IEEE SA
STANDARDS
ASSOCIATION

IEEE Standard for Test Access Architecture for Three-Dimensional Stacked Integrated Circuits

STANDARDS

IEEE Computer Society

Developed by the
Test Technology Standards Committee

IEEE 1838™-2019

IEEE

IEEE Standard for Test Access Architecture for Three-Dimensional Stacked Integrated Circuits

Developed by the

Test Technology Standards Committee
of the
IEEE Computer Society

Approved 7 November 2019

IEEE SA Standards Board

Abstract: IEEE Std 1838 is a die-centric standard; it applies to a die that is intended to be part of a multi-die stack. This standard defines die-level features that, when compliant dies are brought together in a stack, comprise a stack-level architecture that enables transportation of control and data signals for the test of (1) intra-die circuitry and (2) inter-die interconnects in both (a) pre-stacking and (b) post-stacking situations, the latter for both partial and complete stacks in both pre-packaging, post-packaging, and board-level situations. The primary focus of inter-die interconnect technology addressed by this standard is through-silicon vias (TSVs); however, this does not preclude its use with other interconnect technologies such as wire-bonding.

Keywords: 3D test access, flexible parallel port, FPP, IEEE 1838, multi-tower stack, primary test access port, scan, secondary test access port, test, through-silicon via, TSV

The Institute of Electrical and Electronics Engineers, Inc.
3 Park Avenue, New York, NY 10016-5997, USA

PDF: ISBN 978-1-5044-6343-0 STD23997
Print: ISBN 978-1-5044-6344-7 STDPD23997

Important Notices and Disclaimers Concerning IEEE Standards Documents

IEEE documents are made available for use subject to important notices and legal disclaimers. These notices and disclaimers, or a reference to this page, appear in all standards and may be found under the heading "Important Notices and Disclaimers Concerning IEEE Standards Documents." They can also be obtained on request from IEEE or viewed at http://standards.ieee.org/ipr/disclaimers.html.

Notice and Disclaimer of Liability Concerning the Use of IEEE Standards Documents

IEEE Standards documents (standards, recommended practices, and guides), both full-use and trial-use, are developed within IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Association ("IEEE SA") Standards Board. IEEE ("the Institute") develops its standards through a consensus development process, approved by the American National Standards Institute ("ANSI"), which brings together volunteers representing varied viewpoints and interests to achieve the final product. IEEE Standards are documents developed through scientific, academic, and industry-based technical working groups. Volunteers in IEEE working groups are not necessarily members of the Institute and participate without compensation from IEEE. While IEEE administers the process and establishes rules to promote fairness in the consensus development process, IEEE does not independently evaluate, test, or verify the accuracy of any of the information or the soundness of any judgments contained in its standards.

IEEE Standards do not guarantee or ensure safety, security, health, or environmental protection, or ensure against interference with or from other devices or networks. Implementers and users of IEEE Standards documents are responsible for determining and complying with all appropriate safety, security, environmental, health, and interference protection practices and all applicable laws and regulations.

IEEE does not warrant or represent the accuracy or content of the material contained in its standards, and expressly disclaims all warranties (express, implied and statutory) not included in this or any other document relating to the standard, including, but not limited to, the warranties of: merchantability; fitness for a particular purpose; non-infringement; and quality, accuracy, effectiveness, currency, or completeness of material. In addition, IEEE disclaims any and all conditions relating to: results; and workmanlike effort. IEEE standards documents are supplied "AS IS" and "WITH ALL FAULTS."

Use of an IEEE standard is wholly voluntary. The existence of an IEEE standard does not imply that there are no other ways to produce, test, measure, purchase, market, or provide other goods and services related to the scope of the IEEE standard. Furthermore, the viewpoint expressed at the time a standard is approved and issued is subject to change brought about through developments in the state of the art and comments received from users of the standard.

In publishing and making its standards available, IEEE is not suggesting or rendering professional or other services for, or on behalf of, any person or entity nor is IEEE undertaking to perform any duty owed by any other person or entity to another. Any person utilizing any IEEE Standards document, should rely upon his or her own independent judgment in the exercise of reasonable care in any given circumstances or, as appropriate, seek the advice of a competent professional in determining the appropriateness of a given IEEE standard.

IN NO EVENT SHALL IEEE BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO: PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE PUBLICATION, USE OF, OR RELIANCE UPON ANY STANDARD, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE AND REGARDLESS OF WHETHER SUCH DAMAGE WAS FORESEEABLE.

Translations

The IEEE consensus development process involves the review of documents in English only. In the event that an IEEE standard is translated, only the English version published by IEEE should be considered the approved IEEE standard.

Official statements

A statement, written or oral, that is not processed in accordance with the IEEE SA Standards Board Operations Manual shall not be considered or inferred to be the official position of IEEE or any of its committees and shall not be considered to be, or be relied upon as, a formal position of IEEE. At lectures, symposia, seminars, or educational courses, an individual presenting information on IEEE standards shall make it clear that his or her views should be considered the personal views of that individual rather than the formal position of IEEE.

Comments on standards

Comments for revision of IEEE Standards documents are welcome from any interested party, regardless of membership affiliation with IEEE. However, IEEE does not provide consulting information or advice pertaining to IEEE Standards documents. Suggestions for changes in documents should be in the form of a proposed change of text, together with appropriate supporting comments. Since IEEE standards represent a consensus of concerned interests, it is important that any responses to comments and questions also receive the concurrence of a balance of interests. For this reason, IEEE and the members of its societies and Standards Coordinating Committees are not able to provide an instant response to comments or questions except in those cases where the matter has previously been addressed. For the same reason, IEEE does not respond to interpretation requests. Any person who would like to participate in revisions to an IEEE standard is welcome to join the relevant IEEE working group.

Comments on standards should be submitted to the following address:

Secretary, IEEE SA Standards Board
445 Hoes Lane
Piscataway, NJ 08854 USA

Laws and regulations

Users of IEEE Standards documents should consult all applicable laws and regulations. Compliance with the provisions of any IEEE Standards document does not imply compliance to any applicable regulatory requirements. Implementers of the standard are responsible for observing or referring to the applicable regulatory requirements. IEEE does not, by the publication of its standards, intend to urge action that is not in compliance with applicable laws, and these documents may not be construed as doing so.

Copyrights

IEEE draft and approved standards are copyrighted by IEEE under US and international copyright laws. They are made available by IEEE and are adopted for a wide variety of both public and private uses. These include both use, by reference, in laws and regulations, and use in private self-regulation, standardization, and the promotion of engineering practices and methods. By making these documents available for use and adoption by public authorities and private users, IEEE does not waive any rights in copyright to the documents.

Photocopies

Subject to payment of the appropriate fee, IEEE will grant users a limited, non-exclusive license to photocopy portions of any individual standard for company or organizational internal use or individual, non-commercial use only. To arrange for payment of licensing fees, please contact Copyright Clearance Center, Customer Service, 222 Rosewood Drive, Danvers, MA 01923 USA; +1 978 750 8400. Permission to photocopy portions of any individual standard for educational classroom use can also be obtained through the Copyright Clearance Center.

Updating of IEEE Standards documents

Users of IEEE Standards documents should be aware that these documents may be superseded at any time by the issuance of new editions or may be amended from time to time through the issuance of amendments, corrigenda, or errata. An official IEEE document at any point in time consists of the current edition of the document together with any amendments, corrigenda, or errata then in effect.

Every IEEE standard is subjected to review at least every 10 years. When a document is more than 10 years old and has not undergone a revision process, it is reasonable to conclude that its contents, although still of some value, do not wholly reflect the present state of the art. Users are cautioned to check to determine that they have the latest edition of any IEEE standard.

In order to determine whether a given document is the current edition and whether it has been amended through the issuance of amendments, corrigenda, or errata, visit IEEE Xplore at http://ieeexplore.ieee.org/ or contact IEEE at the address listed previously. For more information about the IEEE SA or IEEE's standards development process, visit the IEEE SA Website at http://standards.ieee.org.

Errata

Errata, if any, for IEEE standards can be accessed via https://standards.ieee.org/standard/index.html. Search for standard number and year of approval to access the web page of the published standard. Errata links are located under the Additional Resources Details section. Errata are also available in IEEE Xplore: https://ieeexplore.ieee.org/browse/standards/collection/ieee/. Users are encouraged to periodically check for errata.

Patents

Attention is called to the possibility that implementation of this standard may require use of subject matter covered by patent rights. By publication of this standard, no position is taken by the IEEE with respect to the existence or validity of any patent rights in connection therewith. If a patent holder or patent applicant has filed a statement of assurance via an Accepted Letter of Assurance, then the statement is listed on the IEEE SA Website at https://standards.ieee.org/about/sasb/patcom/patents.html. Letters of Assurance may indicate whether the Submitter is willing or unwilling to grant licenses under patent rights without compensation or under reasonable rates, with reasonable terms and conditions that are demonstrably free of any unfair discrimination to applicants desiring to obtain such licenses.

Essential Patent Claims may exist for which a Letter of Assurance has not been received. The IEEE is not responsible for identifying Essential Patent Claims for which a license may be required, for conducting inquiries into the legal validity or scope of Patents Claims, or determining whether any licensing terms or conditions provided in connection with submission of a Letter of Assurance, if any, or in any licensing agreements are reasonable or non-discriminatory. Users of this standard are expressly advised that determination of the validity of any patent rights, and the risk of infringement of such rights, is entirely their own responsibility. Further information may be obtained from the IEEE Standards Association.

Participants

At the time this IEEE standard was completed, the 3D Test Working Group had the following membership:

Adam Cron, *Chair*
Erik Jan Marinissen, *Vice Chair*
Michael G. Wahl, *Editor*
Eric Cormack, *Secretary*

Saman Adham
Sandeep Bhatia
Tapan Chakraborty
Jonathon E. Colburn
Jean-Francois Cote
Alfred Crouch
Heiko Ehrenberg
Sandeep Goel
Saurabh Gupta
Jon Haldorson
Gurgen Harutyunyan
Shuichi Kameyama
Harry Linzer
Teresa McLaurin
Sophocles Metsis
Seetal Potluri
Etienne Racine
Gunasekaran Ramasamy
Mike Ricchetti
A.T. Sivaram
Naveen Kumar Srivastava
Craig Stephan
Min-Jer Wang

Previous members of the 3D Test Technology working group are:

Vincent Chalendard
Chen-An Chen
Vivek Chickermane
C. J. Clark
Zoe Conroy
Damon Domke
Ted Eaton
William Eklow
Tom Heilmann
Michael Higgins
Chun-Lung Hsu
Marc Hutner
Hongshin Jun
Shuichi Kameyama
Rakesh Kinger
Amit Majumdar
T.M. Mak
Arie Margulis
Benoit Nadeau-Dostie
Christos Papameletis
Ben Rogel
Francisco Russi
Iftikhar Soomro
Brian Turmelle
Bill Tuthill
Lee Whetsel
Jae Wu

The following members of the individual Standards Association balloting group voted on this standard. Balloters may have voted for approval, disapproval, or abstention.

Saman Adham
Ken-Ichi Anzou
Sandeep Bhatia
Bill Brown
Demetrio Bucaneg Jr
Tapan Chakraborty
Jonathon E. Colburn
Eric Cormack
Jean-Francois Cote
Adam Cron
Alfred Crouch
Heiko Ehrenberg
Peter van den Eijnden
Randall Groves
Jon Haldorson
Peter Harrod
Gurgen Harutyunyan
Werner Hoelzl
Michael Laisne
Philippe Lebourg
Adam Ley
Erik Jan Marinissen
Benoit Nadeau-Dostie
Mike Ricchetti
Anthony Sparks
Naveen Srivastava
Jon Charles Stewart
Walter Struppler
Srinivasa Vemuru
Michael G. Wahl
Lisa Ward
Karl Weber

When the IEEE SA Standards Board approved this standard on 7 November 2019, it had the following membership:

Gary Hoffman, *Chair*
Ted Burse, *Vice Chair*
Jean-Philippe Faure, *Past Chair*
Konstantinos Karachalios, *Secretary*

Masayuki Ariyoshi
Stephen D. Dukes
J. Travis Griffith
Guido Hiertz
Christel Hunter
Joseph L. Koepfinger*
Thomas Koshy
John D. Kulick
David J. Law
Joseph Levy
Howard Li
Xiaohui Liu
Kevin Lu
Daleep Mohla
Andrew Myles
Annette D. Reilly
Dorothy Stanley
Sha Wei
Phil Wennblom
Philip Winston
Howard Wolfman
Feng Wu
Jingyi Zhou

*Member Emeritus

Introduction

This introduction is not part of IEEE Std 1838-2019, IEEE Standard for Test Access Architecture for Three-Dimensional Stacked Integrated Circuits.

Advancements in interconnect, assembly, and packaging technology have lead to a wide range of multi-die stack architectures. These die stacks need to be tested before they can be shipped with acceptable quality levels to customers. Consequently, three-dimensional design-for test (3D-DfT) structures that provide test access between the external stack I/Os and the various dies and inter-die interconnect are needed. Test access is needed for manufacturing phases that include both partially assembled and complete stacks. This standard addresses these issues.

Contents

List of Figures

List of Tables

IEEE Standard for Test Access Architecture for Three-Dimensional Stacked Integrated Circuits

1. Overview

1.1 Scope

IEEE Std 1838™-2019 standardizes mandatory and optional on-chip hardware components for 3D test access. It is intended that in the future a standard is developed for a formal, computer-readable language in which implementation choices for the three-dimensional design-for-test (3D-DfT) hardware can be specified and described. An idea of a language/data structure has been described in [B5].[1]

The aim of IEEE Std 1838 is to define at die-level standardized and scalable 3D-DfT features based on and working with digital scan-based test access, such that when compliant dies are stacked, a stack-level 3D-DfT test access architecture emerges with a minimum functionality and many optional extensions. IEEE Std 1838 provides a modular test access architecture, in which dies and interconnect layers between adjacent stacked dies can be tested individually. The focus of the standard is 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 provides test access via a mandatory one-bit ('serial') input/output test port and optional multi-bit ('parallel') test ports.

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 design-for-test (DfT) features comprise a stack-level test access architecture. In this way, the standard enables interoperability between die makers 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 for board-level interconnect testing (although the standard certainly does not prohibit application thereof).[2]

IEEE Std 1838 does not mandate specific defect or fault models, specific test generation methods, nor specific die-internal 2D-DfT features. However, the standard leverages existing 2D-DfT wherever applicable and appropriate, including test access ports (such as specified in IEEE Std 1149.1), on-chip DfT such as internal scan chains and wrappers of embedded cores (such as specified in IEEE Std 1500™), and on-chip design-for-debug and embedded instruments (such described in IEEE Std 1687™).

Stacking of dies requires that the vertical interconnects [e.g., micro-bumps and through-silicon vias (TSVs)] are aligned with respect to footprint (i.e., matching x,y layout locations), mechanical properties (i.e.,

[1]The numbers in brackets correspond to those of the bibliography in Annex B.
[2]Information on references can be found in Clause 2.

matching materials, diameter, height, etc.), and electrical properties (i.e., matching driver/receiver pairs). As a generic DfT-only standard, IEEE Std 1838 does not govern these items. Similar to IEEE Std 1149.1 and IEEE Std 1500, 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 IEEE Std 1838-compliant dies are not guaranteed to 'plug-n-play' with each other without further physical alignment. The additional test I/Os as defined by IEEE Std 1838 require more parameters: footprint, mechanical, and electrical properties (but the same is true for the functional interfaces).

1.2 Three-dimensional integrated circuits (ICs) stacking technology

The market continues to pull for integrated circuits (ICs) with higher performance, better energy-efficiency, and lower cost. While conventional transistor scaling runs into increasing technical and financial hurdles, the baton in the race to create attractive new IC products that meet market expectations is gradually being taken over by innovations in multi-die stack-assembly and packaging techniques. Large-array fine-pitch micro-bumps implement dense high-performance low-power inter-die interconnects. Through-silicon vias in combination with wafer thinning provide electrical connections via the back-side of a die's substrate, enabling stacks of more than two dies. Interposer dies, possibly implemented in a passive technology, offer low-cost high-performance ways to interconnect multiple dies. Packaging costs can be drastically reduced by using cheaper materials and processes and by turning packaging into a wafer-level operation. The I/O-to-pin fan-out functionality of package substrates can be replaced by redistribution metal layers, cost-effectively manufactured at wafer level. And the cost of plastic or ceramic packages can be circumvented by applying epoxy mold compounds at wafer level.

These and other interconnect, assembly, and packaging technology innovations have led to a wide range of multi-die stack architectures, including so-called "2.5D"-stacked ICs (SICs) consisting of multiple active dies stacked side-by-side on a passive silicon interposer base, "3D"-SICs comprising a tower of stacked active dies, and multi-tower-SICs that consist of multiple towers of stacked active dies side-by-side on a passive silicon interposer.

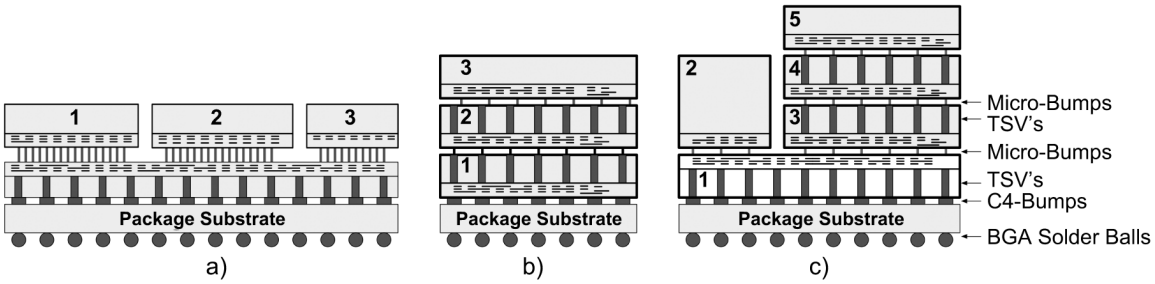

Figure 1—2.5-SIC (a), 3D-SIC (b), and 5.5D-SIC (c)

These new SIC product compositions offer various compelling benefits: (1) heterogeneous integration, i.e., the ability to use the most effective 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 large die is partitioned into multiple smaller dies that are stacked on top of each other— provided faulty dies can be adequately removed from the manufacturing process by means of testing before stacking.

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: (1) *pre-bond tests* prior to stacking, (2) *mid-bond tests* on incomplete, partial stacks, (3) *post-bond tests* on complete yet 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 and hence is subject of automated trade-off evaluation and optimization (see, [B1], [B4], and [B8]).

1.3 Motivation for a 3D-DfT standard

A well-architected DfT 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/or built-in self-test (BIST) engines) that provide test access within a single die, we also need new approaches for testing stacked systems. Especially once a (partial or complete) vertical stack has been formed (i.e., in mid-bond, post-bond, or final testing phases), we also need novel 3D-DfT structures that provide test access from (and to) the external stack I/Os to (and from) the various dies and inter-die interconnects. For example: if a stack consists of three dies and test access from external test equipment is exclusively possible via the stack I/Os that are concentrated in, say, Die 1, then Die 1 and Die 2 need to cooperate in transporting test stimuli and responses up and down the stack in order to be able to test Die 3.

To enable separation of the test development as well as test execution for the various dies in the stack, the 3D-DfT architecture should enable modular testing, i.e., tests for dies and interconnect layers between adjacent stacked dies can be developed and executed individually. Several ad-hoc 3D-DfT architectures have been proposed, among others based on IEEE Std 1149.1, IEEE Std 1500, and IEEE Std 1687. These architectures all have their specific strong and weak points. However, these ad-hoc 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 (even if designed and developed by different teams or different companies) are brought together in a die stack, a basic minimum of test features should work across the stack. This is exactly the aim of IEEE Std 1838.

1.4 Context

The Standard for Test Access Architecture for Three-Dimensional Stacked ICs is conceptually related to previously developed design-for-test (DfT) standards, in particular IEEE Std 1149.1, IEEE Std 1500, and IEEE Std 1687. The first two standards specify test access architectures for ICs on boards (IEEE Std 1149.1) and IP cores embedded within an IC (IEEE Std 1500). IEEE Std 1687 describes an access architecture to embedded instruments. These three previously developed standards have influenced and are referred to in this standard, so a solid understanding of these standards is strongly recommended.

1.5 Organization of the standard

Clause 1 provides an overview and context for this standard.

Clause 2 provides references necessary to understand this standard.

Clause 3 defines terminology and acronyms used in this standard.

Clause 4 is a tutorial that outlines the technologies addressed and utilized by this standard.

Clause 5 defines the serial access bus and structure which forms the mandatory basis of the standard.

Clause 6 specifies the construction of the mandatory die wrapper register.

Clause 7 defines the optional flexible parallel port: a mechanism to add a higher bandwidth channel to the die stack.

1.6 Word usage

The word *shall* indicates mandatory requirements strictly to be followed in order to conform to the standard and from which no deviation is permitted (*shall* equals is *required to*).[3,4]

The word *should* indicates that among several possibilities one is recommended as particularly suitable, without mentioning or excluding others; or that a certain course of action is preferred but not necessarily required (*should* equals is *recommended that*).

The word *may* is used to indicate a course of action permissible within the limits of the standard (*may* equals is *permitted to*).

The word *can* is used for statements of possibility and capability, whether material, physical, or causal (*can* equals is *able to*).

2. Normative references

The following referenced documents are indispensable for the application of this document (i.e., they must be understood and used, so each referenced document is cited in text and its relationship to this document is explained). For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments or corrigenda) applies.

IEEE Std 1149.1™-2013, IEEE Standard for Test Access Port and Boundary-Scan Architecture.[5,6]

IEEE Std 1500™-2005, IEEE Standard Testability Method for Embedded Core-based Integrated Circuits.

IEEE Std 1687™-2014, IEEE Standard for Access and Control of Instrumentation Embedded within a Semiconductor Device.

3. Definitions, acronyms, and abbreviations

3.1 Definitions

For the purposes of this document, the following terms and definitions apply. The *IEEE Standards Dictionary Online* should be consulted for terms not defined in this clause. [7]

active functional interface: An active functional interface includes all of the terminals under test while using the die-wrapper register (DWR).

apply event: A derivative event that coincides with the Shift, Capture, or Update event, whereby test data become active and effective as test stimuli. For example, while the wrapper is in IF mode, the apply event is when test data is applied from input cells onto the internal die functional logic.

[3]The use of the word *must* is deprecated and cannot be used when stating mandatory requirements, *must* is used only to describe unavoidable situations.
[4]The use of *will* is deprecated and cannot be used when stating mandatory requirements, *will* is only used in statements of fact.
[5]The IEEE standards or products referred to in Clause 2 are trademarks owned by the Institute of Electrical and Electronics Engineers, Incorporated.
[6]IEEE publications are available from the Institute of Electrical and Electronics Engineers (http://standards.ieee.org/).
[7]*IEEE Standards Dictionary Online* is available at: http://dictionary.ieee.org. An IEEE Account is required for access to the dictionary, and one can be created at no charge on the dictionary sign-in page.

bidirectional terminal: A terminal that can be controlled to switch between driving or receiving signals.

capture event: An event whereby the value present on the cell function input (CFI) or cell functional output (CFO) is stored in a storage element in the shift path.

capture mode: The mode in which all required die-wrapper register (DWR) cells apply the capture event. For instance, during an external test mode, only the DWR cells attached to die terminal inputs are required to capture, all other die-wrapper register (DWR) cells can apply any event.

cell test input (CTI): A die-wrapper register (DWR) cell's test data input.

cell test output (CTO): A die-wrapper register (DWR) cell's test data output.

chip: Active component typically mounted on a board or some other package with other components. It can be a packaged die or die stack (often a user does not need to know whether his/her chip contains one die or multiple dies stacked together). The more general term "component" is normally used in this standard for a compliant object as there may be multiple integrated circuits in a package, even when they behave as a single object. *Syn:* **integrated circuit (IC)**.

component: An active or passive electronic part. For the sake of this standard, this usually refers to an integrated circuit, although it could include non-integrated-circuit devices mounted on a board. See also: chip.

core: A functional circuit design that is part of a single die. In case of non-hierarchical design of the die's circuitry, all functional circuitry on that die is part of the core. In case of hierarchical design and test architecture of the die's circuitry, there are multiple embedded cores on a single die, which are assumed to be equipped with an IEEE Std 1500™ core-test wrapper.

core test: A test methodology that is applied to an embedded core.

dedicated shift path: A shift path comprising storage elements that do not participate in functional operation.

dedicated wrapper (cell): A wrapper style that does not share hardware with core functionality. This style allows certain test operations to occur concurrently and transparently during functional operation. This definition could apply to individual cells.

die: A (typically rectangular) piece of processed semiconductor substrate, containing circuitry. A die is the result of dicing a semiconductor wafer after processing.

die sequence: The first die in the stack is accessible to the outside world. The following dies are referred to as the second, third, etc. The sequence of the dies does not imply any orientation of the dies. A stack can have multiple dies that are last dies; this is called a multi-tower stack.

die stack: A construction in which multiple individual dies are stacked and electrically interconnected, such that together they form a bigger circuit.

die terminal: The physical connection point between dies or between the first die and the interposer or PCB.

die-wrapper register (DWR): The register surrounding the die for facilitating die test or die-to-die tests.

external safe state: A configuration of safe register values in which the outputs of a die are in a state that prevents them from interfering with a block of logic outside the die. See also internal safe state; safe state.

EXTEST: This is a shortening of the two words external test.

external test: Occurs when the die-wrapper register (DWR) is used to test logic or interconnect external to the die while the DWR is in outward-facing mode. This may include logic on the die if the DWR cell is not at the die boundary.

falling-edge: A transition from a high to a low logic level. In positive logic, a change from logic 1 to logic 0. Events that are specified to occur on the falling edge of a signal should be completed within a fixed (frequency independent) delay specified by the component supplier.

first die: The first die is the die which is connected to the printed circuit board (PCB) or an equivalent carrier or interposer. It is typically the bottom die of the stack.

first die external-I/O: Package-level connection to the board or system level.

flexible parallel port (FPP): Optional multi-bit test access mechanism allowing test data and protocols to be transferred over a set of FPP lanes.

flexible parallel port (FPP) channel: A set of identical FPP lanes controlled by common signals.

flexible parallel port (FPP) clock lane: An FPP lane used to provide an FPP lane clock to registered FPP lanes.

flexible parallel port (FPP) configuration element: A test access port (TAP)-accessible test data register (TDR) bit which controls the functionality and configuration of the FPP channels.

flexible parallel port (FPP) lane: A collection of lane paths connecting: a primary interface terminal, a secondary interface terminal, die-level DfT, and/or other lanes.

flexible parallel port (FPP) lane clock (signal): Clock signal used to drive registration within registered FPP lanes.

flexible parallel port (FPP) lane path: A single-bit logical connection between a lane input terminal (source) and a lane output terminal (sink). This connection may contain sequential elements.

fully-provisioned (cell): A die-wrapper register (DWR) cell with full functionality, including capture, shift, and control features. In addition, it may have safe or update features.

input cell: A die-wrapper register (DWR) cell that is provided on a die input.

input pin: A component pin that receives signals from an external connection.

instruction: A binary data word shifted serially into the test logic in order to define its subsequent operation.

integrated circuit (IC): A collection of transistors, resistors, and capacitors and their interconnections constructed to perform specific functions on a single thin slice of a semiconductor crystal.

internal safe state: A configuration of safe register values whereby a core is protected from the impact of a test outside the core. See also external safe state; safe state.

interface: The collection of terminals connecting to another die or package substrate.

INTEST: is a shortening of the two words internal test.

internal test: occurs when the die-wrapper register (DWR) is used to test logic on the die while the DWR is in inward-facing mode.

inward-facing (IF) mode: The test mode where core inputs are controlled by the die-wrapper register (DWR) and core outputs are observed by the DWR.

last compliant die: Die number k in a stack with n dies of which the first k dies are compliant with this standard ($1 \leq k \leq n$).

last die: This is the last die in the stack. Last die refers to the logical position in the test hierarchy.

least significant bit (LSB): The digit in a binary number representing the lowest numerical value. For shift registers, the bit located nearest to the serial output, or the first bit to be shifted out. The least significant bit of a binary word or shift-register is numbered 0.

mission mode: The mode in which the die-wrapper register (DWR) does not interfere with the functional operation of a wrapped die.

most significant bit (MSB): The digit in a binary number representing the greatest numerical value. For shift registers, the bit farthest from the serial output, or the last bit to be shifted out. Logic values expressed in binary form are shown with the most significant bit on the left.

multi-tower stack: A system with a multi-tower design has a structure, where one base die is connected to two or more other dies or die stacks, where these dies or die stacks are not connected to each other.

no-connect: A signal of the component, which is brought to an input-output pad of the die but not connected to a package pin due, for instance, to a constrained pin-count for the package.

output cell: A die-wrapper register (DWR) cell that is provided for a die output.

output pin: A component pin that drives signals onto external connections.

outward-facing (OF) mode: The test mode where wrapper functional outputs are controlled by the die-wrapper register (DWR) and wrapper functional inputs are observed by the DWR.

parallel configuration: Selected architecture of the die-wrapper register (DWR) test logic wherein more than one serial input and serial output are used to access multiple DWR segments during the test.

partially-provisioned (cell): Die-wrapper register (DWR) cell with less than the maximum functionality. It supports the Shift event and either the Capture event or the Apply event. In addition, it may have safe or update features. *See also:* f**ully-provisioned.**

pin: The point at which connection is made between the integrated circuit and the substrate on which it and other components are mounted (e.g., the printed circuit board). For packaged components, this would be the package pin; for components mounted directly on the substrate, this would be the bonding pad. The actual form of connection (bonding wire, landing pad, solder ball, or metal pin inserted into a via) is not material to the definition. S*ee also:* **terminal.**

port: A single signal, terminal, or pin; or collection of signals, terminals, or pins.

power-up reset: A reset of test or system logic that occurs when the power supply goes from OFF to ON. This can be achieved by design of latches that causes them to power up in a specific state, or by an ON-component or OFF-component circuit that generates an asynchronous reset signal that stays active for some nontrivial time after the power supply has reached its operating voltage. *Synonym:* **power-on reset (POR)**.

primary interface: The interface connecting towards the first die or to the package substrate.

register: A set of typically contiguous or related storage elements.

reset: The establishment of an initial logic condition that can be either logic 0 or logic 1, as determined by the context.

rising-edge: A transition from a low to a high logic level. In positive logic, a change from logic 0 to logic 1. Events that are specified to occur on the rising edge of a signal should be completed within a fixed (frequency independent) delay, specified by the component supplier.

sample: Action to capture the value of a signal at a specific moment in time, such as defined by a clock edge.

safe data: Data that satisfy safe state configuration requirements. These data are user-defined.

safe state: A property whereby a test of one block of logic is prevented from interfering with or damaging another block of logic. See also external safe state; internal safe state.

scan path: The shift-register path through a circuit designed using the scan design technique.

secondary interface: An interface connecting away from the first die or package substrate.

segment: A set of contiguous (in terms of the scan chain) register bits obeying the rules for test data registers and from which a test data register may be assembled.

segment insertion bit (SIB): A configuration of at least a shift and an update cell and a scan-multiplexor that is used to include or exclude a scan chain segment from the active scan path.

selected test data register: A test data register is selected when it is required to operate as instructed by associated test logic.

shared wrapper (cell): The wrapper style that shares logic between the test and mission modes of operation of a system storage element. This style prevents simultaneous functional and test operation uses of the shared storage element.

shift event: An event whereby the data stored in the die-wrapper register (DWR) shift path are advanced one storage position closer to the DWR's test output (TO). The data present at the DWR's test input (TI) are loaded into the shift path storage element closest to the DWR's TI. If multiple DWR segments are implemented, this definition applies individually to each DWR segment.

shift mode: This mode occurs while the shift event is applied to all die-wrapper register (DWR) cells with the intent of loading/unloading DWR scan chains.

stackID: A field in an electronic chip identification (ECID) that describes the position of the die in the stack.

storage element: A bi-stable circuit element, such as a latch or a flip-flop, which retains state until certain enabling and/or clocking conditions are met that cause it to take on a new state.

system logic: Any circuitry that is dedicated to realizing the non-test function of the component or is at the time of interest configured to achieve some aspect of the non-test function.

terminal: A logical endpoint from which connections can be made between dies, hierarchical modules, and leaf cells. A terminal can be unidirectional or bidirectional. *See also:* **pin**.

test access mechanism (TAM): A feature of a system-on-chip (SoC) design that enables the delivery of test data to and from dies, cores, or core wrappers.

test access mechanism (TAM) harness: Die wrapper register (DWR) logic that enables the coupling of a TAM to cell test inputs (CTIs) and cell test outputs (CTOs).

test input (TI): The serial test data input of a die-wrapper register (DWR) segment.

test logic: Any item of logic that is a dedicated part of the test logic architecture or is at the time of interest configured as part of the test logic architecture.

test mode: The state of a component in which the component's test logic interferes with the flow of signals to and from the system logic. In addition, the system logic may be controlled as needed to prevent an undesired response to system inputs, excessive heating, and so on.

test output (TO): The serial test data output of a die-wrapper register (DWR) segment.

three-state: An adjective describing the ability of an output pin capable of being in one of 3 states: driving a logic 1 or 0, or not driving and inactive (for example, at high impedance).

through silicon via (TSV): A via that electrically connects the front side to the back-side of a (thinned-down) wafer substrate.

update: Transfer a logic value from the shift-register stage of a data register cell or an instruction register cell into the latched parallel output stage of the cell.

update event: An optional event whereby data stored in a die-wrapper register (DWR) cell's shift path storage element is loaded into an off-shift-path storage element (e.g., update register).

update register: The register used to prevent the outputs of a shift register from propagating to other circuitry during a Shift operation. After shifting is complete, the content of the associated shift register is parallel-loaded (updated) into the update register or otherwise applied.

wrapper: A wrapper is a test access and isolation ring at the boundary of a test module, being either an embedded core or a stacked die. A wrapper is typically based on scannable flip-flops.

3.2 Acronyms and abbreviations

ATE	automatic test equipment
ATPG	automatic test pattern generation
AUXCK	auxiliary clock
BiDi	short for bidirectional
BGA	ball grid array (packaging style)
BIST	built-in self-test
BSDL	Boundary-Scan Description Language or a file containing BSDL statements
CFI	cell functional input
CFO	cell functional output
CMOS	complementary metal-oxide semiconductor
CTI	cell test input
CTO	cell test output

dc	direct current
DFI	die functional input
DFO	die functional output
DfT	design-for-test
DWR	die wrapper register
ECID	electronic chip identification (ID)
FPP	flexible parallel port
IC	integrated circuit
ICL	Instrument Connectivity Language
IF	inward facing
I/O	input/output
IP	intellectual property
LSB	least significant bit
MSB	most significant bit
OF	outward facing
PDL	Procedural Description Language or a file containing PDL statements
PLL	phase-locked loop
POR	power-on reset
PTAP	primary test access port
SIB	segment insertion bit
SoC	system on chip
STAP	secondary test access port
TAM	test access mechanism
TAP	test access port
TCK	test clock
TDI	test data input
TDO	test data output
TDR	test data register
TI	test input
TMS	test mode select
TO	test output
TSV	through silicon via
WIR	wrapper instruction register
WRCK	wrapper clock

4. Technology

4.1 Stack model

IEEE Std 1838 is defined at the logical level and abstracts from the physical implementation of the stack. To that end, it avoids terms such as ‘top’, ‘bottom’, ‘up’, and ‘down’ (which intuitively seem obvious terms to use in the context of vertical die stacking). Instead, IEEE Std 1838 has adopted terminology based on the logical position of the interconnects relative to the external stack I/Os, which the standard assumes are 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. IEEE Std 1838 assumes that every die has a single primary interface, which connects to its unique previous die; an exception is the first die, for which its primary interface connects to the external system logic. Relative to the external stack I/Os, an adjacent die connected to this die in the opposite direction, (i.e., *away* from 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 stacked 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*.

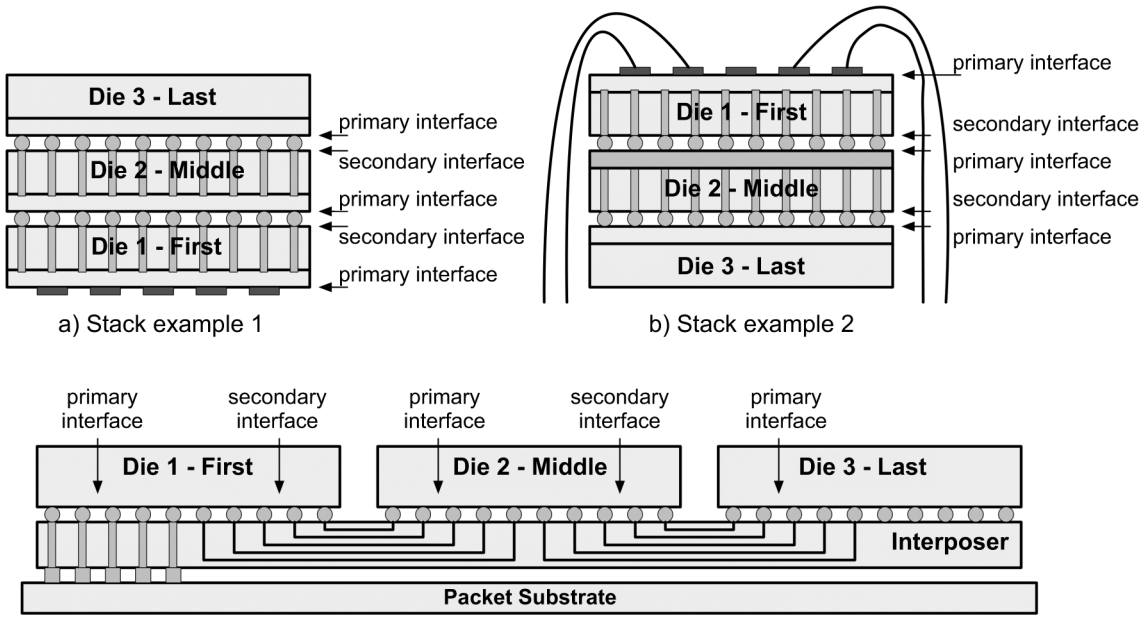

Figure 2—Three physically different implementations of a logically equivalent stack

Figure 2 depicts three different physical implementations of what are for IEEE Std 1838 logically equivalent stacks. The stacks consist 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 Figure 2 part a) and Figure 2 part b) are both single-tower 3D stacks. In Figure 2 part a), all dies are face-down (i.e., with the processed, active side 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 part 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, in Figure 2 part b), 'up' and 'down' are reversed in comparison to Figure 2 part 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 part 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.

4.2 Wafer-level die access

For components containing multiple stacked dies, we distinguish the following moments during the manufacturing flow (including wafer processing, stack assembly, and packaging) at which tests for manufacturing defects can be performed: (1) *pre-bond tests* prior to stacking, (2) *mid-bond tests* on incomplete, partial stacks, (3) *post-bond tests* on complete, though not yet packaged, stacks, and/or (4) *final tests* on the final packaged product. At each of these moments, it might be possible (but certainly not mandated by IEEE Std 1838) to test dies and inter-die interconnects for manufacturing defects incurred during wafer processing, stack assembly, and/or packaging. The number of possible test flows grows quickly with the number of dies in the stack and determining a test flow that optimizes test quality and reduces test cost for given quality and cost parameters is therefore a complex task that typically requires automated trade-off evaluation and optimization.

IEEE Std 1838 focuses on standardizing die-level DfT features such that when compliant dies are brought together in a die stack, a minimum of test access is guaranteed. The benefits of the 3D-DfT in compliant dies apply mainly at those test moments after at least one stack-assembly operation has been performed, i.e., from mid-bond testing onwards. Pre-bond testing takes place prior to stacking, and hence does not require cooperative test access via standardized, compatible DfT through other dies. However, to limit the associated silicon area and other implementation costs, it is preferred to reuse the 3D-DfT architecture as provided in the various dies in the stack as much as possible, so also during the pre-bond test of these dies.

Pre-bond test may be distinctly different from the stack test because the semiconductor test equipment accesses the die-under-test via wafer-level probing. For die stacks, probe access is typically via the stack's external I/O interface, which consists of a regularly-sized interface to the package substrate and/or the package pins. The functional I/Os on to-be-stacked dies prior to stacking normally comprise large arrays of fine-pitch micro-bumps. Typical examples are given by JEDEC standards JESD229 ('Wide I/O Single Data Rate'), JESD229-2 ('Wide I/O 2'), and JESD235B ['High Bandwidth Memory DRAM (HBM1, HBM2)'], which contain micro-bump arrays of 1200, 1752, and 4955 micro-bumps, respectively, at pitches of 40/50 μm, 40/40 μm, and 55 μm, respectively.

For pre-bond tests of dies accessed via such large fine-pitch micro-bump arrays, there are essentially three options:

— Do not execute pre-bond tests. However, this might imply that compound yields of stacked dies are unacceptably low and hence product costs soar.

— Use advanced MEMS-type probe cards on these large-array fine-pitch micro-bumps.

— Provide dedicated pre-bond probe pads. This requires additional design effort and silicon real estate, but does make pre-bond testing feasible.

The DfT infrastructure in non-first dies is connected via micro-bump connections to the previous die for mid-bond, post-bond, and final testing. In the latter case, the same on-die DfT infrastructure now also needs to be connected to the dedicated pre-bond probe pads. This requires a mechanism to select between (1) pre-bond test access via dedicated pre-bond probe pads and (2) mid-bond and later test access via micro-bump connections coming from the previous die in the stack.

Figure 3 and Figure 4 show two alternative implementations for the above-mentioned selection mechanism. In Figure 3, a die detector switches between dedicated probe pads and micro-bump connections from/to the previous die. The solution in Figure 4 eliminates the need to provide a dedicated "post-bond" control signal from the previous die to this die, as it uses the fact that VDD is connected between the previous die and this die as a detector for the post-bond situation.

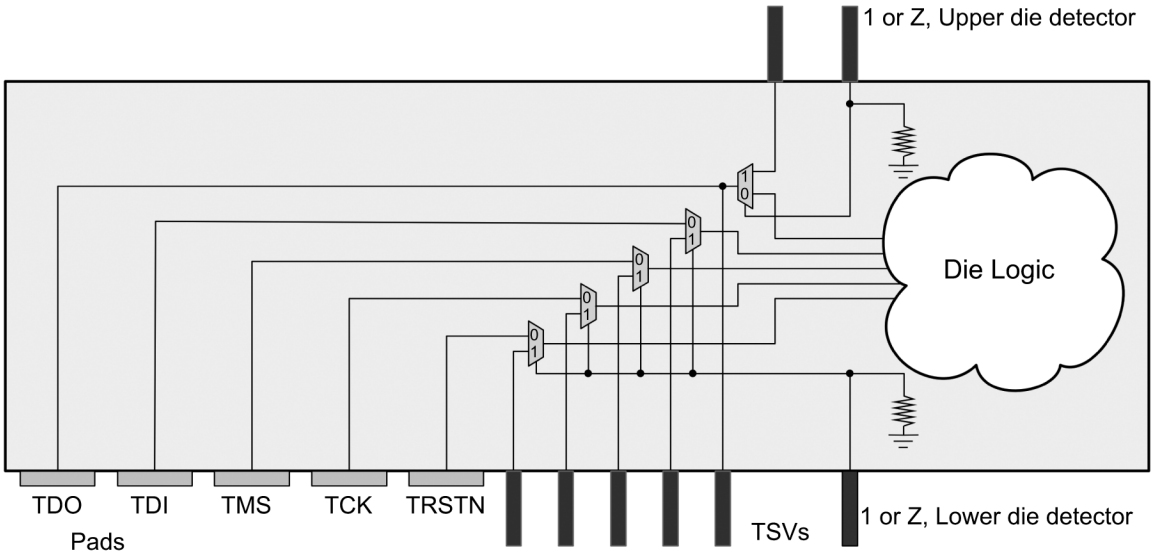

Figure 3—Die detectors switching mechanism proposal

Dedicated probe pads (sacrificial pads shown in Figure 4) to support pre-bond testing and a selection mechanism for them are not part of IEEE Std 1838, as it does not affect the interoperability of the test access architecture in die stacks. The decision to provide them (or not), and if so, how exactly to implement them, is a per-die decision, which does not affect the testability of other dies in the stack. It is certainly possible to combine dedicated pre-bond probe pads and compliance to IEEE Std 1838.

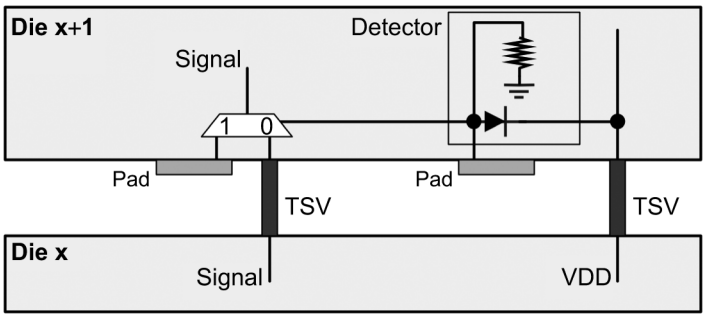

Figure 4—Power-rail-based path selection

4.3 Physical attributes

In addition to the exclusions mentioned in the previous subclause, physical aspects of construction are also not addressed by this standard. It is assumed that any die stacked upon another die will have aligned micro-bumps and other physical geometries that allow such a prospect to occur. As such, it is likely that port directionality

and other design attributes will also be shared amongst teams participating in building the stack so that their design goals can be met. But this standard will only focus on the logical design to enable interoperability between die in the stack and tools interfacing to the databases produced by such a flow and methodology.

5. Serial test access ports

Each die is a separate logical plug-and-play entity enabling a 3D component to be assembled with test and debug features. Being a per-die standard, terminals and terminal behavior will be defined. The mating of terminals from one die within a 3D stack to the complementary terminals of an abutting die enables a contiguous test and debug access architecture that spans multiple die within the stack.

It should be noted, though, that this standard defines the functional, behavioral, and logical but not physical terminal connections—defining the physical location of terminals, pads or bumps and the sizing and pitch of physical connections is beyond the scope of this standard.

The goal of this clause is to define the signals/terminals, terminal behavior, and associated logic associated with the serial access and control of test and debug features on each die, and how those features are extended by abutting, bonding, or otherwise connecting to other die.

As shown in Figure 5, one subset of the primary interface shall be the primary test access port (PTAP) that will contain the signals and internal die logic connections that are associated with the Primary Interface; and a subset of the secondary interface will be the one or more secondary test access ports (STAPs) that will contain the signals and internal die logic connections that are associated with the secondary interface. Note that the primary and secondary interfaces may also include an optional flexible parallel port (FPP), which will be further discussed in Clause 7.

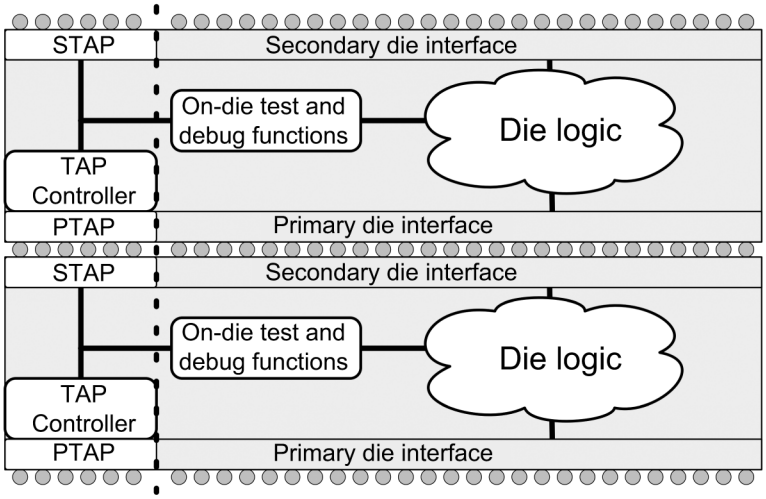

Figure 5—Example of a stack comprised of two semiconductor dies

5.1 Primary test access port

In support of a per-die standard, the mandatory test interface and access to all on-die test and debug features is the serial PTAP. The PTAP is associated with the surface closest to the board connection or package interface and is represented by five die terminals: TCK, TMS, TDI, TDO, and TRSTN.

5.1.1 Specifications

The rules are as follows:

a) Each compliant die shall have a serial test access port as part of its primary interface and it shall be referred to as the PTAP.

b) The PTAP shall consist of five signals named TCK, TMS, TDI, TDO and TRSTN.

c) The TCK signal shall be a clock input signal that synchronizes test architecture operations and registers.

d) The signal presented at TMS (Test Mode Select) shall be sampled into the test logic on the rising edge of TCK.

e) The signal presented at TDI (Test Data Input) shall be sampled into the test logic on the rising edge of TCK.

f) Changes in the state of the signal driven through TDO (Test Data Output) shall occur only on the falling edge of either TCK or upon assertion of Reset*.

g) The TRSTN (Test Reset) signal shall be an asynchronous active-low control input signal which provides an asynchronous reset function to test registers defined by this standard.

h) The TCK, TMS, and TDI signals shall be driven from dedicated terminals.

i) The TDO signal shall drive a dedicated terminal.

j) The TRSTN signal for a first die shall be driven from either a dedicated terminal or logic providing a power-on-reset function (POR).

k) The TRSTN signal for a non-first die shall be driven from a dedicated terminal.

The recommendation is as follows:

l) The TRSTN signal for a first die should be driven from a dedicated terminal.

5.1.2 Description

The five die terminals represent the signals normally associated with a compliant IEEE 1149.1 interface—and in the case of an IEEE 1838 die used as a first die (i.e., a die nearest to the board attachment), the five die terminals may actually represent the IEEE 1149.1 TAP interface to the board connection if an IEEE 1149.1 compliance mandate is required (note that IEEE 1838 compliance does not require IEEE 1149.1 compliance—IEEE 1149.1 compliance is only mandated if there is a separate IEEE 1149.1 compliance imposed on the final component). The five signals are: the input test clock, TCK, that synchronizes all test architecture registers; the test mode select, TMS, that provides the input control signal to the test controller finite-state-machine (FSM) that generates the operation protocol for all test architecture registers; the active-low test reset, TRSTN, that provides the input asynchronous reset signal that can be used to put all test architecture registers into a known default state; the serial test data input, TDI, that allows serial data to be delivered to test architecture data registers on the rising edge of the TCK clock when the test controller FSM is in one of the shift states; and the serial test data output, TDO, that allows serial data from the test architecture data registers to be presented to the die interface on the falling edge of the TCK clock when the test controller FSM is in one of the shift states.

TMS, TCK, TDI and TDO are connected to dedicated test terminals. For a first die, the TRSTN signal should be connected to a dedicated test terminal unless there is POR logic that can drive this signal.

5.2 Primary test access port controller

The PTAP signals drive the PTAP controller, which is an IEEE 1149.1 compatible TAP controller. The register architecture associated with the PTAP controller includes registration to support the 3D features that are required or optional in IEEE Std 1838.

5.2.1 Specifications

The rules are as follows:

a) The PTAP signals shall be connected to a PTAP controller and associated register architecture.

b) The PTAP controller shall operate in accordance with the sequence of operations defined by the IEEE 1149.1 TAP controller finite-state-machine based on the TMS input control signal and synchronized by the TCK input clock signal (see 6.1 of IEEE Std 1149.1-2013).

5.2.2 Description

Like the IEEE 1149.1 TAP, the PTAP can access any number of die-internal registers available in the stack. However, the PTAP controller register architecture shall, at a minimum, have the bypass register, one or more die-wrapper register (DWR) segments, the 3D configuration register (3DCR), and all required instructions that support these registers.

The PTAP Controller operates in the same manner as the IEEE Std 1149.1 TAP controller described in 6.1 of IEEE Std 1149.1-2013. Figure 6 shows the PTAP controller state diagram and names the states with hexadecimal numbering which coincides with the waveforms of the PTAP controller shown in Figure 7.

NOTE—The hexadecimal state numbering of the PTAP Controller shown here is for illustration purposes only. The actual state numbering of the PTAP controller is determined by the designer.[8]

[8]Notes in text, tables, and figures of a standard are given for information only and do not contain requirements needed to implement this standard.

Controller state	DCBA (hex)
Exit2-DR	0
Exit1-DR	1
Shift-DR	2
Pause-DR	3
Select-IR-Scan	4
Update-DR	5
Capture-DR	6
Select-DR-Scan	7
Exit2-IR	8
Exit1-IR	9
Shift-IR	A
Pause-IR	B
Run-Test/Idle	C
Update-IR	D
Capture-IR	E
Test-Logic-Reset	F

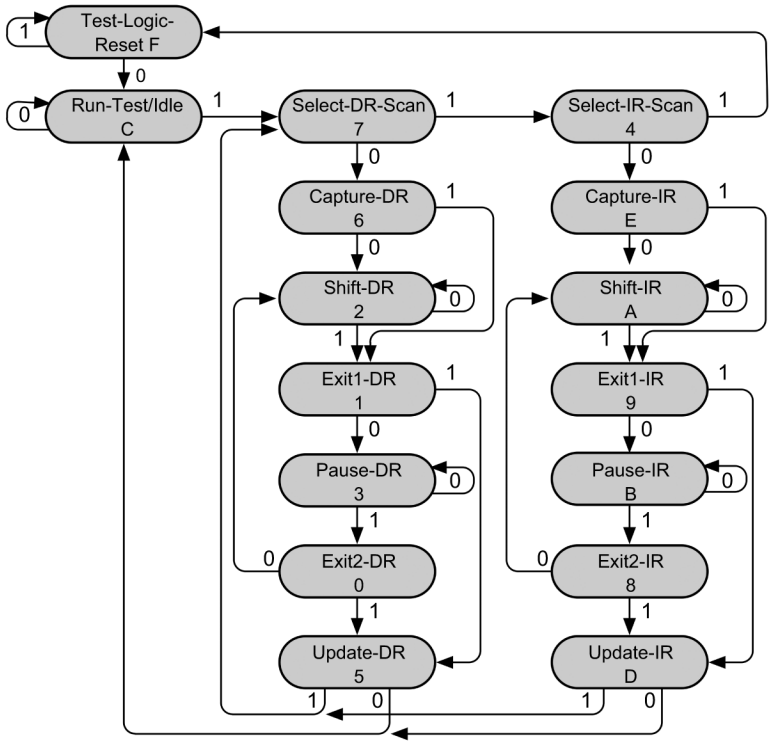

Figure 6—PTAP controller state machine diagram and state definitions

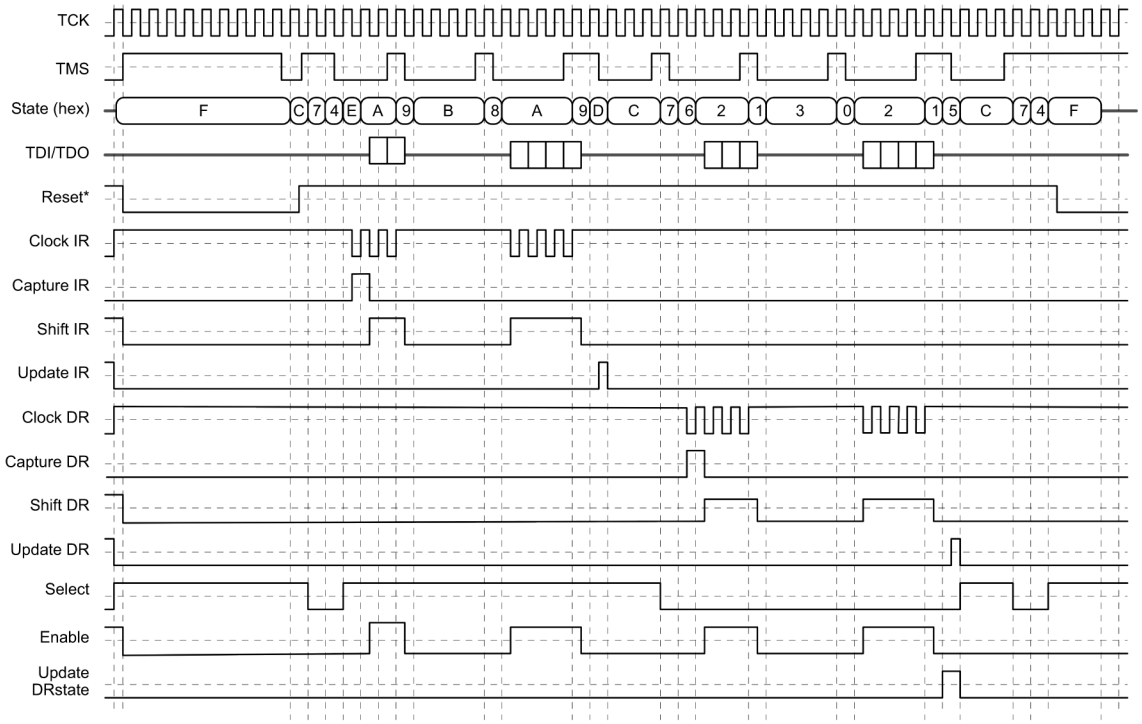

Figure 7—PTAP controller waveforms

5.3 Secondary test access port (STAP)

The baseline IEEE 1149.1 architecture is extended to the secondary interface with the addition of STAPs and their associated selection and configuration control logic. This control logic is composed of the STAP itself, and control from the Primary TAP Controller's three-dimensional configuration register (3DCR, see 5.5.1). The Secondary Interface may support multiple STAPs, each with a mandated five-pin test interface enabling it to plug into a next die's PTAP.

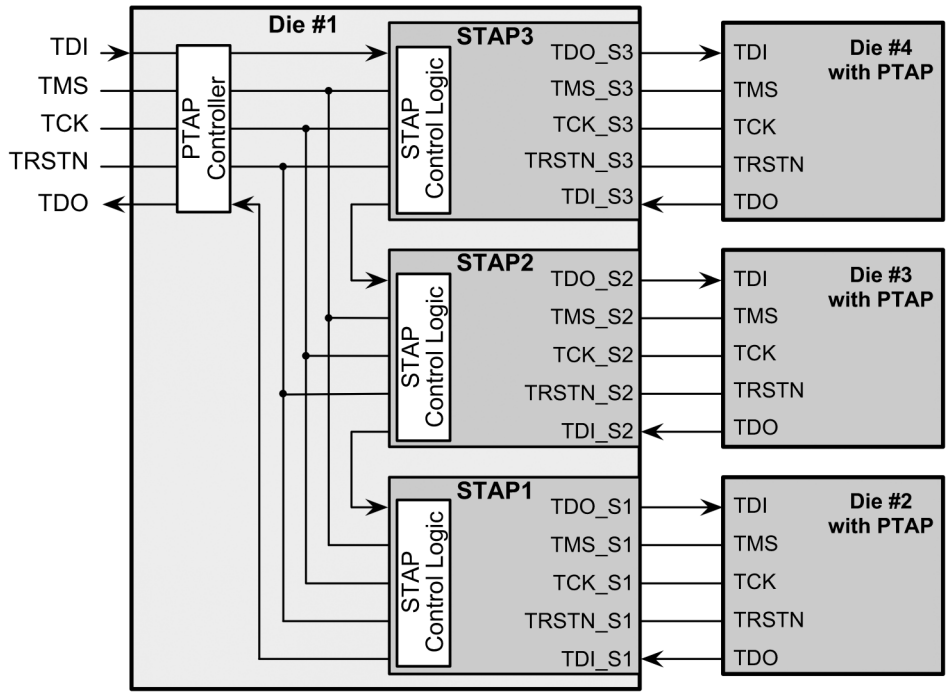

Figure 8—Multiple numbered STAPs connected to a stacked next die

In this standard, an STAP shall be referred to as STAP*n*, with *n* beginning at 1. As shown in Figure 8, the STAPs will be numbered from 1 to *n* starting from the TDO PTAP signal and tracing backwards to the TDI PTAP starting point of the serial scan chain. If more than one STAP exists, then the numbering will increment in scan path order from the PTAP TDO terminal toward the PTAP TDI terminal. Therefore, the STAP named STAP1 will be connected closest in scan path order to the PTAP TDO terminal.

5.3.1 Specifications

The rules are as follows:

a) Each die shall have N selectable STAPs, one for each directly connected die.

NOTE—N may be zero if the die does not support any next dies to be stacked.

b) Each STAP shall comprise the terminals named TDI_S*n*, TDO_S*n*, TMS_S*n*, TCK_S*n*, and TRSTN_S*n* where *n* represents the STAP number, where $0 < n \leq N$.

c) The TCK_S*n* output terminal shall be driven from the PTAP TCK terminal with no added in-line registration or combinational gating logic.

d) The TRSTN_S*n* output terminal shall be driven from the PTAP TRSTN signal with no added in-line registration or combinational gating logic.

e) The TMS_S*n* output terminal, when driven by the PTAP TMS terminal, shall be driven with no added in-line registration.

f) The TDI_S*n* input terminal shall sample data on the rising edge of TCK.

g) The TDO_S*n* output terminal shall present new data on the falling-edge of TCK.

The permission is as follows:

h) It is permitted to buffer signals driving the TCK_S*n*, TMS_S*n* and TRSTN_S*n* terminals for drive strength.

5.3.2 Description

The TCK and TRSTN signals are delivered directly to the STAP, as is, except for some buffering (but no additional in-line registration is allowed). They should always be enabled to the next die whether or not the STAP is selected. The TMS, TDI, and TDO signals are only enabled to or from the next die when the STAP is selected. Each of these signals has at least one multiplexer or in-line gate delay as part of the handling process to enable the selection and deselection.

The STAP terminals drive terminals on the secondary interface of the current die that then connect to their associated PTAP terminals on the next die. TCK_S*n*, TMS_S*n*, and TRSTN_S*n* outputs drive the associated PTAP terminals on the next die: TCK, TMS, and TRSTN, respectively. The STAP serial data terminals, the TDO_S*n* output and the TDI_S*n* input, are meant to, respectively, drive and receive the next die PTAP terminal counterparts, the TDI input and TDO output.

TDI_S*n* and TDO_S*n* also require registration, as they react on the rising or falling edge of TCK_S*n*. TDI_S*n* should sample data on the rising edge of TCK_S*n* and TDO_S*n* should present data on the falling edge of TCK_S*n*.

5.4 Secondary test access port control logic

The STAP Control Logic (or STAP, for short) consists of several sub-components: the selector and the retiming elements.

5.4.1 Specifications

The rules are as follows:

a) Each STAP shall be selectable using the corresponding Select_S*n* signal from the 3DCR.

b) When the Select_S*n* signal is asserted, it shall cause the STAP to select the associated PTAP of the next die between TDO_S*n* and TDI_S*n*.

c) The first STAP (numbered *k*) shall source TDO_S*k* from TDI_S*k*_int of the PTAP on the falling edge of TCK.

d) STAP*n* ($n < k$) shall source TDI_S*n*_int from TDO_S*n*+*1*_int of STAP*n*+*1*.

e) If any STAPs are selected (at least 1 Select_S*n* is asserted) the PTAP shall source TDO from TDO_S1_int of STAP1.

NOTE—Refer to Figure 8 for reference numbering of STAPs.

f) The last STAP (numbered *1*) shall connect TDO_S*1*_int to a point prior to the hold-time element (clocked by the falling edge of TCK) of the PTAP TDO.

g) TDO_S*n*_int shall be driven by a register element (clocked by the rising edge of TCK only in the ShiftDR and ShiftIR states) with data driven by TDI_S*n* or TDI_S*n*_int according to the value of Select_S*n*.

NOTE—ShiftDR and nd ShiftIR is represented in Figure 9 by the Enable signal. The Enable signal is the OR of the Shift IR and Shift DR signals (see Figure 7).

h) TDO_S*n* shall be driven by a hold-time element (for example, clocked by the falling edge of TCK) with data driven by TDI_S*n*_int.

i) When STAP_S*n* is not selected (Select_S*n* is 0), TMS_S*n* shall be driven by RTI_or_TLR_S*n* (see 5.5.1).

j) When STAP_S*n* is selected (Select_S*n* is 1), TMS_S*n* shall be driven by TMS.

k) If no STAPs are selected (all Select_S*n* signals are deasserted) and all of the configuration elements reside in the PTAP controller, PTAP shall source TDO from TDI_S*k*_int (see Figure 10 and Figure 11), else PTAP shall source TDO from TDO_S1_int.

l) Shift register elements in the serial data path shall be clocked by the rising edge of TCK, and hold state in all but the ShiftDR and ShiftIR states.

NOTE—Shift register elements are supported by recommendations m) and n), and permissions o).

The recommendations are as follows:

m) If required, extra shift register elements should be added along the serial data path between the PTAP TDI_S*k*_int output and the STAP_S*k* TDI_S*k*_int input.

n) If required, extra shift register elements should be added along the serial data path between the STAP_S1 TDO_S1_int output and the PTAP TDO_S1_int input.

The permission is as follows:

o) If required, extra shift register elements may be added along the serial data paths within each STAP.

5.4.2 Description

At power-up, the STAPs are de-selected. Prior to being selected and activated, each STAP should be in a configuration that keeps its associated next die's PTAP Controller parked in an idle state so that it processes no Capture, Shift, or Update operations. The default parked state is the Test-Logic-Reset (TLR) state; however, the STAP configuration register (3DCR) allows selection of the Run-Test-Idle (RTI) state.

To activate an STAP, the 3DCR segment for the associated STAP(s) is accessed. The 3DCR bits can then be configured and assert its new configuration on the falling edge of TCK when the PTAP Controller is in the Update-DR state. De-activation requires a similar operation.

Registration within the STAP Control Logic block is required along the serial scan path through each STAP to maintain a negative-edge TCK register to present serial data to the TDO_S*n* terminal that goes off-die and a positive-edge TCK register to sample serial data presented from off-die to the TDI_S*n* terminal. These shift-only timing-adjust (resynchronizing) data bits are used in the scan data path to help ensure that the shift data integrity is maintained.

Any of the STAPs may be active and included in the active scan path; or may be deselected and bypassed and not part of the active scan path.

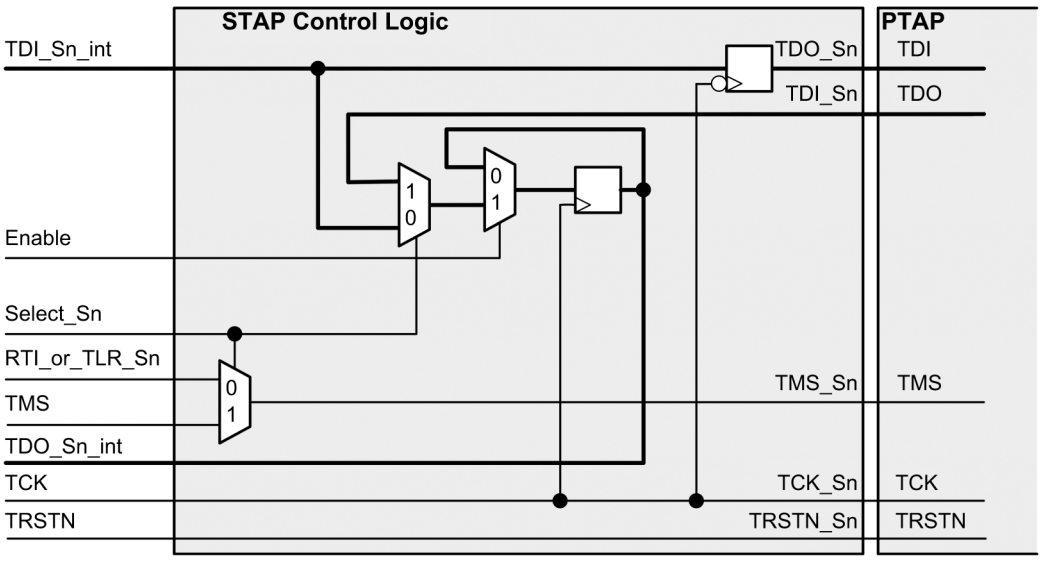

Figure 9—STAP Control Logic

The behavior of the STAP control terminals, when selected to be active, shall follow the port operations of the PTAP and shall be synchronized to the operation of the PTAP. However, when not selected, each STAP may be configured to operate the PTAP of the associated next die to be held in a parked state: Run-Test-Idle (RTI) or Test-Logic-Reset (TLR). The selected parking state depends on the state of the RTI_or_TLR_*Sn* configuration bit for that numbered STAP within the STAP Configuration Register. The selection and parking requirement is accomplished by managing the TMS signal as it passes from the PTAP interface of the current die to the STAP interface on the same die (see the example in Figure 9).

When an STAP is selected, the PTAP TMS, is passed on to the STAP via TMS_S*n*, with no modifications or registration. When an STAP is not selected, the TMS_S*n* should be driven to logic-0 to hold the TAP FSM on the associated next die in a parked Run-Test-Idle (RTI) state or to a logic-1 if the parking state of the TAP FSM on the associated next die is desired to be Test-Logic-Reset (TLR). Note that selecting (un-parking) an STAP mandates a protocol sequence as follows: un-parking requires exiting the UpdateDR state directly to the RTI state.

The STAPs' physical location in relation to each other may incur a large physical separation such that the actual terminals associated with the various TDI_S*n* and TDO_S*n* connections may be connected through long wire routes. Because of these possible impacts to shift-data timing, the positive timing adjust registers should not be isolated to only the secondary TDI_S*n* ports returning from the connected die, but should exist in the main scan path between secondary serial test access ports themselves. Figure 9 shows a preferred configuration where the timing-adjust registers remain in the active scan path whether the secondary serial test access port is selected or deselected.

The STAP should be selected and configured from the on-die IEEE Std 1838 PTAP Controller and Register Architecture and the PTAP Controller should remain in the active serial scan path with the requirement that the source of the scan path to the STAP TDO signal (TDO_S1_int) must be placed at a point after the register architecture of the PTAP register architecture (see the output of the mux that sources TDI_Sk_int in Figure 11).

If the 3DCR is accessed using the Select3DCR instruction as in Figure 10, IEEE 1149.1 compliance is difficult to maintain. For example, the BYPASS instruction datapath would include the bypass register and the STAP registration, as well, even when no STAPs are selected. For this reason, a short path is enabled upon power-up or assertion of the TRSTN pin. This path simply bypasses the STAP paths. This short-cut path is enabled until the user decides to enable an STAP pathway. So, other instructions (other than BYPASS, for example)

will also enable this short path. In effect, the stack will actually resemble an IEEE 1149.1 compliant die until the user enables an STAP. Once and as long as an STAP is enabled, the short path is no longer used, and the STAP registration will be added to the serial data path. Figure 10 shows the decoding of the Select_*Sn* signals (through the OR gate) selecting the stack bypass path when no STAPs are selected.

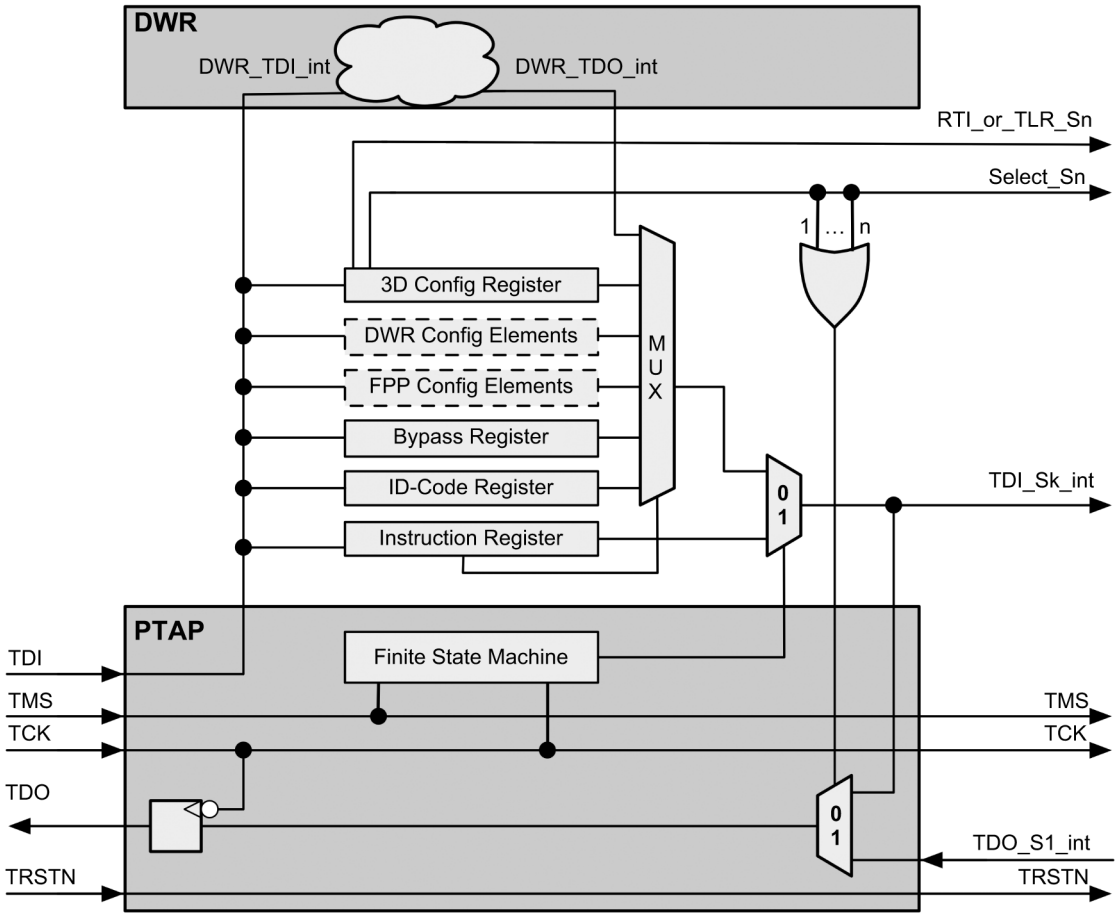

Figure 10—Per-die PTAP and register with signal connections for 3D extensions units and feature Config-Registers

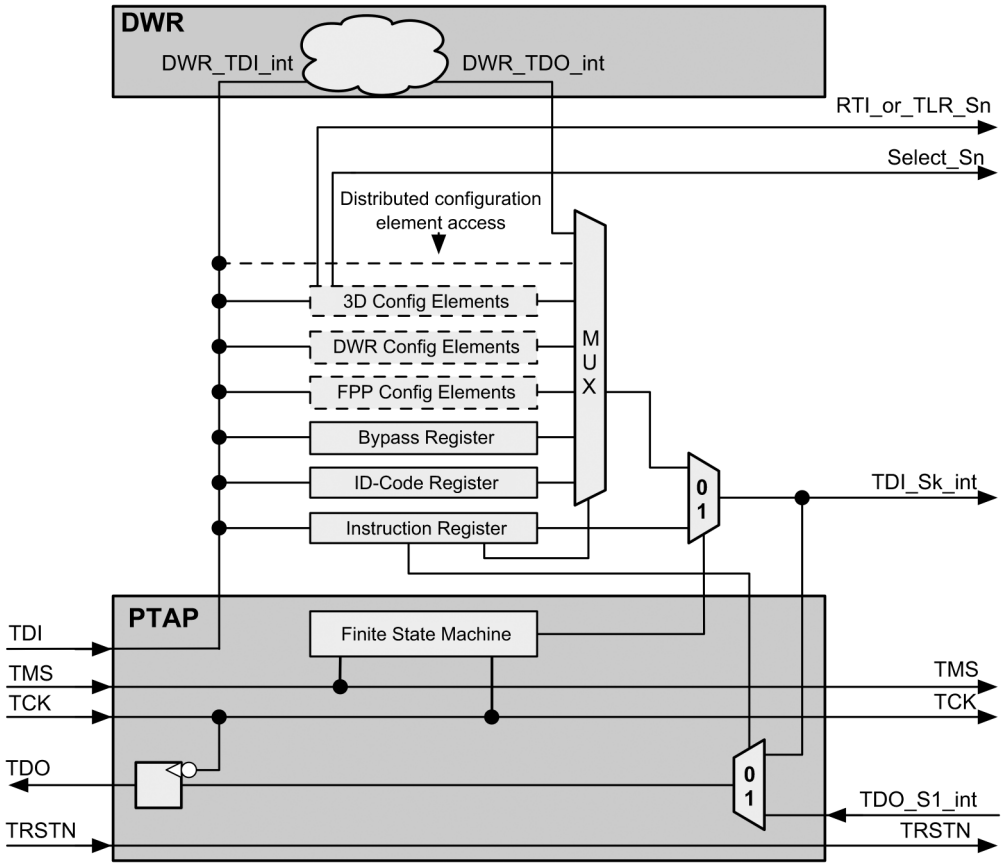

Figure 11—Alternative implementation of the per-die PTAP and register architecture with signal connections for 3D extensions units, feature Config Registers and the distributed configuration element access

It should also be noted that some die could be targeted to be the very last or terminating die of a stack and would therefore not require a Secondary Interface—for this reason, the support of a STAP is designated as an option. The requirement is that a secondary interface may support zero to *n* STAPs.

5.5 Registers

5.5.1 Secondary test access port configuration register (3DCR)

5.5.1.1 Specifications

The 3DCR is the source of three basic types of signals:

— The mandatory 3DCR Config-Hold signal, which resets to a deasserted state (logic 0), that makes STAP configuration bits and the STAPs persistent through the Test-Logic-Reset action of the PTAP controller's FSM;

— The Select_S*n* signals, which reset to a deasserted state (logic 0), that select and activate the individual numbered STAP_S*n;* and

— The RTI_or_TLR_S*n* individual parking state definition signals, which reset to a logic 1 state (corresponding to the TLR state), that defines the parking state of the individually distributed TMS_S*n* signals associated with the individual numbered STAP_S*n*.

All of these signals are generated from a one-hot encoding (logic-1 equals assert) of the update bits, which are matched one-to-one with the shift bits in the 3DCR and have a specified order from the LSB to the MSB of the 3DCR. All assertion and de-assertion actions of these signals occur on the falling edge of TCK in the Update-DR state of the PTAP Controller.

The rules are as follows:

a) Any die containing one or more STAPs shall have a 3DCR.

b) The 3DCR shall be designed as defined in Clause 9 of IEEE Std 1149.1-2013 describing capture-update test data registers (TDR).

NOTE—An example TDR is illustrated within of Figure 9-6 of IEEE Std 1149.1-2013.

c) Each 3DCR shift register element shall have an associated update element.

d) 3DCR register outputs (driven by the update elements) shall change on the falling edge of TCK in the Update-DR state.

e) The 3DCR shall be selected and configured from the on-die IEEE 1838 PTAP Controller and Register Architecture.

f) The 3DCR shall have one selection bit (i.e., Select_S*n*) for each STAP.

g) The 3DCR shall have one TMS hold bit (i.e., RTI_or_TLR_S*n*) for each STAP.

h) After a power-on-reset (POR), or when TRSTN is asserted, or upon entering the TLR state when persistence is disabled, all STAPs shall be deselected with the RTI_or_TLR_S*n* signals configured to drive a logic-1.

NOTE—A logic 1 on RTI_or_TLR_S*n* would cause die connected to the STAP to be held in the Test-Logic-Reset (TLR) state.

i) After a power-on-reset (POR), or when TRSTN is asserted, or upon entering the TLR state when persistence is disabled, all STAPs shall be deselected with the Select_S*n* signals configured to drive a logic-0.

NOTE—A logic 0 on Select_*Sn* would cause die connected to the STAP to be deselected.

j) The 3DCR selection configuration bit (Select_*Sn*) shall be asserted with a logic-1 value.

NOTE—If a secondary interface exists and is not used, its Select_*Sn* bit shall be disabled with a logic-0 value.

k) Setting the RTI_or_TLR_*Sn* bit for STAP_S*n* to logic 0 shall drive STAP_S*n* terminal TMS_S*n* to logic 0 when STAP_S*n* is deselected.

NOTE—This sets the deselected STAP_S*n* in the Run-Test/Idle state.

l) Setting the RTI_or_TLR_*Sn* bit for STAP_S*n* to logic 1 shall drive STAP_S*n* terminal TMS_S*n* to logic 1 when STAP_S*n* is deselected.

NOTE—This sets the deselected STAP_S*n* in the Test-Logic-Reset state.

m) The 3DCR shall have one configuration hold bit (i.e.: Config-Hold_S*n*) for each STAP.

n) Each 3DCR Config-Hold bit shall power-up in the deasserted state (i.e.: the “not holding” configuration).

o) Each 3DCR Config-Hold bit shall be reset to its deasserted state when TRSTN is asserted.

p) When a 3DCR Config-Hold bit is deasserted, its associated STAP logic and 3DCR register bits shall be reset when the PTAP controller enters the Test-Logic-Reset state.

q) When a 3DCR Config-Hold bit is asserted, its associated STAP logic and 3DCR register bits shall not change state when the PTAP controller enters the Test-Logic-Reset state.

r) The 3DCR shall hold its state when the 3DCR is not the selected active data register in the scan path.

The recommendation is as follows:

s) The capture-scan register of each 3DCR bit should capture the value of the update register of the associated update element in the Capture-DR state.

NOTE—One of the reasons for the recommendation is to avoid having unknown data in the capture-scan register transfer to the update register when entering the Update-DR state without passing through the Shift-DR state.

The permissions are as follows:

t) The 3DCR configuring elements can be in series with any other register.

u) A die may have multiple 3DCR segments, each with a separate Config-Hold bit.

5.5.1.2 Description

The STAP 3D Configuration Register (3DCR) is a register selected by an instruction encoding placed in the PTAP Controller instruction register or serially accessible elements. If there are no STAPs on the die, then there is no need for a 3DCR – therefore, it is a dependent option that is mandated only if the die has at least one STAP. A last die does not require an STAP, so may not have a 3DCR.

The STAP 3DCR provides a selection and configuration method to propagate the TAP signals associated with the Primary Interface to the Secondary Interface for use by the next die in the stack. This feature represents the process of passing data and control up and down, and to and from, the next die. The 3DCR may be implemented as multiple register segments, for example in the case where a die has multiple towers.

In the remainder of this standard, 3DCR is used to refer to the configuration and hold bits collectively, even when implemented as multiple register segments.

The 3DCR should have a reset-persistent configuration (Config-Hold), allowing the PTAP controller to pass through the Test-Logic-Reset (TLR) state without affecting the test function. Asserting TRSTN should then reset the 3DCR. Specific rules are defined to enable the Config-Hold function.

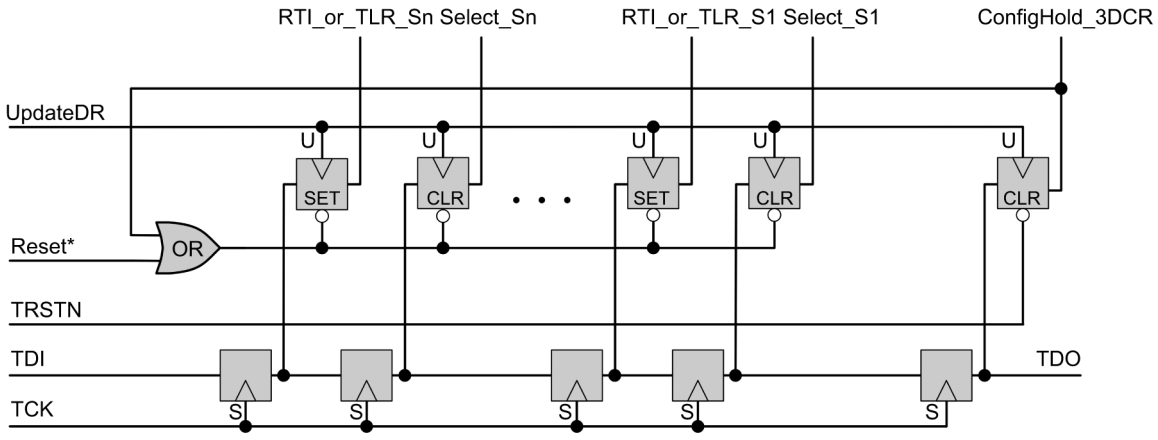

Figure 12—Example STAP configuration register (3DCR)

Figure 12 shows an example of a compliant 3DCR. The command bits of the STAP configuration register are organized in the configuration register beginning with the persistence command bit, ConfigHold_3DCR, at the scan-output end (the LSB). The select signals: the select bits, Select_S*n,* and the hold state bits, RTI_or_TLR_S*n* may be placed anywhere else in the register. The 3DCR is a data-persistent register that will hold its state when the 3DCR is not the selected active data register in the scan path. When the ConfigHold_3DCR bit is asserted, then the 3DCR becomes 'reset persistent', as well, in that the update bits that hold the Select_S*n*, RTI_or_TLR_S*n*, and the ConfigHold_3DCR values will not revert to a reset state when the PTAP Controller FSM passes through the TLR state. A TRSTN assertion will modify all the 3DCR register values, resetting them to their deasserted state.

5.5.2 Die wrapper register (DWR)

The DWR is required in this standard and enables controllability and observability of the logic to and from die terminals for both INTEST and EXTEST modes. There are multiple modes for the DWR. There may also be multiple configurations of the DWR. The rules, recommendations and permissions for the DWR are described in Clause 6. Control of the modes and the configuration of the DWR is done using either IEEE Std 1838 instructions and/or descriptions with a language such as PDL, ICL, and supporting hardware such as SIBs.

5.5.2.1 Specifications

The rule is as follows:

a) Each die shall have a DWR.

NOTE—The DWR is further defined in Clause 6.

5.5.2.2 Description

The DWR can be built using different structures. For instance, it may use only IEEE 1838 DWR cells at the boundary of the die. It may reuse one or more segments of an IEEE 1500 WBR as part of the DWR. It may reuse the IEEE 1149.1 BSR as part of the DWR. It may contain SIBs in the path for local configuration capability.

5.5.3 Flexible parallel port configuration register

Another expansion of the IEEE 1149.1 two-dimensional architecture is the addition of a Flexible Parallel Port and associated Flexible Parallel Port configuration elements that are meant to select and configure the Flexible Parallel Port. This will be discussed and defined further in Clause 7.

5.5.4 ECID register

5.5.4.1 Specifications

The rules are as follows:

a) If the StackID is implemented per recommendation 5.2.1, the ECID register and associated ECIDCODE instruction as described by IEEE Std 1149.1-2013, Clause 13 and 8.15, respectively, shall also be implemented.

b) If a die position register field is included [see 5.5.4.1 recommendation d)], there shall be a field in the ECID called StackID.

c) The StackID value shall not be all 1s.

The recommendation is as follows:

d) There should be a field called StackID, to describe the die position in the stack, accessed via the ECID register and associated ECIDCODE instruction as described by IEEE Std 1149.1-2013 Clause 13 and 8.15, respectively.

NOTE 1—The StackID should be programmed with a unique value to help identify each die in the stack.

NOTE 2—Due consideration should be given to width of this field to accommodate a unique value for any potential position of the die in the stack.

5.5.4.2 Description

The StackID numbering system is user-defined. The correlation between the die StackID value and the actual die should be maintained by the stack producer and/or user. The die provider should make their StackID fields wide enough to handle a unique value within a collection of die in a stack when the number of die will not be known when the individual die is fabricated.

5.5.5 Bypass register

5.5.5.1 Specifications

The rules are as follows:

a) There shall be a 1-bit Bypass register that is operated in accordance with IEEE Std 1149.1.

b) The all-zeros and all-ones Instruction register opcodes shall select the Bypass register between TDI and TDO and leave the device in mission mode.

The recommendation is as follows:

c) Unused Instruction register opcodes should decode to selecting the Bypass register between TDI and TDO and leave the device in mission mode.

5.5.6 Instruction register and instructions

5.5.6.1 Specifications

The rules are as follows:

a) Each PTAP controller and register architecture shall include an instruction register.

b) If a 3DCR can be accessed exclusively from the PTAP instructions, then the instruction Select3DCR shall be used to select it as the active register in the scan path.

NOTE—Configuration elements, including SIB bits or distributed control elements, could select various 3DCR register segments into the scan path. The instruction register may provide multiple instructions to select and operate the 3DCR segments, instead of just the Select3DCR instruction.

c) There shall be at least one instruction opcode called BYPASS that selects the Bypass register.

The recommendation is as follows:

d) If a DWR test mode can be controlled exclusively from the PTAP instruction register, then the instructions SelectDWR-EXTEST, SelectDWR-INTEST, and SelectDWR-TRANSPARENT should be used.

NOTE—Configuration elements, including SIB bits or distributed control elements, could select various DWR elements into the scan path. The instruction register may provide multiple instructions to select and operate the DWR in the various configurations instead of just these three SelectDWR instructions.

5.5.6.2 Description

Elements that need instruction (access) consideration are as follows:

— The optional 3D Configuration Register used to enable, configure, and render reset-persistent STAPs; and is mandated if STAPs are present

— The Die wrapper register (DWR)

— The optional FPP configuration elements used to enable and configure FPP connections

— The single bit Bypass Register

For a die with STAPs, there is a requirement to have a mechanism to select, activate, and place into the active scan path the configuration elements of the (virtual) 3DCR that enables and configures access to connected dies in the stack.

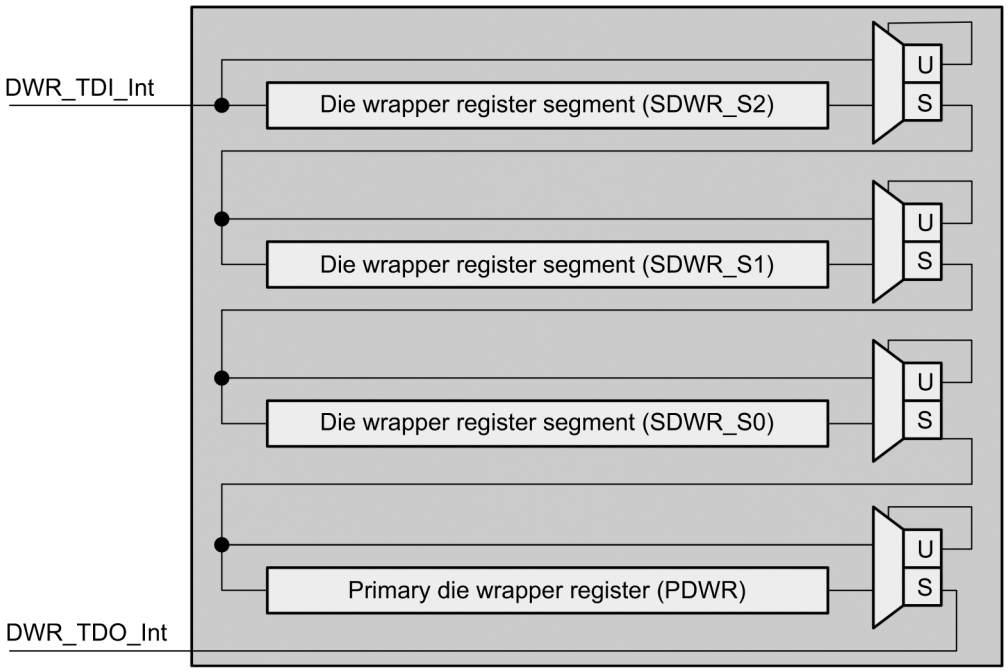

Figure 13—Primary and secondary DWR segment concatenation

This standard provides access to DfT structures inside a die stack, but does not specifically mandate test functionality (for the most part). How this access is used depends on the desired application. Figure 13 shows an example of Primary and Secondary Die Wrapper Register segments that may be selected as a whole (viewed as one contiguous register), perhaps using existing IEEE Std 1149.1 instructions such as EXTEST, INTEST, SAMPLE, and PRELOAD. Segments may also be configured into and out of the active scan path using in-line SIB elements or even elements not directly accessible in the active scan path. As an example, using the IEEE Std 1149.1 PRELOAD instruction, accessing the DWR or setting the DWR mode could be set up prior to taking these registers off-line for test purposes.

A more complex DWR may be composed of boundary-scan cells, wrapper elements from IEEE 1500 compliant cores, and individual IEEE 1838 compliant wrapper cells. Control of this complex DWR might come from the instruction loaded into the Instruction register, SIBs, scannable TDR bits, WIRs, or other serially accessible elements. These elements may, in turn, not be available in the fully composed DWR. So, accessing the DWR might take several scan operations and various configuration settings before the proper content and test mode is established. But, in some cases, a single instruction such as SelectDWR-EXTEST, SelectDWR-INTEST, and SelectDWR-TRANSPARENT could be used.

Flexible Parallel Port configuration could also involve multiple scan operations to completely configure the optional channels for broadband access.

The PTAP controller Bypass Register is available to be enabled within the active scan path to quickly bypass the die while accessing other die in the stack using the IEEE Std 1149.1 BYPASS instruction.

5.5.7 Device identification register

5.5.7.1 Specifications

The rule is as follows:

a) The test access controller register architecture shall contain a device identification register that shall operate in accordance with IEEE Std 1149.1-2013 as described in its Clause 12 and 8.13.

5.6 Configuration elements

The 3DCR, DWR, and FPP configuration elements specify the functionality and configuration of the STAP DWR and FPP lanes.

5.6.1 Specifications

The rules are as follows:

a) Configuring elements shall be accessible by the TAP using one or more instruction encodings.

b) The configuration elements shall be designed in accordance with the rules in 7.4.2 of IEEE Std 1149.1 describing capture-update test data registers (TDR).

NOTE—An example TDR is illustrated within Figure 9-6 of IEEE Std 1149.1-2013.

c) The configuration element update registers shall be set to their reset value when TRSTN input signal of the PTAP controller is asserted.

d) Each configuration element's reset value shall be documented in a human-readable and machine-readable format.

e) If a configuration element has a Config-Hold function, and the Config-Hold bit is asserted, all configuration elements associated with that Config-Hold function shall not change when the PTAP FSM asserts Reset*.

f) There shall be a description in a machine-readable language of the control of the modes and configuration of the STAP, DWR, and FPP, if a single instruction is not sufficient.

NOTE—This standard has not yet defined a specific language suggestion for this purpose.

g) All cells which control register segmentation multiplexers (SIBs) shall include a shift, capture, and update capability; the update shall be with delay such that the output of the inline segment select cell does not change until at least one half TCK cycle after UPDATE-DR.

The recommendation is as follows:

h) SIB control registers should be placed after their corresponding segment selection multiplexer.

The permissions are as follows:

i) There may be a single instruction to configure the DWR.

j) There may be a single instruction to configure the FPP.

k) There may be a single instruction to configure the STAP using the 3DCR Register.

5.6.2 Description

The STAPs, DWR segments, and FPP lanes can have numerous settings to adjust their resources as appropriate for the tests being applied. These settings are controlled by test data registers accessible from the PTAP. These settings should be able to hold their state for as long as the test access is required, but will definitely be reset when TRSTN is asserted.

There should be a way to control the configuration and the modes of the STAPs, DWR, and FPP. The STAP, DWR, and FPP are not required to have a configuration register, though they are allowed to do so. It can be configured through distributed control such as SIBs. If distributed control is used, it must use the distributed control register access control logic (Figure 11) and there should be a description of how to enable the distributed control for every possible mode and every possible configuration. This should be described in a machine-readable language such as ICL and PDL.

The configuration elements for the STAPs, DWR, and FPP can be scattered throughout the serial network. SIB selection mechanisms might be placed to gate access to these configuring elements. When a SIB is used to gate the access, the select signal from the SIB will be used to block segment element upsets by turning off the control signals to that segment like Capture, Shift, and Update. This select signal will change with the Update timing. As such, the segment's Update signal might be gated off too soon. A delay in the selection signal will solve this timing issue as described in rule g).

6. Die wrapper register

6.1 Register design

This standard mandates a serial test interface and supports an optional parallel test interface for accessing an IEEE 1838 compliant wrapped die. The serial mode of the Die Wrapper Register (DWR) incorporates a single chain composed of all DWR cells accessed by the IEEE 1838 serial test interface. The parallel mode of the DWR may make use of one or more IEEE Std 1838 wrapper chain segments, accessed by the FPP. Figure 14 shows the DWR in an IEEE Std 1838 serial test interface architecture.

The DWR is constructed of DWR cells, each of which may include four data terminals: cell functional input (CFI), cell functional output (CFO), cell test input (CTI), and cell test output (CTO).

Figure 14 shows the following:

— Die functional input terminals (DFIs) are connected to the CFI terminals of DWR cells provided for die functional input logic.

— CFO terminals of DWR cells provided for functional die inputs are connected to the corresponding functional logic.

— Functional die output logic is connected to the CFI terminals of corresponding DWR cells provided for that die functional output logic.

— The CFO terminals of DWR cells provided for die functional output logic are connected to corresponding die functional output terminals (DFOs).

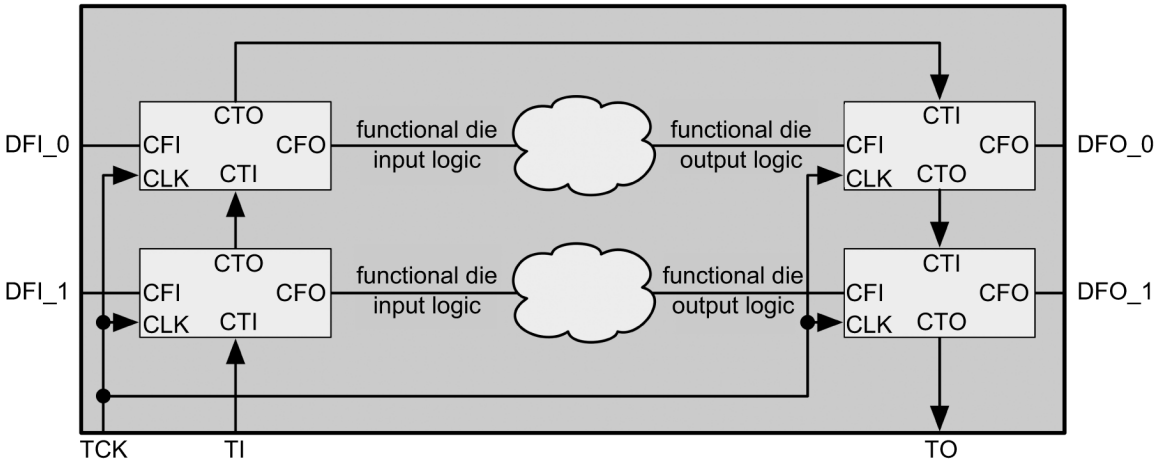

Figure 14—Example of DWR in serial test interface configuration to be accessed from IEEE 1149.1 Test Access Port

DFI and DFO are generic names for wrapped die terminals that are to be connected within a die stack. It is often the case that identical DWR cell types are used for wrapping both die input terminals and die output terminals. These DWR cells are connected differently and may receive different control signals (not shown). The DWR shift path is formed by connecting the CTO terminal of one DWR cell to the CTI terminal of the next DWR cell in a daisy chain fashion. TI and TO are defined to be the scan input and scan output terminals of the DWR as a whole, as shown in Figure 14.

In Figure 15 an implementation of an IEEE Std 1838 DWR interface using two parallel scan chains is depicted. Although this figure shows all input terminals (DFI) placed on one scan chain and all output terminals (DFO) on a second scan chain, the IEEE Std 1838 parallel test interface is not limited to this configuration. In addition, although TCK is shown in both Figure 14 and Figure 15. IEEE Std 1838 does not mandate the use of TCK for clocking the wrapper cells. A different clock, i.e., a system clock, may be used although the TCK is required for serial wrapper access at the die wrapper level.

In Figure 14 and Figure 15 the DFI_k terminals are DFI terminals, while the DFO_k terminals are DFO terminals.

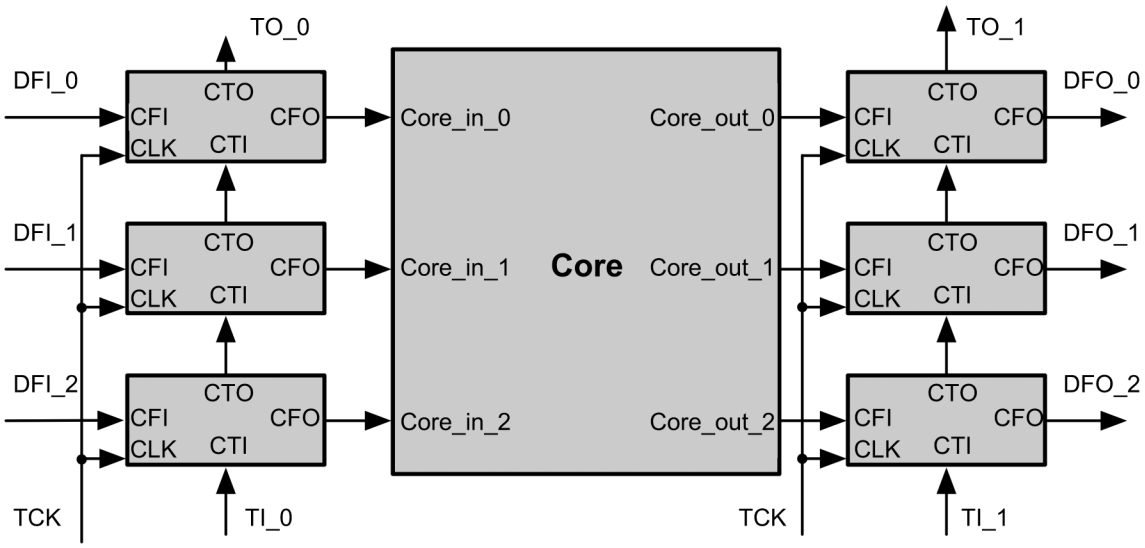

Figure 15—Example of DWR in parallel test interface configuration to be accessed from the IEEE 1838 flexible parallel port DWR structure and operation

6.1.1 Specifications

The rules are as follows:

a) Every digital signal connected to a die terminal shall be provisioned with a fully-provisioned DWR cell, except for signals that cause data to be loaded into a sequential element or dedicated test signals specified in this standard.

NOTE—Examples of signals that cause data to be loaded into a sequential element include the clock of a flip-flop, the gate of a latch, and the asynchronous set or reset of either a flip-flop or a latch.

b) The DWR shall have at least one configuration in response to the state of the configuration registers, allowing serial access to and from all DWR cells between TI and TO and utilizing TCK as the clock.

c) Every wrapped terminal shall be associated with at least one DWR cell, except as exempted under permission 6.1.1 u).

d) While a DWR or DWR segment is not selected, its UpdateDR signal shall not cause the output of the DWR cell with the update capability to change state.

e) A selected DWR segment shall place all of its DWR cells between the TI and TO of the DWR during Shift mode.

f) All DWR cells in a selected DWR segment shall support the shift event.

g) All DWR cells connected to an active functional interface shall be included in a selected DWR segment during the associated test mode.

h) All DWR cells in selectable DWR segment combinations shall support a common shift frequency.

i) For DWR configurations mandated in this standard in which the DWR segment is accessed through the PTAP, the DWR segment clock shall be TCK during shift.

NOTE—This rule verifies that PTAP-only access to DWR segments can shift using only TCK. Other configurations defined by the user could use other clock sources.

j) The configuring elements of the DWR shall be accessible by the PTAP using one or more instruction encodings.

k) The DWR configuration elements shall be designed in accordance with the rules in 7.4.2 of IEEE Std 1149.1 describing capture-update test data registers (TDR).

NOTE—An example TDR is illustrastrated within Figure 9-6 of IEEE Std 1149.1-2013.

l) DWR configuration element update registers shall be set to their reset value when TRSTN signal is asserted.

The recommendations are as follows:

m) Every digital input signal connected to a die terminal exempted from wrapper cell insertion per rule 6.1.1 a) should be provided with an alternate access mechanism for observability.

n) Every digital output signal connected to a die terminal exempted from wrapper cell insertion per rule 6.1.1 a) should be provided with an alternate access mechanism for controllability.

NOTE—Care should be taken when adding DWR cells to test outputs so they do not interfere with the performance of an active test.

o) Separately selectable DWR segments should be provided for the primary and each secondary functional interface.

p) For DWR configurations mandated in this standard in which the DWR segment is accessed through the TAP, the DWR segment clock should be TCK during the update event.

q) Any digital output terminal that has a possibility of connecting to an analog input terminal on another die should be provisioned with a fully-provisioned DWR cell, and a three-state driver between the DWR cell and the output terminal that is initially disabled in test mode and controlled by a fully-provisioned cell with update capability.

r) Differential ac-coupled terminals should follow the boundary scan cell rules, recommendations and permissions from Clause 6 of IEEE Std 1149.6.

The permissions are as follows:

s) Primary or secondary port terminals connected to optional dedicated flexible parallel port lanes may be provisioned with a DWR cell that does not interfere with the operation of the flexible parallel port.

NOTE—If so provisioned, any test mode that utilizes the FPP should not include wrapper cells in these DWR configurations.

t) Any die terminal may be provisioned with a DWR cell, provided doing so does not alter the behavior of standard instructions.

u) In a case in which a die input terminal is used solely as the source of data for an output terminal, a single DWR cell may be shared between the input and the output terminals.

v) A die terminal may be provisioned with more than one DWR cell.

w) There may be combinational logic between the die terminal and the DWR cell.

x) The mode on a deselected DWR segment may be set to a mission or test mode.

y) An IEEE 1149.1 boundary scan cell may be shared as a DWR cell and used in the DWR scan chain(s).

z) An IEEE 1500 wrapper cell may be shared as a DWR cell and used in the DWR scan chain(s) if it is compliant to all IEEE Std 1838 DWR cell rules.

aa) An auxiliary clock (AUXCK) may be used for clock control of the DWR cells.

6.1.2 Description

Rule 6.1.1 a), and recommendations p) and q) define which terminals shall have a wrapper cell and the minimum requirement of those wrapper cells. Considering the permission w), these wrapper cells may actually support testability of multiple die terminals at the same time.

A shift path in a DWR cell is composed of one or more storage elements serially connected between CTI and CTO. The DWR should have a single, uniform shift path between its TI and TO to enable serial access. DWR structure rule 6.1.1 b) supports this. Likewise, for the serial mode of the IEEE Std 1838 wrapper, a shift path is composed of the storage elements of all the cells in the DWR concatenated into a single serial path. While any DWR is selected, there should be at least one DWR cell between TI and TO.

Rule 6.1.1 d) pertains to small set of die outputs per permission 6.2.1 g). For example, die outputs equipped with output enable control may use DWR with update functionality. Likewise, signals known to be controlling important signals connected to other dies such as power control, clock gate control, reset, etc. may want to use the protection of update functionality. If the DWR segment in which the wrapper cell with update capability resides is not selected then the output of that wrapper cell cannot change, even if an Update DR occurs.

If the FPP has wrapper cells, the scan chain(s) should be separate from any other wrapper chains or internal scan chains. This allows separate modes for internal or external tests that use the FPP in its mission mode without enabling the FPP wrapper. There should also be a mode to enable access to the FPP wrapper, to allow for the interconnect test of the FPP.

If a primary or secondary functional interface is active during a test, there should be a selected DWR segment which attaches to all terminals, per rule 6.1.1 g), and is selected between TI and TO. The selected DWR segment should be able to shift values through all DWR cells whether or not they are attached to die terminals. In addition, the DWR segment that contains the DWR cells connected to the active primary or secondary functional interface should be selected during that test.

Though rule 6.1.1 a) exempts the requirement of a fully-provisioned DWR cell from some types of signals, it is recommended that the input signals have the capability to be observed. For instance, an asynchronous reset is exempt from this rule as direct controllability may need to be enabled. However, there may still be a desire to observe whether the signal has been properly connected from another die. For this reason, it is recommended that observability be added to the terminal (i.e. partially-provisioned DWR cell). The output signals that are exempt from Rule 6.1.1 a) should provide controllability to the neighboring die to prevent putting it into an unknown state or prevent it from being damaged.

There is a requirement that one configuration of the DWR is a single scan chain per rule 6.1.1 b). However, it is recommended [in 6.1.1 o)] that each functional interface have its own DWR segment so that testing of each interface can be accomplished while using a shorter DWR scan chain. One area where this may be used is mid-bond stacking where a second die is bonded to the first die in a package. Before placing a third die on top of the second die, it may be desirable to test that the second die is properly connected. This can be done by just enabling the primary DWR segment of the second die and the secondary segment(s) of the first die, if these configuration options are available. This creates much shorter DWR scan chains than if the entire DWR is a single scan chain. However, EXTEST is probably a very short test and it is up to the die creator to decide whether this is a good configuration option to add to the DWR.

It is permissible to use a clock other than TCK to control the DWR cell during test when the DWR is selected. This clock is called AUXCK in this standard. For instance, functional registers may be reused as DWR cells. These are called shared wrapper cells and the functional clock would be connected to these DWR cells. However, at least one configuration of the DWR should be implemented where TCK is used to clock all DWR cells [see rule 6.1.1 b)]. This indicates that if AUXCK is used to clock a DWR cell in a selected DWR segment during a test, there should be a way to switch to TCK to clock that same DWR cell in a separate DWR configuration or mode. Typically, on-chip clock controllers provide a multiplexing mechanism that

chooses between a test clock and a functional clock, and this capability can be used to supply the required clock in different configurations. All timing interactions between TCK and AUXCK should be considered if permission if 6.1.1 aa) is employed.

If the DWR segment is being accessed through the PTAP, then the segment clock will be TCK. It is generally assumed and recommended that any update element in a DWR cell is clocked by TCK. However, there may be reasons to have the update element clocked by AUXCK, so this is allowed.

If the die under test contains a boundary scan register, the boundary scan register can be used as part of the DWR as it follows the rules of IEEE 1838 DWR cell.

If the die under test incorporates an IEEE 1500 WBR cell and it is connected to a die terminal, then it should follow all IEEE 1838 DWR cell rules. An IEEE 1500 cell that allows that Standard's Transfer operation is not compliant to IEEE Std 1838, so cannot be utilized on a die terminal that requires a fully-provisioned DWR cell or utilizes a partially-provisioned DWR cell. However, if the IEEE 1500 WBR cell with transfer is in the shift path of the DWR and not connected to a die terminal, then it can be included in the DWR scan chain to be used during shift only. If the IEEE 1500 WBR cell with transfer is connected to a die terminal, a dedicated DWR cell should be used instead of the WBR cell to communicate with that terminal during test.

6.2 DWR cell structure and operation

The DWR cell operation relies on the Shift, Capture, Update, and Apply events more fully defined in 6.3.1.

6.2.1 Specifications

The rules are as follows:

a) Every DWR cell shall have at least one storage element connected between its CTI and CTO terminals.

b) Every fully-provisioned DWR cell shall have at least one storage element connected between its CFI and CFO terminals.

c) Every fully-provisioned DWR cell shall have a storage element provisioned for the purpose of servicing the Capture event and this element shall be in the shift path.

d) Every fully-provisioned DWR cell shall have a storage element provisioned for the purpose of servicing the Apply event.

e) A partially-provisioned DWR cell shall support the Shift event and either the Capture event to observe the CFI/CFO terminal or the Apply event to control the CFO terminal.

The recommendation is as follows:

f) Partially-provisioned DWR cells should be implemented on die terminals exempted from being wrapped per rule 6.1.1 a).

The permission is as follows:

g) Any DWR cell may have a storage element provisioned for servicing the Update event.

6.2.2 Description

DWR cell structure rules 6.2.1 a) and 6.2.1 b) provide for a minimal implementation consisting of a means to select test data or functional data as inputs to a shared storage element.

DWR cell structure rule 6.2.1 a) allows for multiple storage elements in the shift path in order to support test methodologies requiring the application of sequential patterns (e.g., path-delay, transition delay, piecewise functional).

DWR cell structure rule 6.2.1 c) requires that captured data enter the shift path of a cell at a storage element.

Unidirectional IEEE Std 1149.1 boundary-scan cells are usable as IEEE Std 1838 DWR cells. DWR cell structure permission 6.2.1 g) supports this.

The DWR cell is fully self-sufficient for the processing of a test so that, after shifting in test data and before shifting out test data, all data for the test are sourced from and/or sampled into one or more storage elements in the single cell, i.e., the terminals' test needs are met solely by their respective DWR cells.

DWR cells on die terminals that need not be wrapped per rule 6.2.1 a) may have reduced functionality cells as described in recommendation 6.2.1 f). For instance, one may wish to have an observe-only cell on clocks as defined in rule 6.2.1 a) or other types of observe cells supporting the Shift or the Capture event in outward facing (OF) mode. If the die terminal provided with a reduced-functionality cell is a clock terminal and gets connected to the clock input of the reduced-functionality cell, then this clock would be considered an AUXCK and would have to follow AUXCK requirements and allow for proper operation of the DWR.

6.3 DWR operation events

An event is an uninterrupted, predefined sequence of one or more steps. Within these steps there may be a particular instant that characterizes the event, when the nominal action of the event occurs. This is referred to as the characteristic instant. Predefined events are Shift, Update, Capture, and Apply. Some events may overlap with others as indicated in permission 6.3.1 e). Events apply to the DWR as a whole.

6.3.1 Specification

While the wrapper is in OF mode, the Apply event causes test data to be applied from output cells onto DWR functional outputs. The test data are the data stored in the shift path storage element unless the Update event is supported, in which case the test data shall be the data in the DWR cell storage element loaded during the Update event.

The rules are as follows:

a) The Shift mode shall not occur simultaneously with either Capture or Update events.

b) While the DWR is selected, the DWR wrapper cell shall only change state in the presence of an event that it supports, such as Shift, Capture or Update.

c) In the absence of an active edge from the clock or a reset state, DWR storage elements in a selected segment shall not change state.

The recommendation is as follows:

d) While accessing TDRs, TCK should not cause a deselected TDR storage element to change state.

The permission is as follows:

e) Except where excluded by rule 6.3.1 a), events may be discrete, simultaneous, or overlapping.

6.3.2 Description

During Shift mode, the shift event is applied to all DWR cells so that data is able to be loaded and unloaded from any active scan chain. During Shift mode, no other event, such as capture, is allowed to occur on any DWR cell in an active scan chain.

DWR events are enabled by the active edge of the DWR clock signal(s) (rising-edge for Capture, Shift, and falling-edge for Update) in conjunction with the assertion of a DWR control signal (CaptureDR, ShiftDR, or UpdateDR) as shown in Figure 6 and Figure 7. The DWR clock pulses or edges applied in the absence of active DWR control signals do not result in any event. This requirement comes from the need to enforce interoperability between various types of wrapper cells that are operated simultaneously. A generic wrapper cell services the shift, capture, and/or update events when, respectively, the IEEE Std 1838 TAP Controller output signals ShiftDR, CaptureDR, or UpdateDR are asserted, and holds state when none of these signals are asserted. However, IEEE Std 1838 also allows wrapper cells which do not implement an update stage and hence do not service the update event. Such a wrapper cell should be able to hold state whenever UpdateDR is asserted. IEEE Std 1838 also allows wrapper cells with a single control signal that if asserted, allows the DWR cell to shift and if deasserted, allows the DWR cell to capture. This situation is common for shared wrapper cells with only a ShiftEnable signal connected to the multiplexer of a DWR storage element. Such a wrapper cell should be able to hold state whenever none of the control signals ShiftDR, CaptureDR, and UpdateDR is asserted or when UpdateDR is asserted while the DWR is selected.

DWR clock signal(s) may be held in a high or low state for arbitrarily long intervals without loss of DWR state.

The TCK signal should not cause TDR cells of unselected TDRs to change state. This will help ensure a known or safe state to the external or internal logic, if the TDR cell does not have a static output. For example, an unselected TDR holding the test mode of a core should not be disturbed while other TDRs in the die are accessed.

While a scan chain is in Shift mode, all active DWR segments should be shifting all internal DWR cells; in other words, every DWR cell in the aforementioned segments should be performing the Shift Event. However, during modes other than Shift mode (e.g. Capture Mode), DWR cells may support different events. For instance, during a capture cycle while in INTEST, it may be preferable to leave all DWR cells that are attached to input terminals in the Shift Event. Since they are attached to inputs, they can only capture unknowns from the input terminals. Unknowns can grow the pattern size and prevent delay test on the cone of logic connected to the input terminals (after the DWR cell). Controlling the DWR in this manner may help to facilitate delay testing in IF mode, while isolating these inputs from the outside function logic paths. In this scenario, the DWR cells connected to input terminals would apply the shift event, while all output scan cells would apply the capture event. This is allowable under permission 6.3.1 e).

6.4 DWR operation modes

A mode is a static condition or configuration of the DWR that exists in response to the state of the test mode control signals. This subclause describes the mission mode, the IF mode, the OF mode, and the Safe Operation (SO) mode. The mission mode is the mode in which the DWR does not interfere with the functional operation of the die. The IF mode is a test mode where die inputs are controlled by the DWR and die outputs are observed by the DWR. The OF mode is a test mode where die outputs are controlled by the DWR and die inputs are observed by the DWR. The goal of SO mode is to ensure that entering or configuring a particular test mode does not affect the die or inter-die circuitry in a hazardous manner.

6.4.1 Specifications

The rules are as follows:

a) While in mission mode, the DWR shall not interfere with the operation of the die or with the flow of signals to and from the die.

NOTE—DWR state elements may be shared with functional registers, or used to observe system states during functional operation, so they may be clocked if necessary.

b) While in IF mode, DWR cells provided for die inputs shall respond to Shift, Apply, and, if provisioned for them [permission 6.2.1 g)], the Update events.

c) While in IF mode, DWR cells provided for die outputs shall respond to the Capture and Shift events.

d) While in OF mode, DWR cells provided for die outputs shall respond to Shift, Apply, and, if provisioned for them [permission 6.2.1 g)], the Update events.

e) While in OF mode, DWR cells provided for die inputs shall respond to Capture and Shift.

f) For SO, the control signals of three-state drivers for output and/or bidirectional terminals of a die shall be controlled by a fully-provisioned DWR cell.

g) For SO, during the DWR shift event, three-state drivers for output and/or bidirectional terminals of a die shall be controlled such that a persistent safe drive state is maintained on the associated die terminals.

h) For SO, if the update storage element of a DWR cell that controls a three-state driver is reset in the Test-Logic-Reset TAP state, it shall be reset to the state that will cause the connected three-state drive to be disabled.

The recommendations are as follows:

i) While in IF mode, DWR cells provided for die outputs should present safe data at their CFO terminals (i.e., external safe state).

j) While in OF mode, DWR cells provided for die inputs should present safe data at their CFO terminals (i.e., internal safe state).

k) The design of DWR output terminals should allow the user to bring them to a safe state upon entering test mode.

l) If changing logic levels of the output terminal that is connected to an input terminal of an adjacent die can damage or cause inappropriate activity in that adjacent die, the DWR cell of that output terminal should have an update storage element or a safe gate to control the output data during test.

The permissions are as follows:

m) IF mode and OF mode may be operative at the same time.

n) Whereas the four modes defined in this subclause (i.e., mission, IF, OF, and safe) are applied homogeneously across the entire DWR, other modes may be defined in which the DWR cells respond on an individual basis.

o) While in mission mode, the DWR may respond to the Capture or Shift events, provided that such response does not conflict with rule 6.4.1 a).

p) A DWR control cell may enable and disable one or more three-state drivers on die terminals.

6.4.2 Description

While configuring or entering a test mode, hazardous conditions (e.g. contention) on all ports should be avoided. Whenever there is a three-state driver, the control to that driver should be controllable so that during test mode a hazardous state does not occur on logic external to the die. In addition, there should be capability

to observe logic connected to the control port of the three-state driver. For these reasons, a fully-provisioned wrapper cell should be attached to the control port of a three-state driver.

There shall be a persistent drive state on the control of a three-state driver during Shift mode to help ensure that the output of the three-state driver remains in a safe state no matter what value is shifted into the terminal's wrapper cell. Disabling the three-state driver is one way to help ensure a safe state. Two options for delivering a persistent drive state on the control of a three-state driver are 1) a safe gate or 2) an update capability to the wrapper cell.

If an output terminal of a die that connects to an input terminal of an adjacent die can damage or cause inappropriate activity to the adjacent die, the terminal should output a safe value during the DWR shift event for both INTEST and EXTEST mode. In addition, during INTEST mode, a safe value should be output during the capture event. An example is an output terminal that connects to a bidirectional signal; if both dies enable the drive capability, there will be contention, which can damage the circuit. A second example is if the output terminal of one die can wake or put to sleep an adjacent die, there should be control to put the adjacent die in a stable mode during shift and capture events. In addition, control may need to be considered on the input terminal of the adjacent die for this type of signal.

6.5 Parallel access to the DWR

In addition to its mandatory serial wrapper access mechanism using the IEEE 1838 PTAP, IEEE Std 1838 provides for an optional parallel wrapper access mechanism called the Flexible Parallel Port (FPP).

6.5.1 Specifications

The rule is as follows:

a) Each parallel configuration of DWR segments and the associated configuration registers (see 5.6) shall be setup and enabled by means of the die-level TAP.

The recommendation is as follows:

b) A selected DWR segment should have a scan chain length that is the same or close to the length of other scan chains that are selected simultaneously.

The permissions are as follows:

c) The test input (TI) and test output (TO) of each DWR segment may be connected to one of the following:
   — to the output or input, respectively, of another DWR segment,
   — to the output or input, respectively, of an FPP lane,
   — to a die-internal scan chain output or input, respectively.

d) There may be more than one parallel configuration of the DWR.

e) In a parallel configuration, the DWR may use serial control signals.

6.5.2 Description

The DWR should have a serial configuration, but may have multiple DWR segments for various reasons including to support any parallel configurations desired. Parallel configurations of the DWR allow the flexibility to access multiple short DWR segments simultaneously. After an appropriate IEEE 1838 instruction,

which enables access to the segmentation control infrastructure, is input to the TAP controller, a parallel configuration of the DWR can be enabled in one of multiple ways (e.g., configuration register, seg-select bit).

The flexibility in connectivity allows DWR segments that are controlled and/or observed externally to the die under test to be connected via their test inputs and outputs to FPP or to internal scan chains. FPP data signals that are distinct from TAP signals are used for the DWR segments that require external control/observe during the application of a parallel test. TDI and TDO pins may be re-used for access to a DWR segment during parallel DWR configurations at the same time as the FPP. However, like other FPP resources, this DWR segment might not be accessible when the package is soldered to the PCB.

6.6 DWR cell naming

IEEE Std 1838 mandates a naming convention for DWR cells.

6.6.1 Specifications

The rules are as follows:

a) All IEEE Std 1838 DWR cell names shall match the following regular expression: /[DW]C(_S[DF]\d+)(_C([IO]|\d+)|N)?(_U)?(_O)?(_G[01Z]?)?/.

b) The first field shall be “DC” unless an existing IEEE Std 1500 WBR cell in the die is reused in the DWR as allowed by permission 6.6.1 h), in which case the first field shall be “WC”.

c) The second field shall be one of the following:

   1) “_SD” followed by a decimal integer indicating the number of shift path storage elements for a shift path with dedicated storage element(s), or

   2) “_SF” followed by a decimal integer indicating the number of shift path storage elements for a shift path with functional storage element(s).

d) The third field shall be one of the following:

   1) “_CI” (if the data is being captured from CFI) followed by a decimal integer indicating the location of the capture site counting from CTI, or

   2) “_CO” (if the data is being captured from CFO) followed by a decimal integer indicating the location of the capture site counting from CTI, or

   3) “_CN” to indicate that the wrapper cell that does not perform the capture event.

e) The fourth field shall be “_U” if and only if the wrapper cell has an update storage element.

   NOTE—The field is not allowed in the absence of an updated storage element.

f) The fifth field shall be “_O” if and only if the wrapper cell has an observe-only storage element.

   NOTE—The field is not allowed in the absence of an observe-only storage element.

g) The sixth field shall be “_G” followed by the safe value (“0”, “1”, or “Z”), if and only if the wrapper cell has a guarded safe value.

The permissions are as follows:

h) All IEEE Std 1500 WBR cells with the following regular expression may be used

i) /WC(_S[DF]\d+)(_C([IO]|[IO])|N)?(_U)?(_O)?(_G[01]?)?/ when IEEE 1500 WBR cells are utilized.

j) IEEE 1149.1 BC cell naming may be used for IEEE Std 1838 DWR cells when IEEE 1149.1 cells are utilized.

k) IEEE 1149.6 cell naming may be used for IEEE 1838 DWR cells when IEEE 1149.6 cells are utilized.

6.6.2 Description

DWR cell structures are defined in a regular expression format to ease software tool automation. The regular expression defined in 6.6.1 a) allows for the identification of any IEEE 1838 DWR cell.

The naming of the various types of DWR cells shall be done in a descriptive, parsable method. Each cell type name shall begin with DC (DWR cell). This prefix is followed by a sequence of characters that describe the capabilities and structure of the cell. The information will indicate whether a particular storage element is shared or dedicated to wrapper operation, how many shift path storage elements exist, the existence and type of the optional update cell, and the existence of safe operation.

To support this, several fields will exist in the name in a specified order. Each field begins with an underscore.

— The second field is mandatory and describes the nature and number of shift path storage elements. The first two characters of this field shall be "_S" (shift). The third character is either "D" (dedicated) or "F" (shared functional), followed by an integer indicating the number of shift path storage elements. This second field has the following regular expression format: /_S[DF]\d+/.

— The third field is mandatory and describes the data capture source. The first two characters of this field shall be "_C" (capture). The third character of this field shall specify the origin of captured data: "I (CFI), or "O" (CFO), followed by an integer indicating location of the capture site counting from CTI. In cases where the wrapper cell being described does not perform the capture function, both the third character and the following integer in this field should be replaced by "N" (none) to indicate that there is no origin for captured data and that the capture site does not exist. The regular expression matching this second field is /(_C([IO]\d+)|N)?/.

— The fourth field describes the nature of the optional update element. It is composed of two characters which are "_U" (update). Its regular expression format is /(_U)?/. The absence of this field indicates the absence of an update storage element.

— The fifth field is optional indicating the presence or absence of an observe-only element. It is composed of two characters which are "_O" (observe). Its regular expression format is /(_O)?/. The absence of this field indicates the absence of an observe only storage element.

— The sixth field is optional indicating the presence or absence of safe data support in nonhazardous mode. It is composed of two characters "_G" (guarded data) followed by a 0, 1, or Z indicating the static value. Its regular expression format is /(_G[01Z])?/.

6.7 DWR cell examples

Table 1 describes IEEE 1838 DWR cell example names, and a gallery of bubble diagrams depicting each example follows the table. The bubble diagrams used in these examples are defined in Annex A. It is understood that the means of data path selection shown in these bubble diagram figures is to be configured by the content of the WIR.

Table 1—DWR cell example list

Cell description	Name	Figure number
One storage element shared with functional operation	DC_SF1_CI1	Figure 16
One storage element shared with functional operation with a safe gate that outputs the 1	DC_SF1_CI1_G1	Figure 17
One dedicated shift path storage element capturing from CFO.	DC_SD1_CO1	Figure 18
Two dedicated shift path storage elements, capturing from CFI into the shift storage element closest to CTO (the 2nd storage element)	DC_SD2_CI2	Figure 19
One dedicated shift path storage element with an update storage element, capturing from CFI	DC_SD1_CI1_U	Figure 20
One dedicated shift path storage element with an update storage element and a safe gate which outputs a 0, capturing from CFI	DC_SD1_CI1_U_G0	Figure 21
A reduced-functionality cell with one dedicated shift path storage element performing an observe-only function, capturing from CFI	DC_SD1_CI1_O	Figure 22

IEEE Std 1838: /DC(_S[DF]\d+)(_C([IO]([IO]|\d+))|N)?(_U)?(_O)?(_G[01Z])?/

IEEE Std 1500: /((WC)|(WH))((_S[DF]\d+)|(_CI?))(_C([IOB][IOU])|N)?(_U[DF])?(_O)?(_G[01]?)?/

The DC_SF1_CI1 cell types have the least circuitry. As this cell services both mission mode and test mode, the two operations are likely mutually exclusive. Also, the CFO terminal toggles as data are shifted from CTI to CTO and when a capture occurs. DC_SF1_CI1_G1 may drive a safe value in addition to supporting the functionality described for DC_SF1_CI1 (see Figure 16).

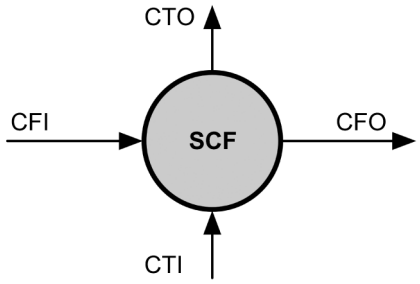

Figure 16—DC_SF1_CI1 DWR cell

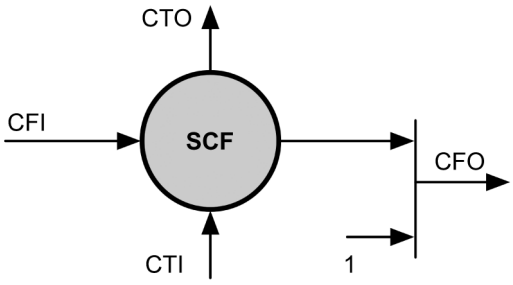

Figure 17—DC_SF1_CI1_G1 DWR cell

DC_SD1_CO1 has a dedicated shift path, and its CFO may toggle during Shift or Capture operations. However, compared to the DC_SD1_CI1 cell, the CFI to CFO connection has superior testability since more of the functional path is tested (see Figure 18).

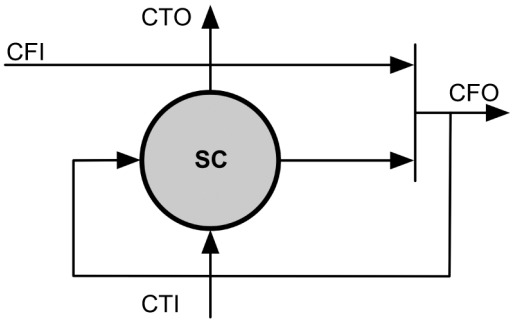

Figure 18—DC_SD1_CO1 DWR cell

The DC_SD2_CI2 cell has the capability to apply a sequential pattern for delay testing applications (see Figure 19).

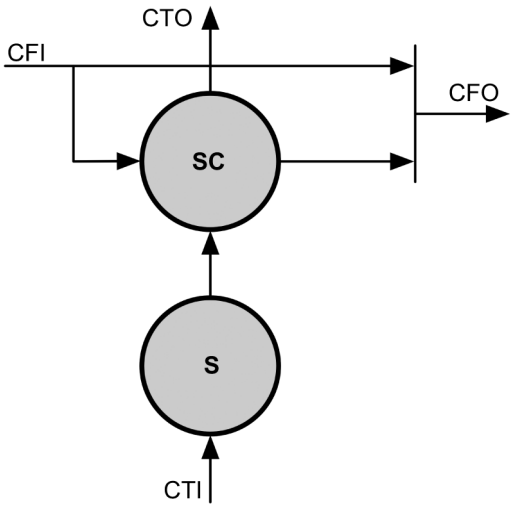

Figure 19—DC_SD2_CI2 DWR cell

DC_SD1_CI1_U has a dedicated shift path and an update storage element (see Figure 20).

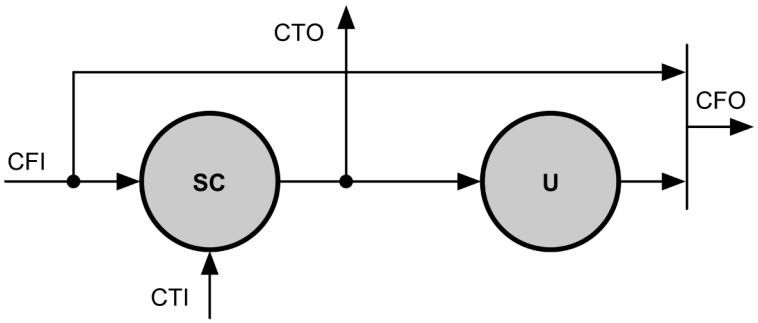

Figure 20—DC_SD1_CI1_U DWR cell

DC_SD1_CI1_U_G0 shows an example where in addition to the update cell, there is also a gated safe value as shown in Figure 21.

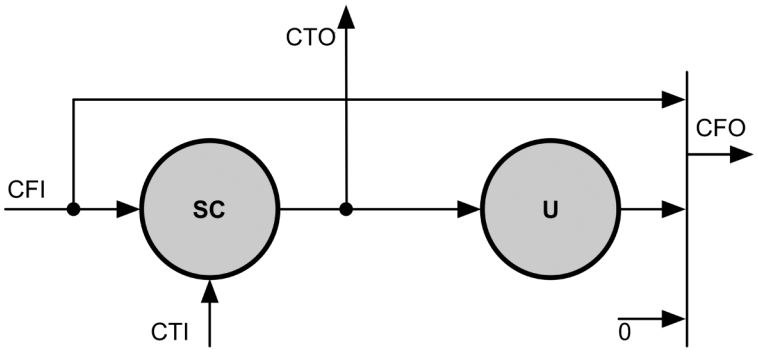

Figure 21—DC_SD1_CI1_U_G0 DWR cell

DC_SD1_CI1_O is an observe-only harnessing cell. This cell is to be used only in accordance with rule 6.2.1 d) and recommendation 6.2.1 f) [see Figure 22].

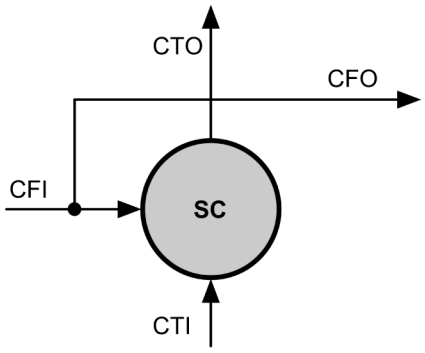

Figure 22—DC_SD1_CI1_O DWR cell

6.8 Wrapper states

The Test-Logic-Reset state can be used to force the DWR logic into a state that enables functional operation of the die. The DWR is forced unconditionally to a Wrapper Disabled state when the Test-Logic-Reset state is entered. In Wrapper Disabled state, the DWR is inactive and functional operation of the die logic can continue unhindered. Test operations are performed while the DWR is in Wrapper Enabled state.

6.8.1 Specifications

The rules are as follows:

a) While in the Test-Logic-Reset state, the DWR shall be in Wrapper Disabled state.

b) While in the Test-Logic-Reset state, all DWR segments shall be deselected.

6.8.2 Description

Independent of the current state of any wrapper cell, they should all enter the Wrapper Disabled state when the PTAP controller enters the Test-Logic-Reset state. The DWR will remain in the Wrapper Disabled state while the PTAP controller is in the Test-Logic-Reset state. In the Wrapper Disabled state, the wrapper is forced to its inactive or mission mode state. This is achieved by ensuring the DWR mode is set to mission mode by setting all configuration elements appropriately. This is the default state of the DWR configuration elements.

7. Flexible parallel port

The flexible parallel port (FPP) provides multi-bit (parallel) test access to the die and stack under test.

The FPP can be used to establish a connection between the primary interface, the secondary interface, and the core of functional logic on the die.

7.1 General introduction

7.1.1 Application of the FPP

The optional FPP is intended for carrying arbitrary test data, clock, and control signals up and down the die stack independent of the die wrapper.

The configuration of the FPP can vary depending on the application. The FPP is composed of a set of lanes. Each lane implements a one-bit wide path. Lanes with identical properties and control may be grouped into channels.

Lanes can be unidirectional or bi-directional, registered (including pipe-lined pathways between terminals of the lane) or unregistered, and include a number of connection points between the bottom and top of the die. The collection of connection points are selectable according to the rules shown below, but may include terminals from the lane to the core and back, from one lane to another lane, and between the primary and secondary interfaces. Multiplexing functions within the lane can select how these terminals interconnect within the lane. Controls for these multiplexing functions are derived from test data register bits sprinkled throughout the scan-accessible network.

7.1.2 Flexible parallel port connectivity

Figure 23 part a) shows a die with only a primary interface. The die functional logic can be supplied with test data using the FPP connected to the primary interface.

Figure 23 part b) illustrates a channel connecting the primary interface to the secondary interface. In this case, the channel is used for transferring test data through the die. A channel consists of lanes which are built identically. Thus, multiple channels may exist in a die. A direction of the data flow is not specified in this conceptual figure.

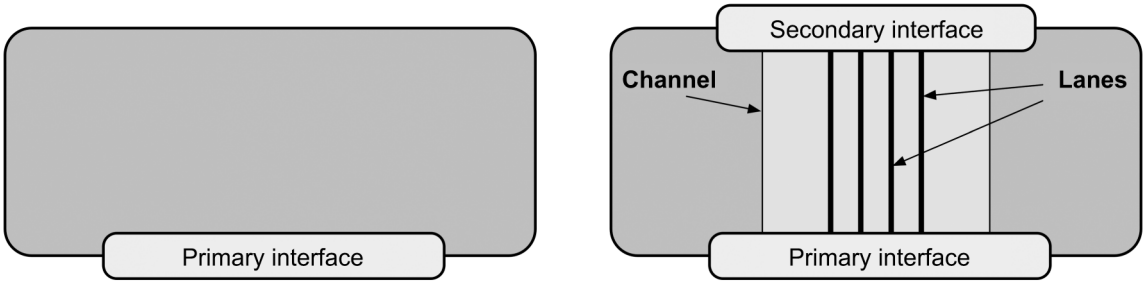

Figure 23—Die with primary and optionally secondary interface

In Figure 24 part a), two channels are required because one channel is bidirectional whereas the other channel is unidirectional.

In Figure 24 part b), a channel originates from a set of primary interface terminals and connects to two secondary interface sets of terminals. The primary interface terminal for two lanes could even be the same, such that a broadcast set of connections could be created. A direction of the data flow is not specified in this conceptual figure.

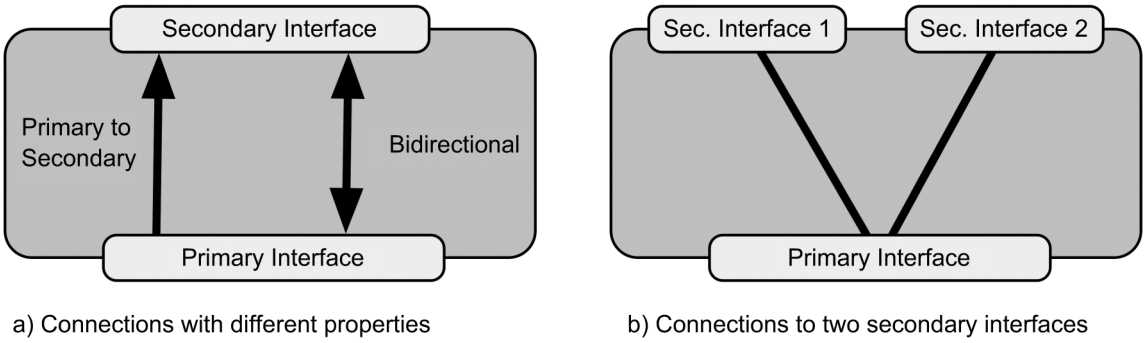

Figure 24—Different connection types

7.1.3 Interface connectivity

The purpose of the FPP is to enable parallel data flow into each die and between dies in the stack. In addition to the inter-die connection, it is also possible to connect the lanes with the core logic of the current die. A lane with the maximal connectivity is shown in Figure 25. In addition, lane-to-lane connections can be made, as illustrated.

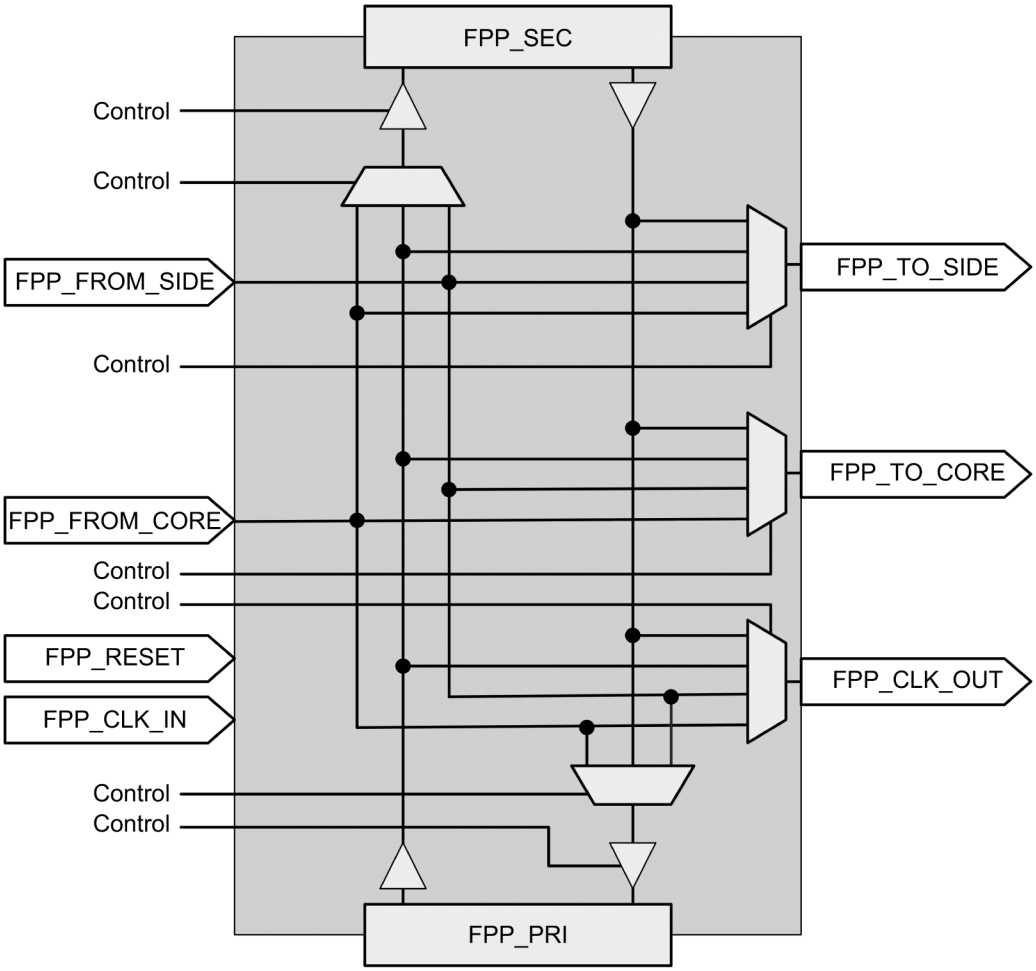

Figure 25—Template of lane connectivity architecture

A lane allows connections to system terminals, named as follows:

Table 2—FPP terminal definitions

FPP_PRI	Input, Output, or Bidirectional	Connection to a die terminal closer to the PCB interface
FPP_SEC	Input, Output, or Bidirectional	Connection to a die terminal further away from the PCB interface
FPP_FROM_CORE	Input	Connection from the core of the die into the lane
FPP_TO_CORE	Output	Connection from the lane into the core of the die
FPP_FROM_SIDE	Input	Connection from the FPP_TO_SIDE terminal of another lane
FPP_TO_SIDE	Output	Connection to the FPP_FROM_SIDE terminal of another lane
FPP_CLK_OUT	Output	Connection terminal on a clock lane driving a lane clock signal
FPP_CLK_IN	Input	Connection terminal of a clock signal for registers contained within the lane

Connections to primary and secondary die terminals can be bidirectional. Table 3 provides possible paths between FPP terminals. An X in the table represents a valid path between an input terminal (Source) and an output terminal (Sink).

Table 3—Undirectional lane connectivity

	Sink				
Source	**FPP_PRI**	**FPP_SEC**	**FPP_TO_CORE**	**FPP_TO_SIDE**	**FPP_CLK_OUT**
FPP_PRI		X	X	X	X
FPP_SEC	X		X	X	X
FPP_FROM_CORE	X	X	X	X	X
FPP_FROM_SIDE	X	X	X	X	X

A path describes the data flow from one input (source) terminal of a lane to one output (sink) terminal of a lane. There may be more than one path active in a lane at the same time. An input terminal can be the source for more than one path. The number of paths that are supported by a lane depends on the lane implementation. The number of simultaneously active paths and their functionality at a certain point in time depends on the settings of the FPP configuration elements.

7.2 FPP lane examples

The following schematics show example implementations for FPP lanes. The drivers on the primary or secondary interfaces should be able to enter a high impedance state for test reasons, in particular for testing before wafer thinning. The FPP configuration elements are the source of the control signals. The control terminal names in the examples are chosen to describe their functionality; they are not part of this standard.

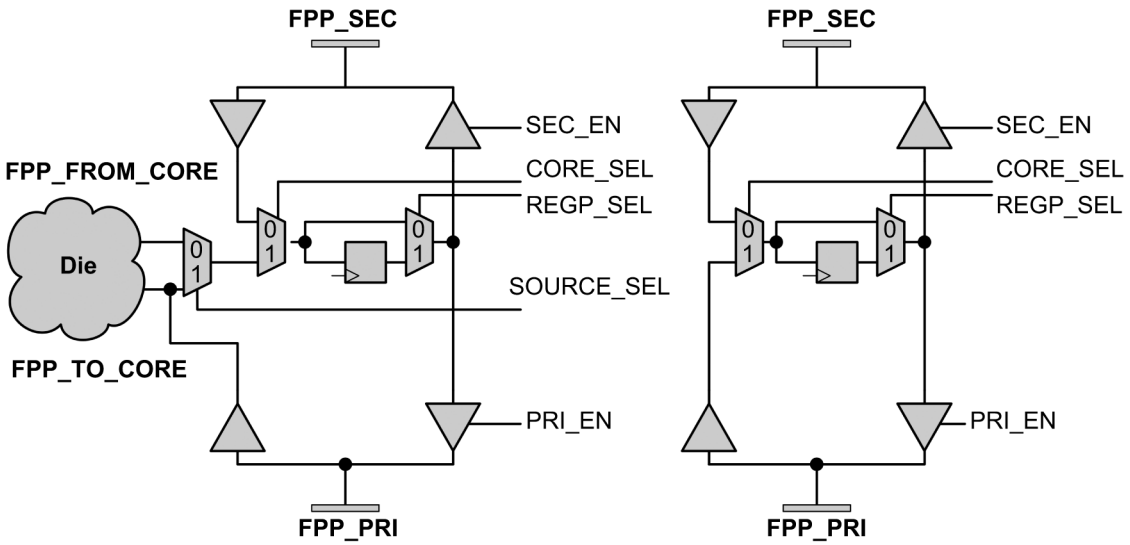

Figure 26—Bidirectional lane examples

Figure 26 provides two examples of a bidirectional lane. Figure 26 part a) is a variant supporting a connection to a die functional core signal via lane sink terminal FPP_TO_CORE and from a die functional core signal via lane source terminal FPP_FROM_CORE. If the lane does not support a connection to the core, Figure 26 part b) illustrates a potential implementation, showing only connections to the FPP_PRI and FPP_SEC bidirectional terminals.

Figure 27 shows the two possible lane implementations for a unidirectional connection, part a) with and part b) without a connection to the functional core logic of the die.

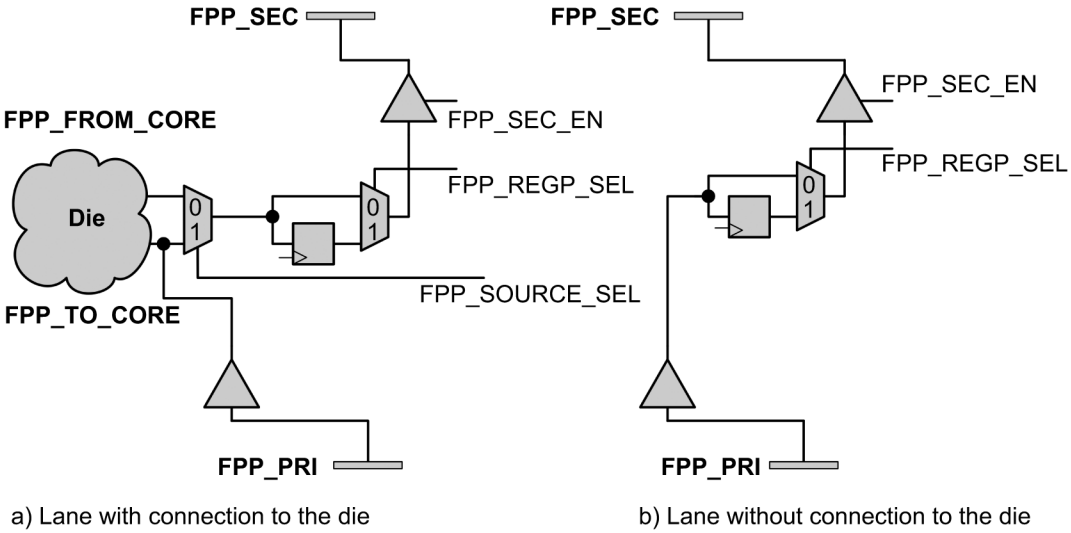

Figure 27—Unidirectional lane examples from primary to secondary port (up)

Figure 28 shows a similar set of examples, but going in the other direction.

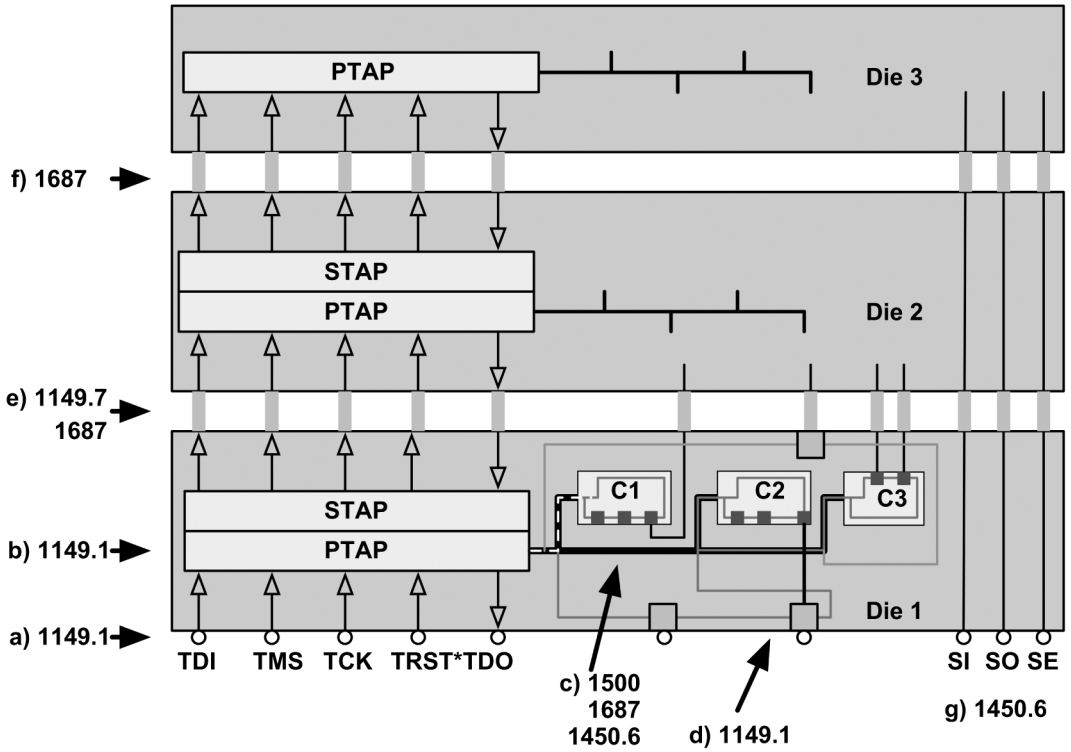

Figure 28—Unidirectional lane examples from secondary to primary port (down)

7.3 Structure of the FPP

7.3.1 Specification

7.3.1.1 Top-level

The rules are as follow:

a) A Flexible Parallel Port shall be composed of FPP lanes and/or FPP channels.

b) An FPP channel shall be composed of identical lanes with identical FPP lane paths, clocking, and control.

c) FPP lane control terminals shall only be controlled by FPP configuration elements accessible via the PTAP.

NOTE—If the FPP_PRI or FPP_SEC terminals share functional output ports of the die, the signals that control the selection of the lane terminal and the enable of the output driver is derived from FPP configuration elememtns and/or other signals sourced by FPP lanes. This defines how to share a functional port with an FPP lane terminal.

d) The reset state of the FPP lane shall enable the functional pathway.

The recommendations are as follows:

e) If FPP_PRI is present and either output-only or bidirectional in nature, the connection from the FPP_PRI terminal to the primary port should be driven by three-state drivers.

f) If FPP_SEC is present and either output-only or bidirectional in nature, the connection from the FPP_SEC terminal to the secondary port should be driven by three-state drivers.

The permissions are as follows:

g) Different channels may be driven by separate clock lanes.

h) A clock lane may drive more than one channel.

i) Combinational logic may be between the FPP configuration elements and the FPP lane control terminals.

j) The FPP_PRI and FPP_SEC terminals of each lane may be connected to functional ports of the die.

7.3.1.2 Lane

The rules are as follows:

a) All lane terminals except FPP_PRI and FPP_SEC shall be unidirectional.

b) Lanes shall be defined as either registered or non-registered.

c) If a lane has an FPP_PRI terminal, it shall connect to a terminal of the primary interface of the die.

d) If a lane has an FPP_SEC terminal, it shall connect to a terminal of a secondary interface of the die.

e) If a lane has an FPP_TO_CORE terminal, it shall connect to circuitry on the same die outside of the FPP.

f) If a lane has an FPP_FROM_CORE terminal, it shall connect to circuitry on the same die outside of the FPP.

g) If a lane has an FPP_FROM_SIDE terminal, it shall connect to the FPP_TO_SIDE terminal of another lane.

h) If a lane has an FPP_TO_SIDE terminal, it shall connect to the FPP_FROM_SIDE terminal of another lane.

i) If the FPP_PRI terminal is bi-directional, it shall provide an output driver with a high-impedance state.

j) If the FPP_SEC terminal is bi-directional, it shall provide an output driver with a high-impedance state.

k) If the output driver of the FPP_PRI terminal of a lane exists and can be set to a high-impedance state, then

   1) there shall be a dedicated lane control terminal which controls the high-impedance state,

   2) a signal connected to this dedicated terminal and derived from FPP configuration elements shall control the enable of the output driver, and

   3) if the FPP configuration elements are reset, this signal shall select its high-impedance state.

l) If the output driver of the FPP_SEC terminal of a lane exists and can be set to a high-impedance state, then

   1) there shall be a dedicated lane control terminal which controls the high-impedance state,

   2) a signal connected to this dedicated terminal and derived from FPP configuration elements shall control the enable of the output driver, and

   3) if the FPP configuration elements are reset, this signal shall select its high-impedance state.

m) Each FPP lane shall be configurable into at least one FPP lane path.

n) An FPP lane sink shall be driven by only one selected source at a time.

o) If an FPP lane sink may be driven from multiple sources, then one or more FPP lane control terminals shall be provided which select one source.

The recommendation is as follows:

p) A lane specified as a clock lane should be dedicated for the purpose of sourcing a lane clock.

The permissions are as follows:

q) The FPP_PRI terminal may be unidirectional or bi-directional.

r) The FPP_SEC terminal may be unidirectional or bi-directional.

s) The design of the circuitry fed from the FPP_PRI and FPP_SEC terminals may be such that an undriven source produces a logical response identical to either the application of a logic 0 or a logic 1.

t) Each lane may be configurable into up to three active FPP lane paths simultaneously.

u) Multiple FPP lanes may share a connection to the same source terminal.

   NOTE—For example, the FPP_PRI terminals of two lanes could be connected to the same primary interface input terminal.

7.3.1.3 Registered lane

The rules are as follows:

a) A registered lane path shall have at least one terminal from each set of Sources and Sinks shown below, without duplicate:

   1) Sources: FPP_PRI, FPP_SEC, FPP_FROM_CORE, FPP_FROM_SIDE,

   2) Sinks: FPP_PRI, FPP_SEC, FPP_TO_CORE, FPP_TO_SIDE.

   NOTE—"Without duplicate" implies that the same source and sink cannot be used to create a lane path. For example, choosing FPP_PRI as source *and* sink is not allowed.

b) Each registered lane path shall contain at least one pipeline register and at least one hold-time element.

c) Between the FPP_PRI and FPP_SEC terminals, if they both exist, shall be at least one pipeline register and one hold-time element.

   NOTE—Permissions 7.3.1.3 n) and 7.3.1.3 o) indicate that these storage elements may be bypassable. In addition, these storage elements, though they may be different, should be between the two terminals in either direction.

d) In a registered lane, the FPP_CLK_IN input terminal shall be provided that connects a lane clock to the pipe-line registers and hold-time elements within the lane.

e) In a registered lane, pipe-line registers and hold-time elements shall be clocked by the lane clock.

f) If pipe-line registers are provided in the lane path, then at least the first of these registers shall be clocked by the rising edge of the lane clock.

g) If a hold-time element is provided in the lane path, it shall be clocked by the falling edge of the lane clock.

h) A hold-time element, if provided, shall be formed out of a single storage element, and shall be the last storage element prior to the FPP_PRI and/or the FPP_SEC and/or the FPP_TO_CORE and/or the FPP_TO_SIDE terminals.

i) If pipe-line registers within a lane are resettable, then an FPP lane control terminal, FPP_RESET, shall be provided that resets these registers.

j) The signal driving the FPP_RESET terminal shall be asserted upon resetting the Flexible Parallel Port configuration elements.

   NOTE—This means that the pipe-line bits could follow the same resetting rules as the FPP configuration elements.

The recommendation is as follows:

k) In a registered lane, the lane clock should be provided by a dedicated clock lane via the primary interface.

The permissions are as follows:

l) A registered lane path may connect its source and sink terminals in all cases where "X" is indicated in Table 4.

Table 4—Registered lane connectivity

	Sink			
Souree	**FPP_PRI**	**FPP_SEC**	**FPP_TO_CORE**	**FPP_TO_SIDE**
FPP_PRI		X	X	X
FPP_SEC	X		X	X
FPP_FROM_CORE	X	X	X	X
FPP_FROM_SIDE	X	X	X	X

m) In a registered lane path, the lane clock may be sourced from TCK.

n) In a registered lane path, a mechanism may be implemented for bypassing the registration in that lane.

o) In a registered lane path, a mechanism may be implemented for bypassing the hold-time element in that lane.

p) Registered lane paths may provide more than one pipe-line register bit per registered lane path.

q) Pipe-line register bits may be clocked by either the rising or falling edge of the lane clock except as noted in rule 7.3.1.3 f).

r) The FPP configuration elements may be equipped with a Config-Hold function which, if TRSTN is not asserted, causes the elements not to revert to a default state when the test controller associated with the primary serial test interface synchronously enters the Test-Logic-Reset state.

NOTE—This does not apply if TRSTN is asserted. Asserting TRSTN will initialize the FPP configuration elements to their respective default state.

s) The FPP_RESET terminal may be driven by the RESET* signal from the PTAP controller.

7.3.1.4 Non-registered lane

The rules are as follows:

a) A non-registered lane path shall have at least one terminal from each set of Sources and Sinks shown below, without duplicate:

1) Sources: FPP_PRI, FPP_SEC, FPP_FROM_CORE, FPP_FROM_SIDE

2) Sinks: FPP_PRI, FPP_SEC, FPP_TO_CORE, FPP_TO_SIDE, FPP_CLK_OUT

b) A non-registered lane shall not contain pipe-line registers or hold-time elements.

c) Non-registered lanes defined as clock lanes shall use the FPP_CLK_OUT terminal to generate a lane clock for the FPP_CLK_IN terminals of registered lanes.

The permission is as follows:

d) A non-registered lane path may connect its source and sink terminals in all cases where "X" is indicated in Table 5.

Table 5—Non-registered lane connectivity

	Sink				
Source	**FPP_PRI**	**FPP_SEC**	**FPP_TO_CORE**	**FPP_TO_SIDE**	**FPP_CLK_OUT**
FPP_PRI		X	X	X	X
FPP_SEC	X		X	X	X
FPP_FROM_CORE	X	X	X	X	X
FPP_FROM_SIDE	X	X	X	X	X

7.3.2 Description

The optional FPP may be composed of registered and non-registered lanes; the latter category can be further subdivided into clock lanes and non-clock lanes. The configuration of each of the lane elements is done with PTAP-accessible register bits (TDRs) that will hold their state as the FPP is being used to apply tests. The lanes can each be controlled (if a pathway exists) to connect signals of the primary interface, secondary interface, and core to each other. These pathways might be registered to enable higher-frequency data communications across them. If registered, a clock can be provided from various sources. But it is envisioned that a clock might be easily connected through a non-registered lane, so special clock terminal names were selected (FPP_CLK_IN and FPP_CLK_OUT). The registration can be bypassed if desired.

Due to wafer-level manufacturing requirements, it might be desirable to have connection to the primary and secondary interfaces be able to be set in a high-impedance state if the lane connection is an output or a bi-directional terminal.

7.4 Allocation of FPP configuration elements to the FPP lane control terminals

7.4.1 Specifications

The rule is as follows:

a) Each control terminal of each FPP lane shall be driven by one true or inverted bit of an FPP configuration element, or by a value derived from a combinational circuit driven by multiple FPP configuration elements.

The recommendation is as follows:

b) Each FPP configuration element should control only one FPP control terminal.

7.4.2 Description

The registers that configure the FPP can directly drive a lane configuration terminal (such as a multiplexer select or a tri-state enable signal). But, the bits can also be combined with some sort of combinational circuit to select the control function.

8. IEEE Std 1838 DWR relationship with other standards

Figure 29 depicts a 3 die stack using die compliant to this standard. Other IEEE standards could support this one in the following ways:

— As mentioned throughout this document, IEEE Std 1149.1 is leveraged for the serial control and datapath. The stack-level and die-level interface is specified with respect to the 5 TAP interface signals.

— The PTAP controller and register architecture is based off of an IEEE 1149.1 TAP controller and register architecture. The instruction register addresses device-level registers to be placed in the TDI-to-TDO pathway.

— The DWR is quite flexible in its construction. As such, an IEEE 1500 core wrapper could become part of its content.

— Selection of elements into and out of its construction at any given time could be accomplished with WIRs from the core, or SIBs (as described in IEEE Std 1687). As well, the construction of the DWR could be described by IEEE Std 1149.1 (BSDL), IEEE Std 1687 (ICL), or IEEE Std 1450.6 (CTL).

— The flexibility of the DWR hardware can also extend to including boundary-scan register bits and segments as described in IEEE Std 1149.1 and by BSDL from that standard.

— To describe the content of the stack, IEEE Std 1149.7 offers some HSDL language constructs. IEEE Std 1687 may also be leveraged to describe the hierarchy.

— IEEE Std 1687 may also be able to use its ICL language to describe the serial pathway from the PTAP through the connected STAPs.

— The optional FPP may be able to be described by IEEE Std 1450.6.

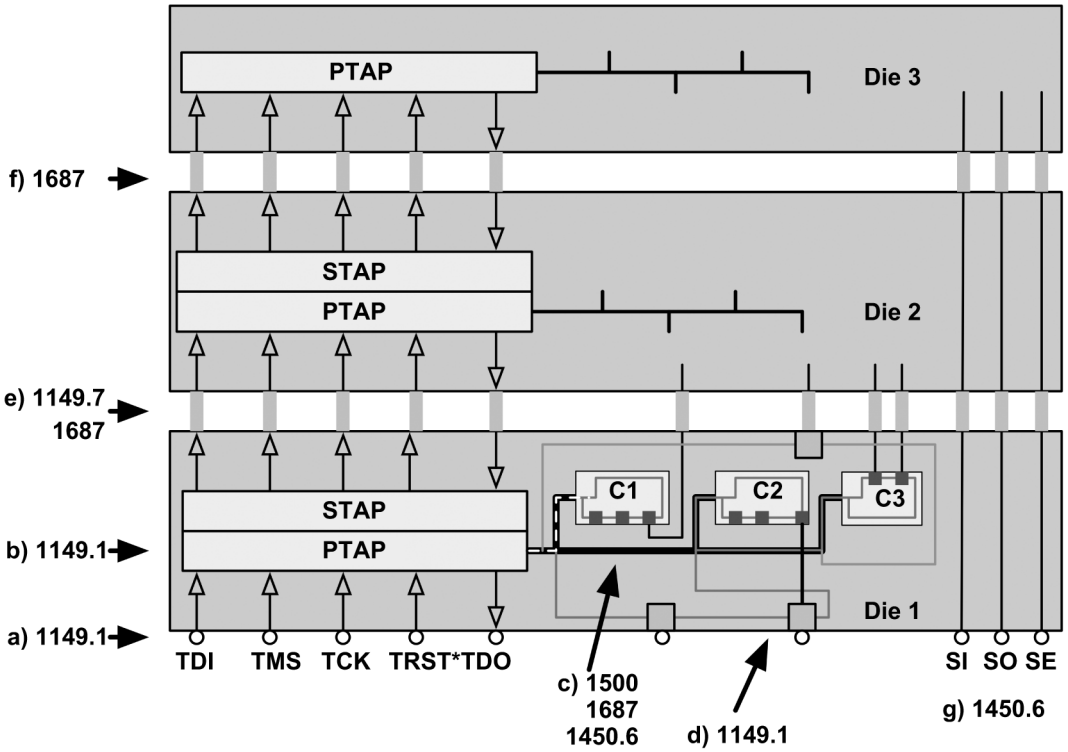

Figure 29—3D die stack using 1838 compliant dies

Annex A

(informative)

Bubble diagrams

In IEEE Std 1838, an attempt has been made to focus on architecture and remain neutral to implementation styles. In this spirit, a graphical vocabulary has been implemented to describe hardware behavior visually without describing the physical implementation of the hardware. Informally, diagrams drawn using these symbols have been called “bubble diagrams” because the much-used symbol for a storage element is a circle.

The symbols to describe behavior are few, and have been re-used from IEEE Std 1500. In fact, there are only four as shown in Figure A.1. The first is the symbol for a storage element, which is a circle. The second is a data path, which is represented as a line with an arrowhead indicating data flow direction. The third is a dataflow selection (multiplexer), which is represented as a vertical line with no arrowhead. The fourth symbol is a connection point, represented as a small filled circle located at the junction of two data paths.

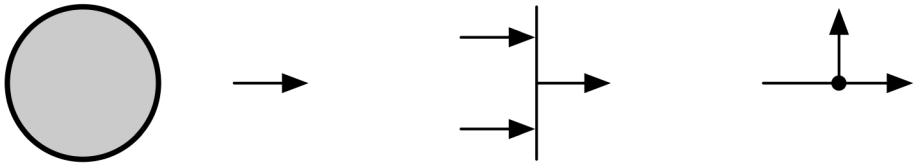

Figure A.1—The four symbols used in bubble diagrams

The storage element has a characteristic associated with it that indicates the events to which it responds. This characteristic is displayed as the presence of one or more characters inside the circle from the set {S, C, U, and F} indicating the Shift, Capture, Update, and any Functional event, respectively. Figure A.2 shows various storage elements.

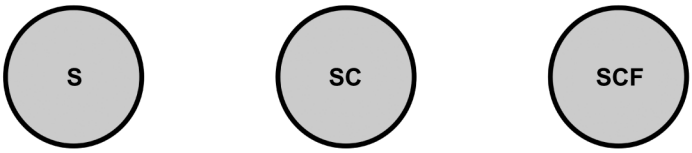

Figure A.2—Various storage elements

Shift data flows from CTI to CTO and can go through multiple storage elements represented by bubbles. Capture data come into a storage element from the CFI terminal or the CFO terminal. Update data comes from the shift path storage element. Functional data comes from CFI.

To show how these elements are used together, an IEEE Std 1149.1 BC_1 cell using the bubble diagram notation is shown in Figure A.2. The BC_1 cell supports the Shift, Capture, Update, and Apply events. It should be noted that the Apply event is a virtual event, composed of other events or modes, and, therefore, is not specifically represented in these diagrams.

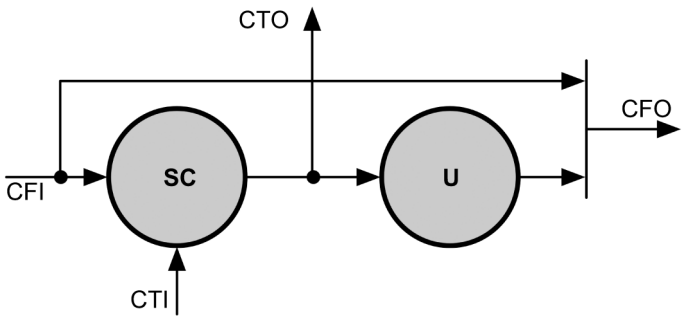

Figure A.3—Example of IEEE Std 1149.1 BC_1 or IEEE STD 1838 DC_SD1_CI1_U cell

Annex B

(informative)

Bibliography

Bibliographical references are resources that provide additional or helpful material but do not need to be understood or used to implement this standard. Reference to these resources is made for informational use only.

[B1] Deutsch, S., B. Keller, V. Chickermane, S. Mukherjee, N. Sood, S. K. Goel, J.-J. Chen, A. Mehta, F. Lee, and E. J. Marinissen, "DfT Architecture and ATPG for Interconnect Tests of JEDEC Wide-IO Memory-on-Logic Die Stacks," IEEE International Test Conference (ITC'12), Anaheim, California, USA, November 2012.

[B2] Durupt, J., P. Vivet, and J. Schlöffel, "IJTAG Supported 3D DfT Using Chiplet-Footprints for Testing Multi-Chips Active Interposer System," 21st IEEE European Test Symposium (ETS), Amsterdam, the Netherlands, May 2016, DOI: http://dx.doi.org/10.1109/ETS.2016.7519310.

[B3] Eklow, B., "Major Milestones for Two IEEE Standard Groups in 2011," IEEE Design & Test, Volume 28, No. 6 (November/December 2011), pp. 85-87, DOI: http://dx.doi.org/10.1109/MDT.2011.128.

[B4] Fkih, Y., P. Vivet, B. Rouzeyre, M.-L. Flottes, and G. Di Natale, "A JTAG-Based 3D-DfT Architecture Using Automatic Die Detection," 9th Conference on PhD Research in Microelectronics and Electronics (PRIME), Villach, Austria, June 2013, DOI: http://dx.doi.org/10.1109/PRIME.2013.6603184.

[B5] Li, Y., M. Shao, H. Jiao, A. Cron, S. Bhatia, and E. J. Marinissen, "IEEE Std P1838's Flexible Parallel Port and its Specification with Google's Protocol Buffers," IEEE European Test Symposium (ETS'18), Bremen, Germany, May 2018.

[B6] Marinissen, E. J., B. De Wachter, S. O'Loughlin, S. Deutsch, C. Papameletis, and T. Burgherr, "Vesuvius-3D: A 3D-DfT Demonstrator," IEEE International Test Conference (ITC'14), Seattle, Washington, USA, October 2014, Paper 20.2,, http://dx.doi.org/10.1109/TEST.2014.7035332.

[B7] Marinissen, E. J., C.-C. Chi, J. Verbree, and M. Konijnenburg, "3D DfT Architecture for Pre-Bond and Post-Bond Testing," IEEE Intl. 3D Systems Integration Conference (3DIC'10), Munich, Germany, November 2010, Paper 3B.1, http://dx.doi.org/10.1109/3DIC.2010.5751450.

[B8] Marinissen, E. J., C.-C. Chi, M. Konijnenburg, and J. Verbree, "A DfT Architecture for 3D-SICs Based on a Standardizable Die Wrapper," Journal of Electronic Testing: Theory and Applications (JETTA), Special Issue on 3D-Testing, Volume 28, No. 1 (February 2012), pp. 73-92, http://dx.doi.org/10.1007/s10836-011-5269-9.

[B9] Marinissen, E. J., J. Verbree, and M. Konijnenburg, "A Structured and Scalable Test Access Architecture for TSV-Based 3D Stacked ICs," IEEE VLSI Test Symposium (VTS'10), Santa Cruz, California, USA, April 2010, pp. 269-274,,,, http://dx.doi.org/10.1109/VTS.2010.5469556.

[B10] Marinissen, E. J., T. McLaurin, and H. Jiao, "IEEE Std P1838: DfT Standard-under-Development for 2.5D-, 3D-, and 5.5D-SICs," 21st IEEE European Test Symposium (ETS), Amsterdam, the Netherlands, May 2016, DOI: http://dx.doi.org/10.1109/ETS.2016.7519330.

[B11] Papameletis, C., B. Keller, V. Chickermane, S. Hamdioui, and E. J. Marinissen, "A DfT Architecture and Tool Flow for 3D-SICs with Test Data Compression, Embedded Cores, and Multiple Towers." IEEE Design & Test, Volume 32, No. 4 (July/August 2015), pp. 40-48.

IEEE SA
STANDARDS ASSOCIATION

RAISING THE WORLD'S STANDARDS

Connect with us on:

Twitter: twitter.com/ieeesa
Facebook: facebook.com/ieeesa
LinkedIn: linkedin.com/groups/1791118
Beyond Standards blog: beyondstandards.ieee.org
YouTube: youtube.com/ieeesa

standards.ieee.org
Phone: +1 732 981 0060

IEEE