being used as a data lane, the registered lane can transport test patterns from/to another die via FPP_PRI or FPP_SEC. The data can also be transferred to/from the functional core via FPP_TO_CORE or FPP_FROM_CORE. To enable 'horizontal' data transfer within the same die, terminals FPP_FROM_SIDE and FPP_TO_SIDE are utilized. Pipeline flip-flops are used in the registered lane for maintaining timing robustness. At least one pipeline flip-flop has to be used in the registered lane for timing robustness with data transfer from die-to-die, while optionally more pipeline flip-flops can be inserted. When a circuit designer designs a single die, there may be no information about how many dies are finally stacked. If many dies are stacked together, there could be timing issues while transferring data vertically from die to die due to the long distance. A flip-flop (at position A or B in Figure 9) is therefore used in each registered lane for vertical pipelining. When the horizontal distance between FPP_PRI and FPP_SEC is long, the pipelining of data in the horizontal direction is also essential. One or more flip-flops are needed at position A or B in the registered lane. When the registered lane is used as a uni-directional data lane, having flip-flop(s) at position A is sufficient for horizontal pipelining. Alternatively, when the registered lane is bi-directional, having flip-flop(s) at position B is essential for horizontal pipelining as well. Flip-flop(s) at positions C and D could also be necessary for horizontal pipelining while transferring data from lane to lane within the same die. One optional bypassing mechanism is provided for flip-flops at positions A and B while not provided for flip-flops at positions C and D. When the data have transferred through the functional core, the data are registered by the flip-flops inside the functional core. Flip-flops at positions A and B are therefore not needed for vertical pipelining and can be bypassed. Alternatively, flip-flops at positions C and D can be omitted if not needed for horizontal pipelining when the horizontal distance between two lanes in the same die is short. Two lock-up latches are in the template, which are used while transferring data from die to die. The lock-up latches guarantee that transported data leave a die at the negative edge of the clock signal, thereby minimizing the chance for hold-time violations [8].

6.2 FPP Non-Registered Lane

The FPP's non-registered lane is shown in Figure 10. The non-registered lane is used as a transportation switch box for transporting signals that do not require registration. Typical utilization of the non-registered lane is as clock lane. The non-registered lane is exactly a non-registered version of the registered lane without pipeline flip-flops or lock-up latch. Both lane types have the same terminals, apart from the following differences: a non-registered lane has one extra output terminal, FPP_CLOCK_OUT. while a registered lane has one extra input terminal, FPP_CLK_IN. In a clock lane, FPP_CLOCK_OUT provides the clock signal for a registered lane by connecting to the FPP_CLK_IN terminal of that registered lane. In case the clock signal needs to be transported further without being consumed yet as clock signal, the FPP_TO_SIDE output is being used. Five control signals (FPP_CORE_SEL, FPP_SIDE_SEL, FPP_PRI_SEL, FPP_PRI_EN, and FPP_SEC_EN)

are provided to configure the non-registered lane. These five controls are also the origin of the flexibility of the non-registered lane. As clock lane, a non-registered lane can obtain a clock signal from another die via FPP_PRI or FPP_SEC. The clock lane can also reuse the test clock (TCK) from the primary pin of the 3D chip as the clock signal.

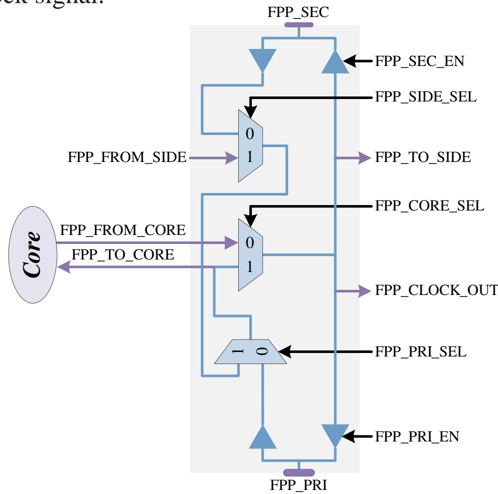


Figure 10: The generic template of IEEE P1838's FPP non-registered lane.

6.3 Multiple FPP Lanes on One Die

Multiple registered and/or non-registered lanes can be used simultaneously on one die, as illustrated in Figure 11. Multiple registered lanes, which share the same control and clock signals, are bundled together as a *channel*.

To control all the FPP lanes on a die, an FPP Configuration Register is essential. During the implementation of an FPP lane, a control signal is hardcoded if a specific logic value is assigned and does not need to be implemented in hardware. The remaining control signals that are not hardcoded have to be put in the FPP Configuration Register. Assume that there is one clock lane and two channels with registered lanes on a die as illustrated in Figure 11. The clock lane does not have any control signal while the two channels have three and four control signals. The FPP Configuration Register therefore has eight bits: seven bits from the two lanes plus one Config_Hold bit. The Config_Hold blocks the (possibly chip-wide) reset signal to be propagated through to the FPP Configuration Register.

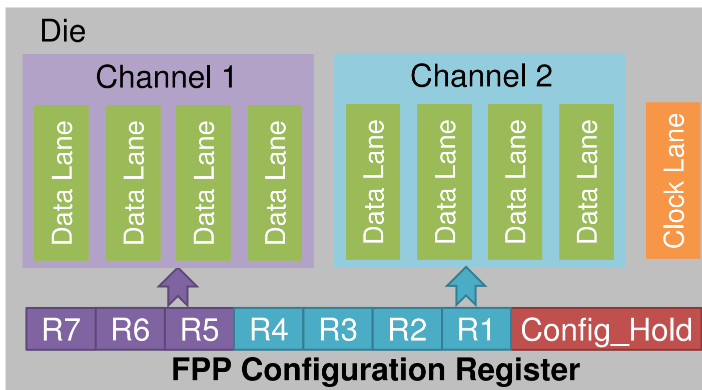


Figure 11: Illustration of using multiple FPP lanes on one die.

6.4 The Flexibility of IEEE P1838's FPP

The flexibility of IEEE P1838's FPP is demonstrated in this section. As discussed in Section 6.1, the flexibility of the FPP can be obtained either during the design phase or during the use phase. The flexibility of the FPP during the use phase is illustrated in Figure 12. Three dies are stacked together. Assume that two non-registered lanes and eight registered lanes exist on each die. One non-registered lane can be used as clock lane while the other non-registered lane can be used to transport a 'Test_Enable' signal. For one particular test scenario, where the three dies require different test patterns, four of the eight registered lanes can be used as upward lanes to transport test stimuli to any die. The other four registered lanes are used as downward lanes to transport test responses. In another test scenario, where the three dies are identical and require the same test patterns, two registered lanes can be used as upward lanes to broadcast test stimuli to the three dies. The other six registered lanes are used as downward lanes to receive test responses where each pair of two lanes serves one die. If these two test scenarios are preferred to be accommodated by the same DfT hardware, two registered lanes have to be implemented as bi-directional lanes to satisfy the different test requirements at different moments for the same chip.

If the circuit designers are clear that only one test scenario occurs for the chip, they can hardcode all or most of the control signals for the FPP lanes to eliminate the unnecessary hardware in the FPP templates. The hardware overhead is avoided by pruning the flexibility of the FPP lanes during the design phase. Consequently, the flexibility of the FPP becomes limited during the subsequent use phase.

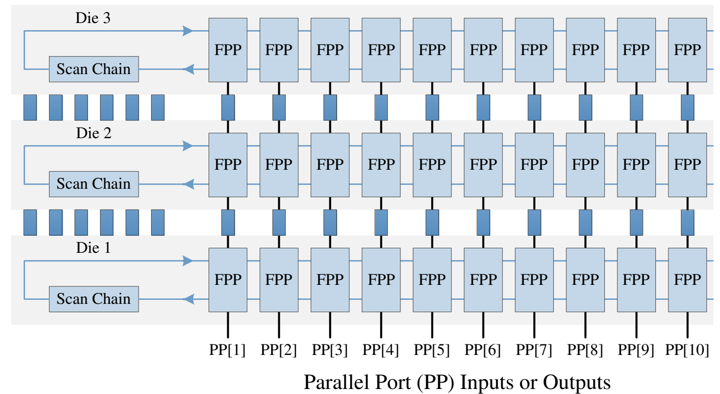


Figure 12: Utilization example with IEEE P1838's FPP.

6.5 Case Study with IEEE P1838's FPP

To verify the feasibility and capability of the IEEE P1838's FPP registered and non-registered lanes, a test case for a single-tower 3D stack is studied in this section.

In [12], a conventional parallel port (CPP) is proposed for testing single-tower 3D-SICs. This proposed CPP is redrawn in Figure 13 in the style of IEEE P1838's FPP. For testing a single-tower 3D-SIC with two dies, one CPP is required on each die. To achieve the mapping from FPP to CPP, two registered lanes have to be used to implement one CPP on each die. One FPP data lane is used for up-

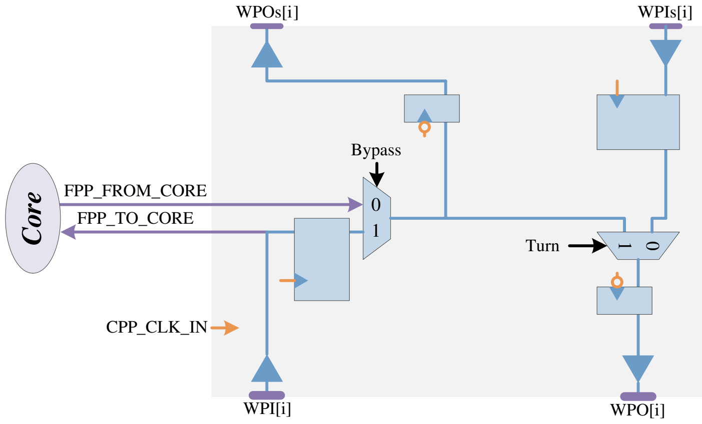
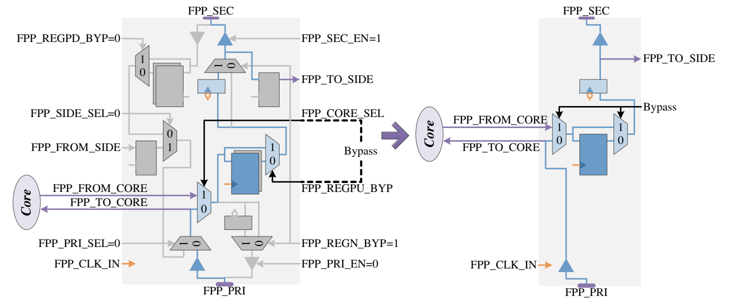


Figure 13: The conventional parallel port that is drawn in the style of IEEE P1838's FPP.

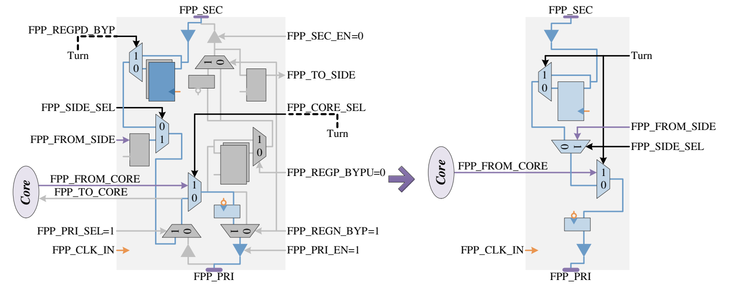
ward data transfer, while the other FPP data lane is used for downward data transfer. The mapping process of the upward FPP data lane is shown in Figure 14(a). By setting FPP_PRI_EN = 0 and FPP_SEC_EN = 1, two of the four buffers are removed from the FPP logic. By setting FPP_PRI_SEL = 0, FPP_REGN_BYP = 1, FPP_SIDE_SEL = 0, and FPP_REGPD_BYP = 0, four multiplexers, one pipeline flip-flop, and one lock-up latch are removed. When the on-die scan chains are bypassed, the test patterns have to go through a pipeline flip-flop for maintaining timing robustness. Alternatively, when the test patterns go through the on-die scan chains, the pipeline flip-flop can be bypassed. Therefore, FPP_CORE_SEL and FPP_REGPU_BYP can be combined into one control bit 'Bypass', as shown in Figure 14(a). After removing all the unnecessary components, terminals, and control signals, the upward FPP data lane is shown in Figure 14(a). The test patterns can be transferred from the primary port (FPP_PRI) to the secondary port (FPP_SEC), while going through or bypassing the on-die scan chains depending on the value of the control signal 'Bypass'.

The mapping process of the downward FPP data lane to CPP is similar to the upward FPP data lane. As shown in Figure 14(b)[3], by setting FPP_PRI_EN = 1, FPP_SEC_EN = 0, FPP_PRI_SEL = 1, FPP_REGN_BYP = 1, and FPP_REGPU_BYP = 0, two buffers, four multiplexers, one pipeline flip-flop, and one lock-up latch are removed. FPP_CORE_SEL and FPP_REGPD_BYP can be combined into one control signal 'Turn'. The test patterns can go from the secondary port (FPP_SEC) to the primary port (FPP_PRI) or from the core logic to the primary port.

Since clock distribution is necessary for this test case, clock lanes need to be implemented by FPP non-registered lanes. The configuration of the FPP clock lane is shown in Figure 15. For Die 1 that is connected to the pins of the chip, the clock signal can reuse the test clock (TCK). This clock signal is transmitted from FPP_FROM_CORE to FPP_CLK_OUT (FPP_TO_SIDE), providing the clock signal used for Die 1. The same clock signal is also transmitted to FPP_SEC, providing the clock signal for Die 2. The clock lane for Die 2 does not need any connection to the core logic, tri-state buffers, or multiplexers. The clock lane for Die 2 receives



(a) Mapping of upward FPP data lane.



(b) Mapping of downward FPP data lane.

Figure 14: The mapping process of FPP data lanes to CPP data lanes. The unused circuits, wires, and terminals are grayed out.

the clock signal from Die 1 via FPP_PRI, providing the clock signal for Die 2 via FPP_CLK_OUT. This clock signal can also be provided to the next die (if there is any) via FPP_SEC.

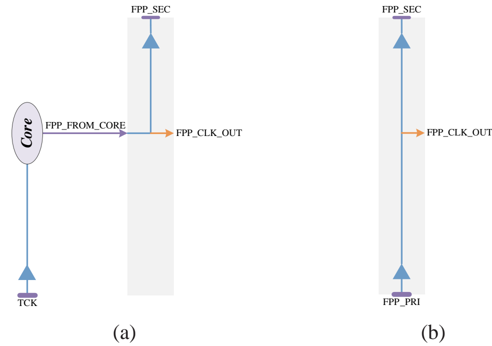


Figure 15: The mapping from the generic FPP non-registered lane to clock lanes in different stacked dies. (a) The clock lane used for a first die. (b) The clock lane used for next dies.

After finishing the configuration of the FPP data lanes and clock lanes, the test infrastructure that is intended to implement the same functionality as the CPP (see Figure 13) is shown in Figure 16. The same test infrastructure is repeated on the two stacked tiers with different clock lane implementations. The FPP-based test infrastructure can test Die 1 or Die 2 alone, or test Die 1 and Die 2 with the same or different test patterns. Note that besides the normal test capability, this FPP-based infrastructure can also perform a special function that is able to be performed by the CPP in Figure 13, which is the combination of bypass + turnaround, thanks to the employment of the two terminals FPP_FROM_SIDE and FPP_TO_SIDE. By setting both 'Bypass' and 'Turn' to 0, the

[3]The flip-flops that are used for horizontal pipelining are all omitted in both mapping procedures in Figure 14 by assuming that the horizontal distances between FPP_PRI and FPP_SEC and between the two data lanes are both sufficiently short.

test patterns do not go into the core logic and leave the upward data lane from FPP_TO_SIDE. The test patterns go into the downward data lane from FPP_FROM_SIDE and come back to the primary port (FPP_PRI) of the downward data lane, as indicated by the dashed line in Figure 16.

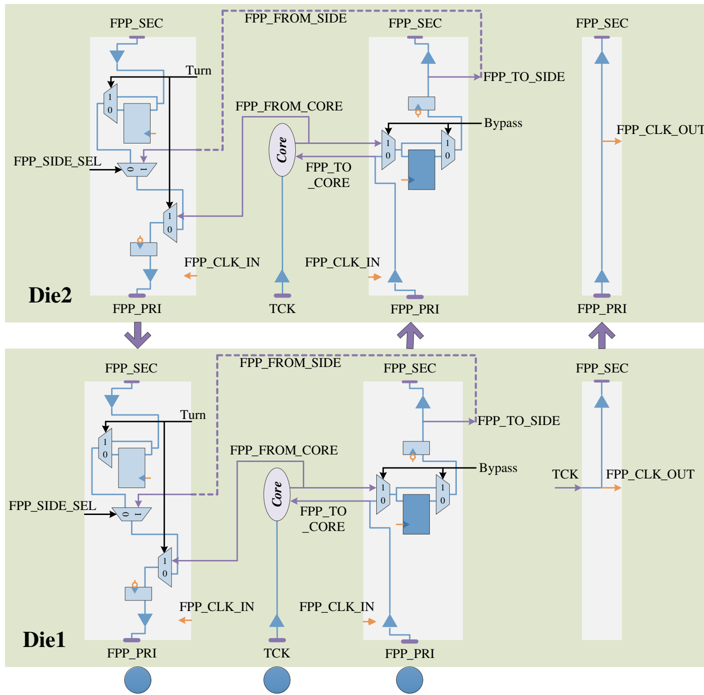


Figure 16: The testing infrastructure that is intended to implement the same functionality as the CPP by using the initially-proposed FPP data lane.

7 Conclusion

3D-SICs and their 2.5D and 5.5D variants will soon be hitting the markets. Effective mid-bond and post-bond test access requires a 3D DfT architecture that transports the test control and test data signals up and down the stack. IEEE Std P1838 is a standard-under-development for such a 3D DfT architecture. This paper provided a status update of P1838, with description of its three main hardware components: the serial control mechanism, the die wrapper register, and the flexible parallel port.

Acknowledgments

The authors thank the other members of the IEEE Std P1838 Working Group for many fruitful discussions. Special thanks go to Adam Cron (P1838 Working Group Vice Chair and Chair of two P1838 Tiger Teams) for his constructive feedback on an earlier version of this paper. The work from Hailong Jiao was partially funded by project FP7 CT217 RESIST.

References

[1] Handel H. Jones. *Technical Viability of Stacked Silicon Interconnect Technology*. Xilinx, October 2010. White Paper, see http://www.xilinx.com/publications/technology/stacked-silicon-interconnect-technology-ibs-research.pdf.

[2] Kirk Saban. *Xilinx Stacked Silicon Interconnect Technology Delivers Breakthrough FPGA Capacity, Bandwidth, and Power Efficiency*. Xilinx, October 2010. White Paper, see `http://www.xilinx.com/support/documentation/white_papers/wp380_Stacked_Silicon_Interconnect_Technology.pdf`.

[3] Christian Freund. Wide-IO DRAM – ST-Ericsson's First Mobile Processor Using TSV 3D-IC Technology. In *CDNLive! EMEA*, May 2011.

[4] Jeff Rearick et al. New Paradigm Shift in 3-D Design and Testing. In *Digest of IEEE International Workshop on Testing Three-Dimensional Stacked Integrated Circuits (3D-TEST)*, October 2015. Paper 1.1, `http://3dtest.tttc-events.org`.

[5] Erik Jan Marinissen. Challenges and Emerging Solutions in Testing TSV-Based 2.5D- and 3D-Stacked ICs. In *Proceedings Design, Automation, and Test in Europe (DATE)*, pages 1277–1282, March 2012. doi:10.1109/DATE.2012.6176689.

[6] Mukesh Agrawal and Krishnendu Chakrabarty. Test-Cost Modeling and Optimal Test-Flow Selection of 3-D-Stacked ICs. *IEEE Transactions on Computer-Aided Design*, 34(9):1532–1536, September 2015. doi:10.1109/TCAD.2015.2419227.

[7] Mottaqiallah Taouil et al. Using 3D-COSTAR for 2.5D Test Cost Optimization. In *Proceedings IEEE International Conference on 3D System Integration (3DIC)*, pages 1–8, October 2013. doi:10.1109/3DIC.2013.6702351.

[8] IEEE Computer Society. *IEEE Std 1149.1™-2013, IEEE Standard for Test Access Port and Boundary-Scan Architecture*. IEEE, New York, NY, USA, 2013. doi:10.1109/IEEESTD.2013.6515989.

[9] Chen-An Chen et al. Cost-Effective TAP-Controlled Serialized Compressed Scan Architecture for 3D Stacked ICs. In *Proceedings IEEE Asian Test Symposium (ATS)*, pages 107–108, November 2013. doi:10.1109/ATS.2013.29.

[10] IEEE Computer Society. *IEEE Std 1500™-2005, IEEE Standard Testability Method for Embedded Core-based Integrated Circuits*. IEEE, New York, NY, USA, August 2005. doi:10.1109/IEEESTD.2005.96465.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] IEEE Computer Society. *IEEE Std 1687™-2014, IEEE Standard for Access and Control of Instrumentation Embedded within a Semiconductor Device*. IEEE, New York, NY, USA, 2014. doi:10.1109/IEEESTD.2013.6974961.

[16] Yassine Fkih et al. 2D to 3D Test Pattern Retargeting Using IEEE P1687 Based 3D DFT Architectures. In *Proceedings of IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*, pages 386–391, July 2014. doi:10.1109/ISVLSI.2014.83.

[17] Francisco da Silva, Teresa McLaurin, and Tom Waayers. *The Core Test Wrapper Handbook – Rationale and Application of IEEE Std. 1500™*, volume 35 of *Frontiers in Electronics Testing*. Springer-Verlag, Boston, MA, USA, 2006. ISBN 0387307516.

[18] Konstantin Shibin et al. At-Speed Testing of Inter-Die Connections of 3D-SICs in the Presence of Shore Logic. In *Proceedings IEEE Asian Test Symposium (ATS)*, pages 79–84, November 2015. doi:10.1109/ATS.2015.21.