Special Session: Test Challenges in a Chiplet Marketplace

M. Hutner
Teradyne, Toronto, Canada
marc.hutner@teradyne.com

R. Sethuram
nVidia, Santa Clara, USA
rsethuram@nvidia.com

B. Vinnakota
Broadcom Inc, San Jose, USA
bapi.vinnakota@ocproject.net

D. Armstrong
Advantest, USA
dave.armstrong@advantest.com

A. Copperhall
Facebook, Menlo Park, USA
acopperhall@fb.com

Abstract—**Chiplet-based designs enable the heterogeneous integration of die from multiple process nodes into a single packaged product. High-bandwidth memory is a well-known high-volume chiplet-based product. Today, most chiplet-based logic products are single-vendor products built with proprietary die-to-die (D2D) interfaces between the chiplets in a package. Industry and academia have developed standards and methods to address the test challenges expected with chiplet-based products. These efforts have focused on improving die yield and test access structures. The Open Compute Project's Open Domain-Specific Architecture (ODSA) is a new effort that aims to define an open physical and logical D2D interface and create a marketplace of chiplets. With an open interface, product developers can integrate best in class chiplets from multiple vendors. An open D2D interface offers both new opportunities and challenges in testing multi-chiplet products. The opportunity is in leveraging economies of scale. The challenge is in enabling greater interoperability between test structures in different chiplets and across vendors. This paper reviews recent developments in chiplet test, especially leveraging work on HBM and discusses their extension to testing products based on open D2D interfaces.**

Keywords—ODSA, 2.5D/3D, chiplet, HBM, D2D, KGD, IEEE1149, IEEE1687, IEEE1838, Test, CoT, BoW, HIR

I. Introduction

Chiplet-based products require physical and logical die-to-die (D2D) interfaces between the chiplets integrated into a package [1]. Today, most chiplet-based products are based on closed interfaces and integrate die from one company, or require direct inter-company technical, workflow and business negotiations. The Open Domain-Specific Architecture (ODSA) project aims to define open physical and logical D2D interfaces between chiplets [2]. These interfaces can potentially enable an open marketplace such that product designers can integrate best-in-class chiplets from multiple vendors and multiple process nodes to create novel logic products. Conversely, chiplet vendors accrue the benefits of scale by being used in a wider range of products. Beyond logical and physical interfaces, product designers will have to address packaging and test challenges in integrating chiplets from multiple vendors. We discuss how work in a commercial chiplet-based product based on open interfaces, HBM and scan test standards for 2.5D/3D integration can be extended to logic chiplets in an open marketplace.

As one thinks about the ODSA chiplet ecosystem, where several logic chiplets from multiple vendors are combined into a single package, what will the test challenges look like? One can look back at complex circuit board faults and get a sense of what is to come. One source of defects in a chiplet-based product are defects in the components. Faults do occur in an MCM package due to IO switching and compute intensity in today's processor packages. The root cause of these faults could be due to power distribution, bus cross-talk, and/or computational use cases not envisioned as a single die or physical faults injected at assembly. As with boards, test access standards to help address these problems have been investigated and developed. This paper is organized as follows. In Section 2, we discuss the known good die problem and relevant technologies from board test and High-Bandwidth Memory test that can be applied to chiplet-based products. In Section 3, we review test access methodologies relevant to testing chiplets from board test, IP test and newer 3D test standards.

Prior discussions on chiplets have usually not directly addressed the unique challenges of assembling products that integrate chiplets from multiple vendors [3,4,5]. In Section 4, we discuss examples of challenges expected in a marketplace model. Our discussion focuses on the need for both logic and information interoperability. Our discussion focuses on three issues: (a) how test workflows may need to change to better provide known good die in a marketplace; (b) how test access from multiple vendors need to interoperate for interconnect, final product test and bring up; (c) an approach to reduce the cost of probe testing chiplets that use open standard die-do-die interfaces. This document summarizes and expands upon panel discussions in ODSA workshops held at IBM and Facebook in September and December, 2019. In the rest of this document, we discuss each of these issues. We start with a brief review of the Open Domain-Specific Architecture project and of chiplet test challenges.

II. Known Good Die

When integrating multiple chiplets into a product, the higher the yield of the component parts, the lower the yield loss in final assembly. Test engineers have recognized the importance of known good die (KGD) in multi-die products for several decades [6,7,8]. With current packaging technologies, a die

978-1-7281-5359-9/20/$31.00 ©2020 IEEE

cannot be removed from a substrate to which it is bonded without damaging it. Therefore, when any die in an assembled multi-chiplet product is defective, all the die in the package have to be scrapped, significantly raising the cost of component defects. The KGD issue has received renewed attention with the increased interest in multi-chiplet products [1].

For example, wafer testing of logic chiplets can be expanded to include speed, power, and die harvesting at the chiplet level. Speed is an interesting test to consider as true speed performance may not be guaranteed at wafer level due to insufficient power supply requirements. Therefore, a derated, characterized, or proxy of speed may need to be used. However, this all raises the cost of chiplet test at wafer probe. What is the right way to balance the cost yield tradeoff? One question for traditional testing methods at wafer and package test is will it be sufficient to ensure sufficient yield for the final product. Will additional coverage be needed, or will new faults arise, or faults be introduced due to the packaging complexity. One only needs to look at board assembly today to see that package interactions between various complex SOC cause additional system issues. These will need to be cooperatively addressed by both the Electronic Design Automation (EDA) industry pre-silicon and the Automatic Test Equipment (ATE) Industry post-silicon.

A. KGD Impact

Economics is the primary driver for where tests are applied in the manufacturing chain. Ideally at the first wafer test step, coverage is achieved with a simple test setup - limited cooling, lower speed digital communication, limited site count and shorter test times. An example of where the cost-of-test for wafer test would significantly increase is if active cooling is required. The cooling can be achieved under the wafer, where the signals are applied above the wafer. This test step is also done before wafers are thinned where the thick wafer acts like an insulator. As a result, additional cooling is required to achieve the desired temperature at wafer test. Many of these challenges can be addressed at package test today because the functional (product) environment can be partially leveraged. For instance, the cooling concepts from a final product can be leveraged into the automatic temperature control (ATC) cooling head used for testing. High speed materials can also be leveraged for signal delivery. In most cases ATE instruments can be used to develop both structural and functional waveforms to be applied to a Device Under Test. In extreme cases for package testing, a golden device could talk to the Device Under Test for special test coverage (this is less than ideal due to reliability, traceability, repeatability and continuity of Golden Units). The tradeoff is the cost of the test solution versus the cost of scrap of the Device Under Test. Ideally one minimizes scrap by doing testing earlier, but it does depend on the incremental test cost increase of the test setup.

Let’s take for example a potential smart computing element targeted for an edge device. This may then be connected to several processing elements in one of the die. To support these processors, we would need some memory. If you start with five die multi chip module where

Table 1. Component Cost

	Cost	*Area*
Dies 1,2	*$100*	*$100mm^2$*
Dies 3,4,5	*$20*	*$25mm^2$*
Packaging	*$30*	
Assembly	*$5*	

This example does not reflect real devices and numbers were selected as an example.

In the marketplace it is assumed that all devices are tested to a level where they are considered known good die. In reality, there are always coverage holes, as well as inter and intra die dependencies (not being able to work together at the X product of a signal margin for example) that may impact the final package test or the assembled product. For this example the total packaged device costs $295. If the final package test yield is 90%, the final product cost will be $327.78 because of the cost of the scrap material.

Yielded Cost = (Assembly Cost)/Yield

At the final assembler of the packaged part, the goal would be to determine what are the primary causes of the 90% package yield. If it is only assembly issues, then the actions are clear. The yield issues may also be attributed to a coverage issue for one or more of the dies in the package. The tradeoffs are not new for our industry, the challenges only become more for larger die-in-package integration. The tools developed for understanding board yields are applicable in the system level tradeoffs for package level assembly. The difference is that the test intensity and test cost may change because the cost of the end product is much higher. Part of the ODSA effort for test challenges should also focus on end-to-end test cost modeling and reliability modeling.

B. Die Testing Trade-offs

For a heterogeneous packaged device, the dies in the product may be sourced from multiple vendors. Ideally the only faults that occur in a packaged device would be assembly faults. This implies that for the chiplet vendor, all testing is conducted at the chiplet layer in a die form either at the wafer or singulated die. It also means that the tests applied on the die need to have the highest-level coverage possible, and the widest of signal margins, to avoid faults propagating to the end package.

If you have a large logic die in an advanced technology, for highly parallel device-to-device interfaces (D2D) the number of D2D wires could exceed a thousand. Product test engineers would like to avoid probing all of these D2D interconnects. Today one common class of chiplet-based products are those assembled are a large logic System-on-chip (SoC) attached to one or more HBM memories such as the AMD Fury. In the case of these multi-chip packages, the memory interfaces are designed to have lane repair and the memories can also be repaired. In addition to this, additional test steps are performed to ensure network traffic is verified between the memories and the SOC. This could be a simple PRBS based test [9], pseudo traffic test, or a system level test. The question becomes what the coverage of each type of test and the cost of is developing and deploying each test.

If each die is targeted for chiplet application, they may have been optimized for on-package communications. An example of this is where a D2D interface would have been designed to drive up to 5-10 mm of package trace vs traditionally they would have been designed to drive 12 inches of board trace (the difference of board trace and package trace in terms of length as well as transport medium is significant). External package interfaces will be designed for higher power, but that should be less than 10% of the potential interfaces for all the dies that are contained in the package. This will create a probing problem for each die individual because if they are testing an output driver, they may not be able to test it at speed due to drive strength issues. It may be difficult to determine if speed issues will occur on the interface. Each die may be sourced by different companies and foundries. The exact combination and interaction of dies may not have even been envisioned by any of the design teams for each of the chips. Each chip provider does not want to be indicted as the reason why the yield of a fully assembled package is not functional.

C. *Learning from Memory*

Memory has been on the forefront of die stacking and there are many lessons that can be learned from it. Relative to chiplet-based SoCs, the one major difference when one looks at an High Bandwidth Memory (HBM) memory is they are stacking two die types. The memory die and the logic die. The Memory die is replicated many times, say 8 or 16 in a stack, with a single logic die acting as an interface. This configuration can be seen in Figure 1.

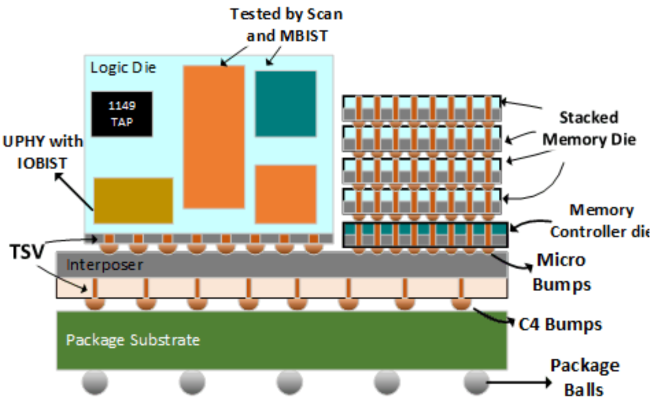

Figure 1. Packaged IC with logic die along-side stacked memory dies

The example above shows a multi-die package which contains a logic die which communicates with a stacked memory using a silicon interposer. It shows a variety of connection technologies which include Through Silicon Vias (TSV) which connect through a die. What learnings can be taken from memory to this new class of heterogeneous integration? [21,22,23] The question is how to optimize yield. It should be noted that each DRAM die is repairable. This enables very high yield. The question is how many spare redundant blocks are necessary to ensure die yields of >95% or higher. For HBM memories there is also an option of lane repair. Similar discussions have also occurred in the 3DIC test research community when Through Silicon Via (TSV)s emerged as a viable technology. It is also beneficial to have redundant IOs which could be achieved using an interface like a Bunch-of-Wires (BoW). It is reasonable to assume that a redundant IO could be added to every "n" IOs to enable high attach yield. This methodology will need to be adopted by both top and bottom dies in a die stack. The configuration of repair can be easily handled by an on die test wrapper, test interface and an efuse or soft repair storage mechanism. However, consideration of the additional IO's, routing and signal integrity need to be considered.

If there are common functions, the designers may consider redundant cores that can be swapped in at various manufacturing steps. This has other benefits in that various manufactures can comply to a standard which ensures a robust chip marketplace. There are additional benefits at the system level, where these repair features can be used at power on or in a periodic health check. This will help to show die aging and be able to further in system diagnostics. It also enables a secondary marketplace for packages that don't make the full grade or quality but could be sold for less to enable a less demanding application.

D. *Shift Left*

One of the key learnings from early memory die stacking efforts was how critical it is to do as much testing as possible, as early as possible in the manufacturing flow. In the past, many critical steps such as sparing out portions of the circuit and setting of threshold voltages was deferred to the final test step. With stacked die devices the known-good-stack never has a final test step. Instead the final testing, fuse blowing, and burn-in is performed at the stack level just prior to shipping.

A similar shifting of test, to the left, can provide significant value to other multi-device assemblies.

Test flow refers to the sequence of tests which are applied throughout the manufacturing process. At wafer test you have a low scrap cost but a high equipment cost. The high equipment cost could be due to the complex interface requirements as we mentioned in the earlier section. This is compared to package, SLT and field test which have a lower test complexity cost, but a possibly significant higher scrap cost.

The HIR, Heterogeneous Integration Roadmap, has developed an assembly and test flowchart which enables visualization of a typical workflow. What one can see in the picture below is a variety of test steps for a multiple device product. The wafer-level and KGD tests are expected to be performed by a chiplet vendor, while assembly, final and system-level test are performed by the product vendor. In a multi-chiplet product, the yield on the final assembled product is a function of the test flow. That is, the yield of the assembled components, final test yield (including the interconnect) is a function of the KGD yield of the devices being assembled, together with the yield of the various critical tests which are newly applied at final test, as well as the yield associated with simply contacting the assembly . Specifically:

$$Yield_{FinalTest} = f(1\text{-}Test\ Coverage\ at\ KGD)^{\#Logic}\ x\ f(Yield_{HighTemp})\ x\ f(Yield_{HighFreq})\ x\ f(Yield_{Contacts})^{\#Bumps}$$

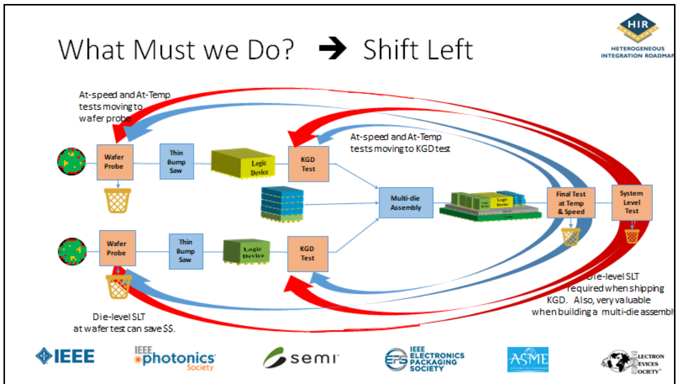
What Must we Do? → Shift Left
HIR
HETEROGENEOUS INTEGRATION ROADMAP
At-speed and At-Temp tests moving to wafer probe
At-speed and At-Temp tests moving to KGD test
Wafer Probe
Thin Bump Saw
Logic Device
KGD Test
Multi-die Assembly
Final Test at Temp & Speed
System Level Test
Wafer Probe
Thin Bump Saw
Logic Device
KGD Test
Die-level SLT required when shipping KGD. Also, very valuable when building a multi-die assembl
Die-level SLT at wafer test can save $$.
IEEE
IEEE photonics Society
semi
IEEE ELECTRONICS PACKAGING SOCIETY
ASME
ELECTRON DEVICES SOCIETY

Figure 2. Test shift-left proposal from the IEEE HIR

Figure 2 shows how test content is moving to the left in an effort to achieve a higher quality device sooner in the flow (and reject the bad devices quickly as well). It's important to note that the cost of test consumables (probe cards, DUT boards, etc.) as well as possibly test equipment, is higher the further you go to the left. At the same time, the cost of yield, that is the cost of rejecting a bad device, is lower. Optimizing the test content, test costs, and yield can have a significant impact on the cost of the final product.

III. The Evolution of Test Access Mechanism – from IEEE1149 to IEEE1838

The IEEE1149 standard was first introduced to perform board-level IC interconnect testing. As ICs became larger, more complex, and the design cycle kept shrinking, a scalable and interoperable test solution was needed as many teams had to collaborate to put together the test solution. To address the interoperability needs, IEEE1500 and IEEE1687 were developed to address scalability and test reusability issues. More recently IEEE1838 was approved that addresses the issues of testing 2.5D and 3D dies. In this Section we take the readers through a journey of the evolution of IEEE test access mechanism standards from IEEE1149 to IEEE1838. We will describe the requirements that drove the industry to adopt these standards while highlighting their differences in their use cases.

A. IEEE 1149 – The JTAG and Board-level Testing [10,11]

The emergence of JTAG (Joint Test Action Group - since disbanded) or IEEE1149 dates to the late 80's when In-Circuit testing was becoming popular. In-circuit testing involves testing each circuit component standalone using a bed of nail fixture. However, with advancements in packaging geometries and mounting technologies at that time in-circuit nodes were inaccessible to the nail fixture. They posed problems accessing the nodes to test the IC and get adequate test coverage. The main purpose of IEEE 1149 standard was to overcome the board level probing issues [12], support board-level testing and test setup in a non-intrusive fashion without sacrificing the IC test coverage. It was to do this electrically so that large PCB's could be tested quickly and efficiently.

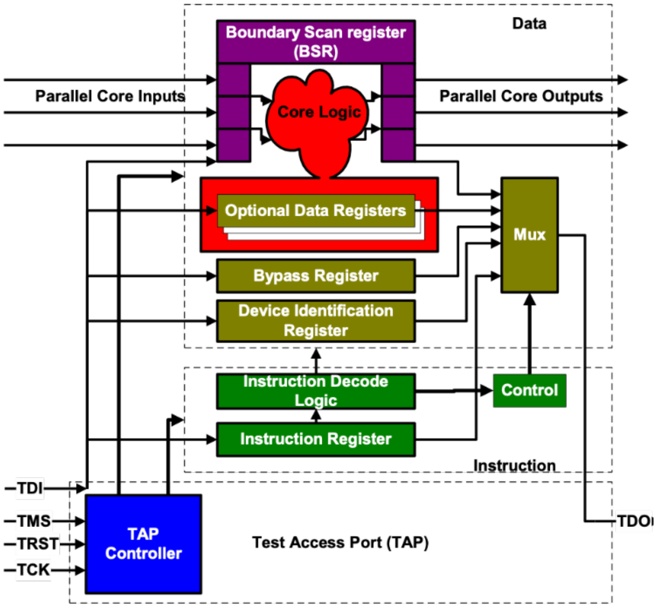

Figure 3. TAP controller with mandatory instruction and data registers and boundary scan register

The main contribution of IEEE1149.1 (DC levels originally) can be summarized as follows:

- A standard 5-pin (four mandatory and one optional pin) interface – Reducing and standardizing the number of pins to access during the test helped mitigate difficulties making contacts to the IC pins. The pins are TDI (Test Data In), TDO (Test Data Out), TCK (Test Clock), TMS (Test Mode Select) and the optional TRSTN (Test Reset In active Low).
- Boundary Scan – Boundary scan chain (BSCAN) isolates IC inputs and outputs to test board-level interconnects non-intrusively with respect to the IC operation. Boundary scan cells are inserted as shown in Figure 3 for all IOs that are configured and controlled by a TAP controller. Depending on the IOs used by the IC and test coverage needed, several variants of boundary scan cells are supported by the

standard including structures to test AC-type at-speed defects (AC-JTAG IEEE1149.6 a later extension to IEE1149.1).

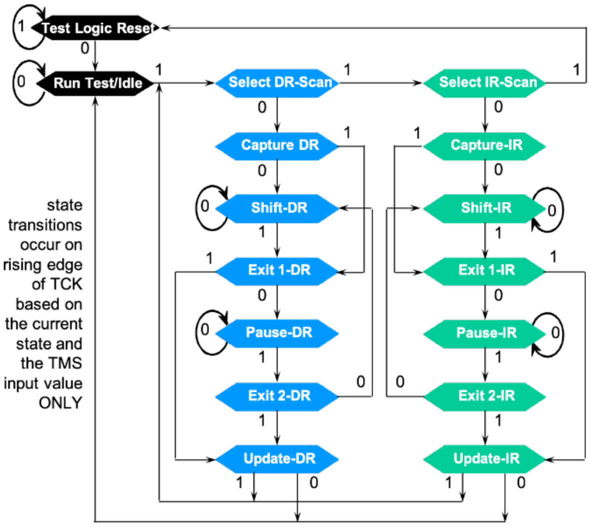

Figure 4. TAP State machine

- The TAP controller – The TAP controller is an on-chip 16 state, finite-state machine, that configures the boundary scan chain as shown in Figure 4. It enables setting up and applying the test program to the IC as well as board-level interconnect tests. The TAP consists of Instruction Registers (IR) and Data Registers (DR). It puts the IC in various test modes to perform SAMPLE/PRELOAD/EXTEST tests. These test modes allow the test program to load stimulus at the output pins, sample responses at the input pins and shift-out the captured value. The BYPASS mode enables skipping an IC from testing while other ICs are tested. It also provides mechanisms to be able to directly identify each IC by way of an IDCODE register. Within this IDCODE register one can also embed the JEDEC Manufacturing ID for further device identification.

- Boundary Scan Description Language (BSDL) File – A file that describes all associated IEEE1149 standard logic such as the list and order of the boundary scan cells, all supported test modes etc. A standardized BSDL file enables EDA vendors to develop test generation tools for board-level interconnects without having to worry about the specific details of the IC or the pin functions of the IC

As mentioned above, the TAP controller is like the central processing unit that executes the test program. The TAP FSM controls the read/write operations into instruction registers (IR) and data registers (DR). Executing an instruction involves shifting in an IR code that selects a DR register and reading/writing values into them. Shifting instruction and data register values are done in a serial fashion using the TDI->TDO ports. The bits in a data register would provide control to set up the test-mode or configure an embedded test instrument. The standard does not restrict how designers could expand the TAP instruction set to support IC-specific test requirements. Designers could extend the instruction set to support new test modes like LBIST, memory BIST and test instruments configuration that are custom to specific ICs. Indeed, in today's very large designs, JTAG DRs are extensively used to override clock/reset/power control bits during test, debug and IC characterizations, as well as implement scan dump and various debug features. One can see that JTAG has fast evolved from simple board level tests, to cover device setup as well.

As designs started growing larger with new IPs, the requirement for register configuration to support customized test modes grew. To support setting up these increased number of test modes, JTAG TAP controllers were expanded with additional private instructions and data registers. However, as we entered the *System-on-Chip* (SoC) era, designs were becoming an integration of many IPs and blocks from many teams and vendors. The IEEE1149 standard could no longer provide a scalable solution for test access. Those high levels of integration came with new challenges that needed plug-and-play and reusable access mechanisms. This led to the development of IEEE1500 that is described next.

B. IEEE1500 – Standardizing Embedded Core Testing [13]

A large system-on-chip comprises many IPs from many teams and sometimes different companies. As mentioned earlier, with increased levels of IP integration came new challenges to provide test access mechanisms for all of the integrated IPs. The IEEE1149 standard was not designed to provide a scalable solution for internal chip test setup. Using a single global TAP controller to meet all the test setup needs for a large SoC has many challenges. As the number of DR registers needed per IP increased, the global routing of wires to/from TAP controllers to various IPs became a problem. The bigger problem with IP integration is the test interface between the SoC and the IP. During the integration process it is important to make sure the test access interface is standardized so test-related changes at the IP level don't trigger design changes during integration. This is even more critical if vendors from different companies are delivering the IP to the SoC team. A standardized test interface translates into a plug-and-play test solution that will help reduce the overall integration turn-around time. This was the main motivation to develop the IEEE1500 standard. Analogous to IEEE1149 TAP controller, IEEE1500 introduced a wrapper TAP (WTAP) controller and a standardized 8-pin interface.

Another main contribution of IEEE1500 is hierarchical testing and IP test reuse. By fencing off the IP core functional interface using a boundary register chain, a test vector can be developed that targets the embedded core without worrying about the functional interface that goes to/from the IP. The IEEE1500 standard introduced a wrapper boundary register (WBR) that wraps the core-under-test analogous to the boundary scan register of the IEEE1149 standard. This wrapper makes sure that the IP core can be tested independently and by itself without worrying about the other functional signals that are going in/out of the IP.

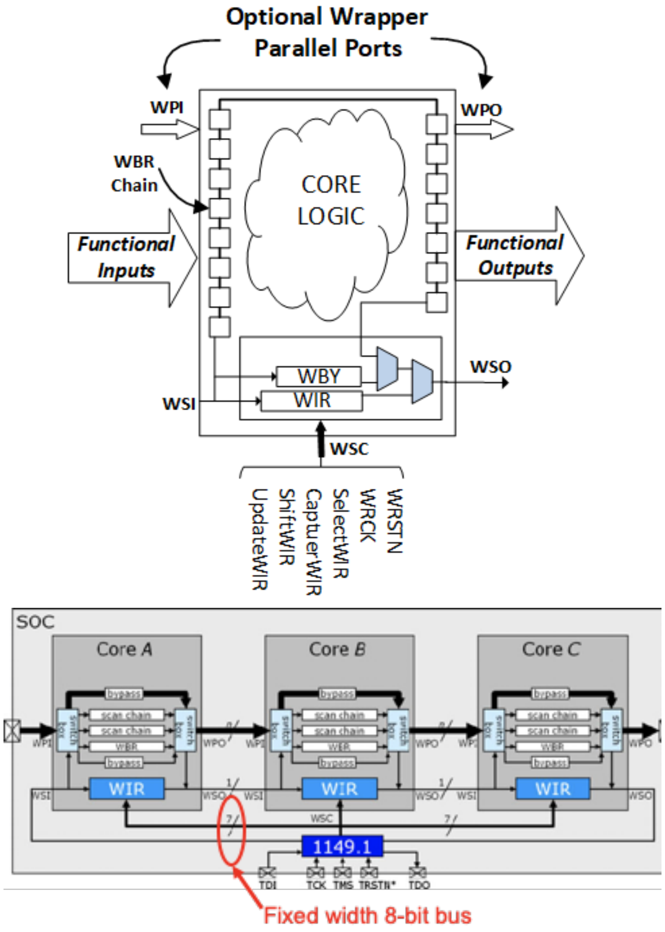

Figure 5. A IEEE1500 compliant embedded core and their integration into SoC [13]

The main motivation for IEEE1500 is standardizing embedded core testing and test reuse whereas IEEE1149 compliance focuses on board level integration and testing. It is possible to use IEEE1149 TAP controllers on each IP for the purpose of embedded core test as described by Vermeulen et. al. [10]. This multi-TAP architecture is popular among several industrial designs although it is not recommended due to the complexities involved in handling IEEE1149 compliance mode. As mentioned previously it also has a variable amount of wires to/from the TAP that are uncountable until final implementation. It is also expensive in terms of area as each embedded IP not only requires implementing the JTAG FSM as well as the mandatory instruction registers required for board level testing compliance. IEEE1500, for each IP, does not have an embedded FSM.

The main contribution of IEEE1500 is summarized below

- Wrapper TAP – WTAP is the main controller block – one per each embedded core. Wrapper Tap controller is analogous to the IEEE1149 TAP without the overhead of implementing the full finite state machine. It has the Wrapper Bypass register WBY that implements core bypass. The port interface for WTAP is called WSP (wrapper serial ports). The WSP port interface is standardized so an IEEE1500 compliant IP enables plug-and-play IP integration as shown in Figure 5. The WSP ports are: *WSI* (similar to TDI), *WSO* (similar to TDO), *WRCK* (similar to TCK), *WRSTN* (similar to TRSTN), *SelectWIR*, *CaptureWR*, *ShiftWR*, and *UpdateWR* (select, capture, shift and update respectively analogous to the IEEE1149 states in the FSM). The wrapper TAP operation is analogous to IEEE1149 TAP in the sense that it houses the WIR - the wrapper IR register analogous to the IR instruction register of IEEE1149. Like the TAP controller, the WTAP generates the control signals that selects a specific WDR (wrapper DR) register. The standard does not mandate how the WSP ports are to be generated at the SoC level but it is implicitly assumed that a SoC level TAP controller could map one or more DR registers to an embedded WTAP.
- WBR – Wrapper Boundary register is analogous to a BSCAN chain of IEEE1149 except that the WBR wraps the IP core boundary. The standard is flexible in how and where these wrappers are instantiated. WBRs are usually overlapped with existing functional registers inside the IP cores. Since most of the IP inputs and output ports are registered so the WBR does not necessarily add any additional area overhead.
- CTL – Core Test Language is the software Description of hardware associated with IEEE1500 and is analogous to IEEE1149 BSDL file.

IEEE1500 solved much of the scalability and reuse issues in testing embedded cores. With IEEE1500 compliant cores, engineers can generate vectors and reuse them wherever the IP is reused. However, in today's large and complex SoCs, the test registers get reused by different hardware teams, at different levels of an SoC design and silicon productization. There was a need to better manage the growing number of test data registers in IPs. Next, we will describe how the IEEE1687 iJTAG standard further improves upon IEEE1500 to help reuse core-level vectors and retarget the vectors at different hierarchical levels of the design. This method, called pattern retargeting, helps reduce turn-around time for chip-level vector generation by reusing IP-level vectors.

C. *IEEE1687 – Internal JTAG (iJTAG) Standard [24]*

The size and overall usage of JTAG test data registers grew significantly as the complexity of the chips grew. Today's SoCs have many clock elements, power structures, complex on-chip reset generation hardware, memories, custom analog macros, along with on-chip sensors to monitor temperature and process shift that require specific settings in various test/debug modes of operation. Programming these structures using JTAG during design verification, test mode, silicon debug, and silicon characterization modes required a tremendous effort from teams spanning across design, verification, test and product engineering. A flat set of JTAG test data registers, each organized as a fixed length chain is not scalable anymore. For ease of managing these JTAG data registers, there is a need for a reconfigurable JTAG register network that could be organized into many hierarchical levels. To understand this, consider an IP that has a block, say, *A* that requires 3 different 8-bit registers, say *RA1[7:0]*, *RA2[7:0]*, and *RA3[7:0]*. If we use conventional IEEE1149 or IEEE1500, we will need to have three separate instruction registers or one long register to implement these DR bits. A better way to organize will be using a reconfigurable JTAG register as shown in Figure 6. Here the bit *SRA1* is a configurable register bit called *Segment Insertion Bit* (SIB). A SIB can *open-up* or reconfigure itself into a JTAG 8-bit register that holds the value RA1[7:0]. Such a hierarchical management of DR registers that locally reconfigures registers helps in limiting the number of instructions and size of the DRs. This helps reduce overall test time and the number of global wires needed to route to these registers. A local SIB bit controls if we need to bypass a segment of JTAG register. From that perspective, IEEE1687 standard does not introduce any new infrastructure such as a controller block like TAP or WTAP. The SIB bit can be inserted in the IR chain or DR chain that can reconfigure the JTAG network to add or remove JTAG register segments as needed.

Another major contribution of IEEE1687 is the standardized software models for describing the reconfigurable register connection and the initialization sequence. The IEEE1687 hardware description language is called *Interconnect Connectivity Language* (ICL). The language allows designers to label various JTAG registers with alias names and describe their connectivity to/from other blocks in the design. This can facilitate EDA tools to read macro-level ICL files and, via introspection, create core-level / chip-level ICL based on the connectivity information represented. A macro can represent a small block, say an embedded test instrument module, within the core-under test. This facilitates software tools to automate the generation of test initialization sequences as we retarget vectors at different levels of design hierarchy allowing various teams to document and reuse initialization sequences. The representation of JTAG vector sequences are also standardized using a TCL-like language introduced as part of IEEE1687. This is called the *Procedure Description Language* (PDL). Verification/test engineers can develop TCL procedures with JTAG read/write sequences for various programming needs at the macro-level, embedded IP level, or at the chip-level. Below we summarize the main contribution of IEEE1687.

Segment Insertion Bit (SIB): As explained above, SIB bits reconfigure the JTAG register network by adding/bypassing JTAG chain segments. Figure 6 shows the SIB bit that, when enabled, allows the JTAG register corresponding to instruments RA1, RA2, and RA3 to be added as part of the JTAG chain.

JTAG Broadcast Write: Figure 6 also shows a JTAG broadcast write. This is useful when there are identical replicated cores in the design that require the same initialization sequence. Being able to broadcast the same JTAG data to multiple registers at once helps reduce test time instead of having to shift them serially. In the figure when the broadcast bit is set to 1, the same test data is broadcasted to all the three test instruments *A*, *B*, and *C*.

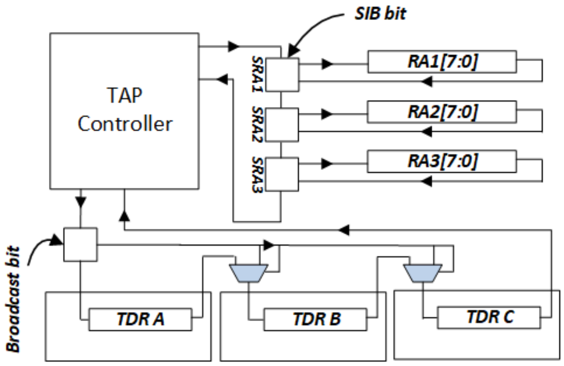

Figure 6. Example design IEEE1687 compliant SIB bit and broadcast bit

ICL/PDL: As mentioned earlier, ICL models the JTAG network connectivity of the design. The description can be done for small IP/macros, or for an entire chip. In a large SoC design environment, an ICL description for various libraries, clock modules, as well as chip-level JTAG block is usually delivered by the designer. An EDA tool is then used to *extract* the SoC-level ICL using the lower level ICL files. A separate PDL procedure file describes the list of modes relevant for the IP/block, along with their programming sequence using the PDL write/read APIs. The register alias names in ICL can be referenced in PDL to simplify read/write programming values into the registers. EDA vendors can develop tools that read ICL and PDL to construct the programming sequences at various levels of hierarchy. The main advantage of a standardized PDL sequence are: a) EDA tools across disciplines can now support reading the sequence for their application, and b) lower level hardware changes such as changes to register bit position do not trigger chip-level sequence changes as the PDL sequences will still remain valid and reusable.

Overall, IEEE1687 standardization helps in making the JTAG network flexible, reconfigurable, and making the programming sequence description more portable so different teams can reuse the sequence. The ever-increasing requirements to integrate more features have caused die sizes to hit the upper reticle limits of what foundries can manufacture with adequate yield. The increased turnaround time to shrink the process technology node makes it harder to reap the benefits of Moore’s law using conventional 2D dies. With advancement in process technologies for micro-bumps and through-silicon-vias (TSVs), 2.5D and 3D die stacking is enabling us to continue benefiting from higher levels of integration. From a test perspective, existing standards such as IEEE1149, IEEE1500, and IEEE1687 do not adequately address the test access challenges. This is especially so when considering stacked dies that have limited die level access ports and very large numbers of die-to-die interface connections. Recently, the IEEE1838 standard was developed to address the challenges specific to 2.5/3D device test access mechanism. This is described next.

D. IEEE1838 – Standardizing Testing of 2.5D and 3D stacked devices

A 2.5D die integration project uses a silicon interposer to integrate multiple dies together. In this paper, the term chiplets will be used to refer to these dies. The interposers have TSVs that in turn are connected to a limited number of C4 bumps. The die-to-die interconnection is typically realized using passive wires running across the interposers. Connections between the die and the interposer are fulfilled using micro-bumps that are typically ~10µm diameter and ~40 µm pitch (currently). Due to its smaller footprint, low power, low drive strength requirements a die can economically support several thousands of micro-bumps compared to ~1K regular bumps. In a 3D integration using TSVs, the vias run through one or more die. They pose similar challenges in providing test access for the stacked die [14, 15].

IEEE1838 [16] provides a standardized test port interface across one or more 2.5D/3D stacked devices. A standardized test port interface will enable plug-and-play integration providing test access across various dies. The main contributions of IEEE1838 are summarized below.

Primary TAP (PTAP) and secondary TAP (STAP)

A PTAP controller is based on an IEEE1149 TAP controller as shown in Figure 7 above. It has the standard 5-pin interface and supports the IEEE1149 instructions such as IR, BYPASS, IDCODE. It introduces a new 3D configuration instruction and an associated register called *Select3DCR* instruction and *3DCR* register that configures test data transport from one die to another. An STAP controller is a secondary TAP that provides a feedthrough / pipelined signal interface which can be routed to the PTAP block residing in an adjacent die. Each die is expected to have one PTAP and any number of STAPs that are necessary to transport the test data to the adjacent die. Note that an STAP structure may not be needed for the die without any adjacent dies to transport the test data. This would be the case for the top die in a stack or the edge die in a 2.5D interposer. For those dies, an STAP is optional. Since the PTAP to STAP signal routing can be long, potentially spanning from one side of the die to another side, the standard supports pipelining the data path. This is a deviation from the IEEE1149 standard as data path pipelining is not supported.

Primary and Secondary Port Interfaces

The standard introduces two signal interfaces (i/f) – a primary port i/f and a secondary port i/f. The TAP 5-pin interface *TDO*, *TDI*, *TMS*, *TRSTN*, and *TCK* at the PTAP-side is referred to as the primary port i/f. The 5-pin port at the STAP-side is called the secondary port i/f that is driven by the PTAP ports as shown in Figure 7. Within a die, a PTAP can distribute scan data to many STAP blocks and receive the test response back. The register bits of the 3DCR register configures the test data transport amongst the PTAP and STAPs. This is achieved by appropriately gating-off the TMS signal of an STAP block and parking any inactive PTAPs in either the test-logic-reset or run-test-idle states.

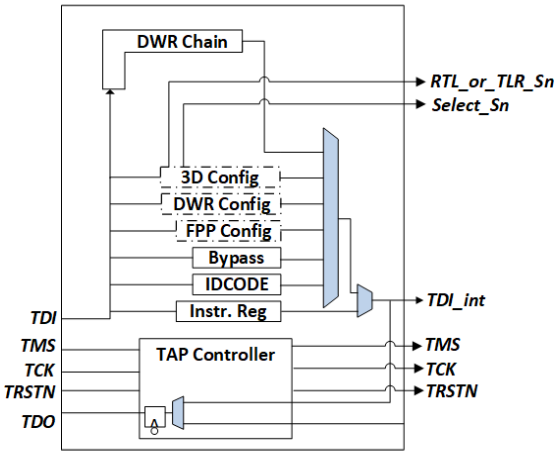

Figure 7. Example IEEE1838 PTAP interface and new IRs/DRs

Die Wrapper Register (DWR)

Analogous to the boundary scan register of IEEE1149 and the WBR register of IEEE1500, IEEE1838 introduces a die wrapper register (DWR). This register isolates the current die under test from other adjacent dies to perform in-test and ex-test modes of testing. The IEEE1838 standard is flexible and allows designers to reuse existing boundary scan cells or wrapper boundary cells to treat them as DWR cells during the die-to-die interface testing.

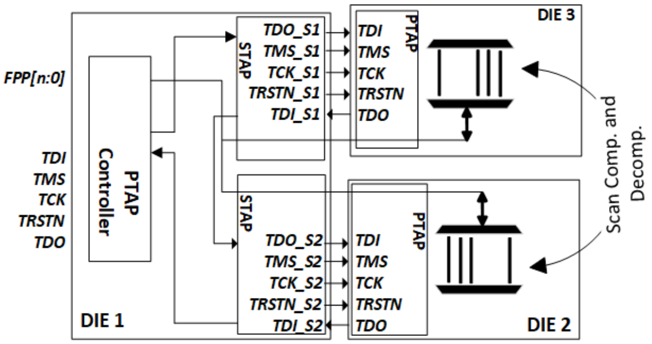

Figure 8. Interposer with Three IEEE1838 Compliant Chiplets

Flexible Parallel Port (FPP)

The standard introduced a user-defined parallel port interface that can be used to transport test data [17], for e.g. scan data, across dies and carry test response back to the main die. Figure 8 below shows three dies where Die #1 has access to C4 bumps. The flexible parallel ports are used to deliver scan data to Die #2 and Die #3.

IEEE1838 defines a standardized set of test interface ports, for dies, to deliver test data to be transported across dies. When chiplets from different vendors co-exist in an interposer, there are many challenges. These challenges are related to IC bring-up, debug, as well as opportunities to reduce test cost and improve overall time to market.

IV. Marketplace Test Challenges

The ODSA aims to create an open interface for chiplets and a marketplace for chiplets that support the open interface. It is clear, from the previous sections, that industry and researchers have put substantial effort into developing standards and practices to test chiplet-based products. An open marketplace with open interfaces offers both new opportunities for testing and presents challenges that require more effort.

- The biggest opportunity with an open approach is economies of scale. Chiplet vendors can make products that now serve designs in a wide range of applications. Equipment vendors can produce systems, including those for testing chiplets, that serve a wide range of designs.
- The biggest challenge is the one experienced with any open approach, both will require far greater interoperability. Interoperability in test hardware and software between the chiplets in a design. Interoperability in test information between chiplet vendors and chiplet consumers.

The objective in an open marketplace is to enable product designers to easily integrate chiplets from multiple vendors manufactured across different process nodes. As with the design, product designers will need to integrate the components across various chiplets. Industry will need to build on and extend current efforts to enable test integration.

For example, HBM flows depend on the ability to repair memory die which leads to high product yields on HBM stacks. This is not an option for most logic die. In most logic-based mult-chiplet designs, redundancy is not a practical or even viable approach. Therefore, it is even more important to provide KGD to enable cost-effective product assembly. To achieve a product yield of 95%, one can calculate that KGD yield may be required to be 98% per chiplet for a multi-die package that contains more than five chiplets. Logic-die based chiplet products will require more attention to producing known good die. To make producing known good die more economical, we explore two approaches. Similarly, current standards for scan need to be extended to enable cooperative tests across chiplets from different vendors in an assembled product.

For each issue, we identify the unique aspects of an open system and how current approaches can be extended to address them. The open interface and marketplace problems we discuss are:

1. Known good die requirements with open test information exchange. This will enable chiplet vendors to directly meet the quality needs of specific use cases.
2. Chiplet test interoperation requirements to test the functionality and performance of interconnect, final packaging, assembly and bring-up.
3. Lowering wafer-level and die-level test costs by leveraging economies of scale for test probe design costs for probe cards for a new open interface.

A. Test Information Exchange for Shift Left

To support the “Shift Left” model, in a marketplace model for chiplets, the entity developing a chiplet will likely not be the entity building a product from it. One advantage of the marketplace is economies of scale by using a chiplet in a diverse range of applications, hence expanding its scope and lifetime. Each new product using a chiplet may exercise functionality not previously triggered. The KGD requirements for the same chiplet will vary with the product in which it is used. A vendor may need to offer the same chiplet at multiple quality levels, each of which targets a different end-product reliability target. We envision that information will need to flow “backwards” in the workflow from the product developer to the chiplet vendor. To support these needs, a marketplace may have to enable a common feedback mechanism from the product developers to the chiplet vendors. This opens an interesting conversation regarding anonymity, liabilities and penalties.

For example, if a large percentage of the failures can be attributed or diagnosed to a single die, then that specific package integrator may want to recover the cost of scrap material. That is, the package integrator may want the chiplet vendor to further screen die for a specific use case. In our previous example, let’s say that the end-product requires one thousand packaged parts. This will result in a product cost increase of $32,780. As one can see the final integrator would be motivated to isolate the fault to a device that they are purchasing to be able to transfer the cost of the reduction in profitability. In an ideal world the assembler would provide the failure diagnosis back to the die manufacturer, so that they could provide additional testing. In our reference case the fault could be diagnosed to a single test that could take 1 second of additional test time. The die manufacturer could then compare the additional test time cost to the penalty cost.

A cost modeling of each test step will enable the tradeoff analysis. This should include the cost of Automated Test Equipment, Handler, Prober, Interface consumables and a calculation of test time per second. A good discussion of this does exist in the HIR roadmap. The analysis should be done per device assembly because the optimizations can be very different depending on the end market. In the “Shift Left” model, the marketplace needs a structure to transfer information from the die supplier to the system integrator. The marketplace will need incentives for a chiplet vendor to supply additional coverage for specific uses. Additionally, the marketplace would need incentives for a chiplet vendor to be able to sell lower quality die for a lower price. This would help drive the chiplet vendors profitability and desire to enter the marketplace. Conversely, a die vendor’s competitive advantage must be protected when they transfer the information.

The goal of the feedback loops is to provide diagnostic information to enable a lower cost of test and moving the test ahead in the workflow. Figure 9 shows a distributed database envisioned by the IEEE HIR. This database now needs to be distributed across chiplet vendors and product developers. Conversely chiplet vendors will need to be able to respond to specific yield requirements from various final products’ characterization, Engineering Verification (EVT) and Product Verification (PVT). Experience working on complex programs suggest that achieving 90+% yield requires early EVT and PVT. Scheduling early verification and investing significant time, resources, and effort at component, module, subsystem and system levels is crucial. By building test vectors for each stage and providing test insertion stages and checks for major manufacturing steps helps improve the overall time-to-market. To that end, die vendors and developers will need to exchange confidential test data around testing and yield while preserving business-confidential information. These trade-offs are typically conducted within a single vendor’s departments for monolithic products, making the conversations and ability to do so easier. By shifting these investments to the left, the yield will increase and stabilize quickly with low field returns in the end, however the conversations and ability to make this happen become harder

B. *Cross-Chiplet Interoperation for Interface and Assembly Test*

Interoperation among chiplets is needed to test the entire product including the interface functionality and performance. IEEE1838 and prior standards have addressed the minimum requirements to standardize test application, test register connectivity network, and processes that make well known classes of tests possible. However, it does not address complete test functionality that enables product debug and bring-up. Since test hardware is reused for characterization, debug, and diagnosis, it is important to consider interoperability requirements of testing from the perspective of product bring-up and debug of RMA/customer returns. Current test access standards are enough for single die products, or for a muti-die from one company product because of intra-company conventions. In a multi-vendor chiplet product, these best practices also must be added to the standards.

We discuss specific problems that require interoperation between chiplets in a marketplace to be addressed effectively:

Interface test: standards have been developed to test 3rd party IP, such as a PCIe interface IP integrated to a monolithic product. The technology industry needs to develop minimum standards to build a product with components from multiple companies. The industry needs to specify an interoperation standard that chiplets can work together to achieve that minimum quality level and interoperation. To test the functionality and performance of the D2D interconnect between devices, tests will have to be sourced in one device and collected, collated or analyzed in a second device. Several researchers have suggested the use of PRBS/MISR for performance characterization for the interconnect. This will require the BIST engine on one chiplet to serve as a PRBS and be coordinated with the BIST engine on another chiplet serving as a MISR through IEEE1838.A standard control bus to manage the interoperation between devices will need to be defined and ratified. This test structure will become critical in getting test coverage as well as providing diagnosis feedback during product debug. Another way to mitigate operation of a D2D interface is adding calibration circuitry for performance of the interface and monitoring Bit Error Rate (BER) w/retry logic. Both techniques will enable yield harvesting and power/performance optimization. The other advantage to adding calibration circuitry on at least one die, is that differences in timing and levels could be compensated for at the package level. D2D interface initialization and calibration now requires interoperation between test structures in the chiplets on either side of the interface which can be achieved by extending current IEEE standards.

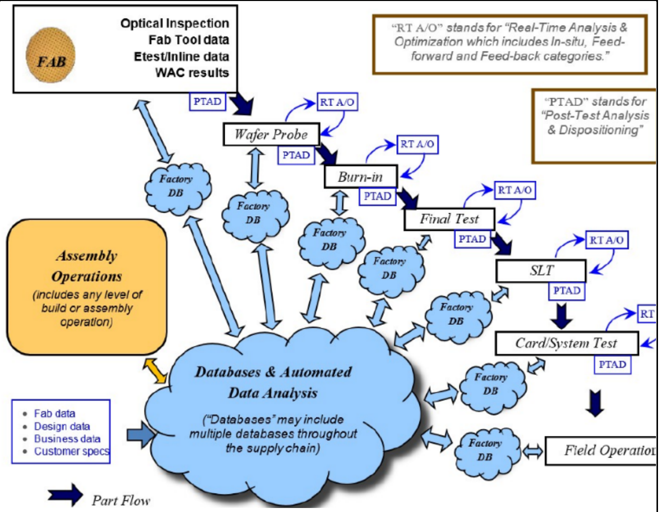
FAB
Optical Inspection
Fab Tool data
Etest/Inline data
WAC results
“RT A/O” stands for “Real-Time Analysis & Optimization which includes In-situ, Feed-forward and Feed-back categories.”
“PTAD” stands for “Post-Test Analysis & Dispositioning”
PTAD
Wafer Probe
RT A/O
PTAD
Burn-in
RT A/O
PTAD
Final Test
RT A/O
PTAD
SLT
RT A/O
PTAD
Card/System Test
RT
PTAD
Field Operatio
Factory DB
Factory DB
Factory DB
Factory DB
Factory DB
Factory DB
Factory DB
Assembly Operations
(includes any level of build or assembly operation)
Databases & Automated Data Analysis
(“Databases” may include multiple databases throughout the supply chain)
Fab data
Design data
Business data
Customer specs
Part Flow

Figure 9. Information feedback from product to chiplet (from the IEEE HIR)

System-Level Test (SLT) and ATE Correlation: An SLT test usually involves loading a set of software programs into the processor memory and executing them sequentially. The SLT board environment is usually different (e.g. power supply source) compared to the ATE environment and may mimic one particular use case for the final product. ATE tests also have more diagnosis resolution than SLT. Therefore, to debug a product with a failing system-level test, it is important to be able to go back and apply an ATE test to the individual chiplets to debug/root-cause the failing chiplet. The main challenge here is to make sure the test ports are fully accessible during system level test-debug and ATE tests can be applied to the chiplet under debug. Not only do the test ports have to be accessible, the ability to take the ATE test from that specific environment and apply it to the chiplet in an SLT environment. This requires cohesion between these environments and software portability. Product debug: JTAG is often used in product bring up and debug to access internal registers, including debug registers not normally part of the user visible state. The industry will need to develop a format to exchange and access internal register states that can be "dumped" to root-cause failures. This format will need to preserve device proprietary information, as well adhere to device security restrictions

C. *A Bunch of Wires (BoW) D2D Interface Probe Card Design*

High yields for KGD are achievable if more tests can be applied to die in wafer form. KGD test requires full probe card access to the die to test the pins, for test data i/o and to trigger test functions on the die. Probe card design is easier at wafer test for a die with regular-sized bumps at 100um or greater spacing and with circuitry for long range PHYs. Die designed to be used as chiplets will have PHYs with limited drive strength and when micropumps are deployed, constricted spacing between bumps as shown in Figure 10. Probe cards for chiplets with micro bumps will need to be physically closer to the chiplet to enable efficient contact during testing and to account for limited drive strength. They may also require amplifiers for test i/o and likely be more expensive to design.

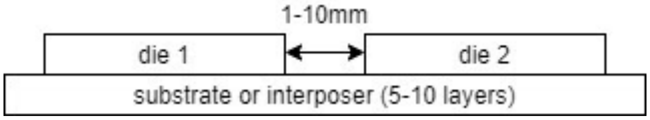

Figure 10. Packaged environment

The design for probe cards for logic tests can build on the designs for HBM. In the HBM workflow, a lot of the testing normally conducted on packaged die, including burn-in, are moved to die in the wafer. HBM workflows aim to reduce costs with probe cards that simultaneously test multiple die in a wafer. The partition of the interface into channels and the availability of a standard bump map in the specification simplifies probe card design. The HBM industry has developed several approaches to delivering test data i/o and power. These approaches also include ways to mitigate the impact of physical contact, using sacrificial bump pads and non-contact technology.

We expect that economies of scale can potentially help with managing probe card costs in chiplet-based products and propose that HBM approaches be extended to open D2D interfaces. Including those being developed by the ODSA, there are several open D2D PHY interfaces under development. The Bunch of Wires is a new open interface under development by the ODSA. The interface aims to be an easy-to-design parallel interface usable across a wide range of process nodes and packaging technologies.

To enable the growth of chiplets with the BoW interface [18], the ODSA aims to develop test chips for the interface. As mentioned before D2D interfaces present new challenges for wafer-level and die test for chiplets. The limited drive strength in D2D interfaces implies tester probes will have to amplify received signals. Microbump density increases design and alignment complexity. Figure 11 shows an expected design for a tester probe for the BoW interface.

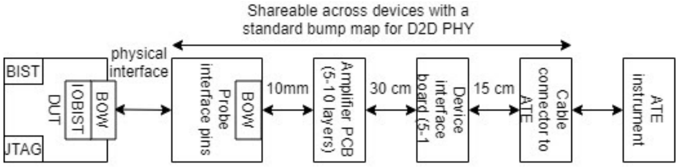
Shareable across devices with a standard bump map for D2D PHY
physical interface
BIST
JTAG
DUT
IOBIST
BOW
Probe interface pins
BOW
10mm
Amplifier PCB (5-10 layers)
30 cm
Device interface board (5-1
15 cm
Cable connector to ATE
ATE instrument

Figure 11. Potential BoW probe card design

A probe design unique to each chiplet will potentially make achieving the high KGD yields required prohibitively expensive. An open interface offers the potential for economies of scale, lowering the cost of test and making high KGD yields more economical.

Through the appropriate standardization, as shown in the Figure 11, it may be possible to reuse a probe head design across a wide range of chiplets. The tester probe will need a bump layout that is compatible with multiple chiplets. The first step to make this possible is the continued development of reference bump maps for the BoW interface. The community will also need to define an appropriate reference physical interface methodology. In particular, microbump BoW interfaces may require sacrificial regular-size bumps as in HBM interfaces or sacrificial microbumps.

Last, a test standardization effort for the BoW interface will have to specify how access to the device and test infrastructure can be accessed and triggered through the BoW interface. The standard will need to specify the electrical environment for receiving low-drive signals from the DUT and the ESD requirements. The integration with the IEEE 1149/IEEE1838 structures on die will have to specify the data delivery mechanism and the delivery of common functional and performance tests such as stuck-at, functional and IO-BIST and loopback tests [19, 20]. The definition will have to align with the potential use of test structures for initialization and calibration of the BoW interfaces.

Concurrent with the development of standards, we propose the development of an open probe design for BoW interfaces that supports these test standards to align with the development of BoW test chiplets. It is likely that it may be easier to stage the design for probe cards, with a first targeting regular bump BoW interfaces. We believe that by addressing these issues in the BoW POC we will enable the D2D interfaces with a testing method which could be adopted by industry to ensure chiplet and system quality. The desire is that teams engage with the ODSA to address issues identified in the previous paragraphs.

V. Conclusion

To enable the chiplet marketplace we must make it economically viable to assemble heterogeneous packages from multiple chiplet vendors. Production tests, along with integrated, universal and cohesive DFT strategies are an important component in realizing this goal. In this paper we discussed the concepts of known good die and moving coverage between test insertions. The notion of Shifting Left this coverage per-se. We have given insight into the economics of test and coverage to help the reader balance the tradeoffs for different products. To achieve this coverage we have also provided background on existing DFT standards which enable both the test fabric and the test description language. Depending on the cost of each component and the cost of implementation of coverage, the test intensity at each stage may be different for different products

Acknowledgment

We like to thanks Karthik Natarajan (Synopsys), Phil Nigh (Broadcom), Bill Chen/Calvin Cheung (ASE) for feedbacks. We also like to thank Lisa Jensen (Cadence), Yervant Zorian (Synopsys), Eelco Bergman (ASE), Mengzi Peng (Tesla), Sam Fuller (NXP) for content in ODSA test and packaging panels.

References

[1] IEEE Heterogeneous Integration Roadmap, https://eps.ieee.org/technology/heterogeneous-integration-roadmap/2019-edition.html

[2] The Open Domain-Specific Architecture, https://www.opencompute.org/wiki/Server/ODSA.

[3] E. J. Marinissen, F. Fodor, A. Podpod, M. Stucchi, Y. Jian and C. Wu, "Solutions to Multiple Probing Challenges for Test Access to Multi-Die Stacked Integrated Circuits," 2018 IEEE International Test Conference (ITC), Phoenix, AZ, USA, 2018, pp. 1-10.

[4] E. J. Marinissen, "Challenges and emerging solutions in testing TSV-based 2 1 over 2D- and 3D-stacked ICs," 2012 Design, Automation & Test in Europe Conference & Exhibition (DATE), Dresden, 2012, pp. 1277-1282

[5] C. Papameletis, B. Keller, V. Chickermane, S. Hamdioui and E. J. Marinissen, "A DfT Architecture and Tool Flow for 3-D SICs With Test Data Compression, Embedded Cores, and Multiple Towers," in IEEE Design & Test, vol. 32, no. 4, pp. 40-48, Aug. 2015.

[6] A. Frisch et al., "Supplying known good die for MCM applications using low cost embedded testing," Proceedings of 1995 IEEE International Test Conference (ITC), Washington, DC, USA, 1995, pp. 328-335.

[7] H.K. Charles Jr., “Tradeoff in multi-chip module yield and cost with known good die probability and repair.”, Microelectronics Reliability, Vol. 41, from 2001, pp. 715-733.

[8] D. Armstrong and G. Maier, "Known-good-die test methods for large, thin, high-power digital devices," 2016 IEEE International Test Conference (ITC), Fort Worth, TX, 2016, pp. 1-6.

[9] Per Torstein Roine, “An Asynchronous PRBS Error Checker for Testing High-speed Self-Clocked Serial Links”, Proceedings Fourth International Symposium on advanced research in asynchronous circuits and systems, 1998, pp133-141

[10] B. Vermeulen, T. Waayers and S. Bakker, "IEEE 1149.1-compliant access architecture for multiple core debug on digital system chips," Proceedings. International Test Conference, Baltimore, MD, USA, 2002, pp. 55-63.

[11] IEEE Standard for Test Access Port and Boundary-Scan Architecture," in IEEE Std 1149.1-2013 (Revision of IEEE Std 1149.1-2001) - vol., no., pp.1-899, 13 May 2013

[12] Kenneth P. Parker. 2014. The Boundary-Scan Handbook (4th. ed.). Springer-Verlag, Berlin, Heidelberg.

[13] IEEE Standard Testability Method for Embedded Core-based Integrated Circuits," in IEEE Std 1500-2005 , vol., no., pp.1-136, 29 Aug. 2005

[14] R. Wang and K. Chakrabarty, "Tackling Test Challenges for Interposer Based 2.5-D Integrated Circuits," in IEEE Design & Test, vol. 34, no. 5, pp. 72-79, Oct. 2017

[15] R. Wang, G. Li, R. Li, J. Qian and K. Chakrabarty, "ExTest Scheduling and Optimization for 2.5-D SoCs With Wrapped Tiles," in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 36, no. 6, pp. 1030-1042, June 2017

[16] IEEE Standard for Test Access Architecture for Three-Dimensional Stacked Integrated Circuits," in IEEE Std 1838-2019, vol., no., pp.1-73, 13 March 2020

[17] Y. Li, 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," 2018 IEEE 23rd European Test Symposium (ETS), Bremen, 2018, pp. 1-6

[18] R. Farjadrad, M. Kuemerle, B. Vinnakota, “A Bunch-of-Wires (BoW) Interface for Interchiplet Communication”, IEEE Micro, Jan. 2020

[19] R. Sethuram, K. Arabi, and S. Kasarla, "Testing Stacked Die", Patent Publication Number 20130043897 Feb. 2013

[20] A. L. S. Loke et al., "Loopback architecture for wafer-level at-speed testing of embedded HyperTransportTM processor links," 2009 IEEE Custom Integrated Circuits Conference, Rome, 2009, pp. 605-608

[21] S. Deutsch et al., "DfT architecture and ATPG for Interconnect tests of JEDEC Wide-I/O memory-on-logic die stacks," 2012 IEEE International Test Conference, Anaheim, CA, 2012, pp. 1-10.

[22] H. Jun, S. Nam, H. Jin, J. Lee, Y. J. Park and J. J. Lee, "High-Bandwidth Memory (HBM) Test Challenges and Solutions," in IEEE Design & Test, vol. 34, no. 1, pp. 16-25, Feb. 2017.

[23] A. Liao, “Verification of Singulated HBM2 Stacks with Die Level Handler, and Review of Wafer Level Sort Challenges”, Form Factor Users Group, Taiwan, 2019.

[24] IEEE Standard for Access and Control of Instrumentation Embedded within a Semiconductor Device," in IEEE Std 1687-2014 , vol., no., pp.1-283, 5 Dec. 2014