Heterogeneous Integrated Product Testability
Best-Known Methods (BKM)

Revision 1.0

Sponsor
Test Technical Committee
of the
IEEE Electronics Packaging Society (EPS)

Approved 10 December 2021
IEEE EPS Test Technology Committee (TTC)
**Notice and Disclaimer of Liability Concerning the Use of IEEE Documents:** IEEE EPS Test BKM documents are developed within the IEEE Test Technical Committee of the IEEE Electronics Packaging Society. IEEE develops its standards through a consensus development process, approved by the American National Standards Institute, which brings together volunteers representing varied viewpoints and interests to achieve the final product. Volunteers are not necessarily members of the Institute and serve without compensation. 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.

Use of an IEEE BKM is wholly voluntary. IEEE disclaims liability for any personal injury, property or other damage, of any nature whatsoever, whether special, indirect, consequential, or compensatory, directly or indirectly resulting from the publication, use of, or reliance upon any IEEE BKM document. IEEE does not warrant or represent the accuracy or content of the material contained in its standards, and expressly disclaims any express or implied warranty, including any implied warranty of merchantability or fitness for a specific purpose, or that the use of the material contained in its standards is free from patent infringement. IEEE BKM documents are supplied "AS IS."

The existence of an IEEE BKM 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 BKM. Furthermore, the viewpoint expressed at the time a BKM 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 BKM. Every IEEE BKM is subjected to review at least every five years. When a document is more than five 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 BKM.

In publishing and making its BKMs 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 BKM 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.
Table of Contents

Revision History

1. Introduction
   1.1. BKM Document Purpose
   1.2. Audience
   1.3. Inputters
   1.4. Assumptions
   1.5. Scope
   1.6. Terminology
   1.7. Reference Documents and References to IEEE Standards
   1.8. Approval / Deciders

2. Heterogeneous Integrated Products and Testability Overview
   1.9. Summary Abstract
   1.10. Background
   1.11. References

3. Testability Ecosystem and Heterogenous Integrated Test Economics
   3.1. Summary Abstract
   3.2. The Impact of Test and Testability on Heterogenous Integrated Products Partners
   3.3. Testability Gaps Assessment
   3.4. High-level Heterogenous Integrated DFx Strategies and IEEE Standards
   3.5. Heterogenous Integrated Product Unique Test Economic Considerations
   3.6 References:

4. Die and Probe Layout Strategies to Enable Probing Best Practices
   4.1. Summary Abstract
   4.2. Industry’s call for KGB from Probe Test
   4.3. Cost-savings and Compromises in Probe
   4.4. Microbumps and Probe Cards
   4.5. Employing Test Pads Among Microbump Arrays

References:
8.3. ATE Testing In Multichip Devices

8.4. Re-invent HVM ATE Functional Testing

8.5. Software-Driven HIR Test


9.1. Summary Abstract


10. Case Studies

Case Study 1: Multi-Die Packages and Implication to Test from an OSAT Perspective

Case Study 2: Debugging Chiplet Failures for Passive Interposer Topology

Case Study 3: Dealing with Field Returns with a product using an Active Interposer

Case Study 4: Package Level TSV or IO Failures

Case Study 5: Die to Die IO Fault Model and Diagnosis Capability

Case Study 6: Direct Micro-bump Probing - HBM2 KGD Test

Table of Figures

Figure 1: High-end packaging roadmap: application-technology

Figure 2: Mapping of high-end packaging players based on technology categories

Figure 3: Technology-agnostic Schematic of a disaggregated product and its testability interactions

Figure 4: Bottom of SoC device in the stack provides test pads in the field of microbumps

Figure 5: (a) SEM image of MEMS cantilever spring probes. (b) Al pads are distributed among the microbumps to enable test using conventional MEMS spring probe technology. (c) Detailed side view of the spring tips contacting the Al pads in relation to the microbumps in both scrub and non-scrub directions.

Figure 6: Schematic of a hybrid probe array for application-processor test. I/Os at fine pitch are shown in red; powers/grounds with larger pitch (and current requirements) are shown in blue.

Figure 7: Power impedance (PI) comparison for critical power supply with 17 bumps. Utilization of higher current, 130 µm-pitch probes for PWR/GND allows 40% lower PI than with 80 µm-pitch alone.

Figure 8: Cartoon of a 3D Stacked SoC showing the use of Die-to-Die communication

Figure 9: Clustering in 3 stacked Die

Figure 10: Logical abstraction of the Die-to-Die Interconnect

Figure 11: Victim-Aggressor Tests (VATs) for die-to-die interconnects.

Figure 12: IEEE P1838 3D DFT Standard
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>13</td>
<td>Yield and Quality are the main concerns</td>
</tr>
<tr>
<td>14</td>
<td>2.5D/3D Package Level Test Flow</td>
</tr>
<tr>
<td>15</td>
<td>2.5D Passive Interposer Topology</td>
</tr>
<tr>
<td>16</td>
<td>Assembly Screen Test Flow for 2.5D</td>
</tr>
<tr>
<td>17</td>
<td>How to Distinguish between TSV vs. IO Failure</td>
</tr>
<tr>
<td>18</td>
<td>Chiplet Test Access Mechanism for 2.5D</td>
</tr>
<tr>
<td>19</td>
<td>Initialization and At Speed Interconnect Test Flow for 2.5D</td>
</tr>
<tr>
<td>20</td>
<td>3D Active Interposer Topology</td>
</tr>
<tr>
<td>21</td>
<td>Assembly Screen Test Flow for 3D</td>
</tr>
<tr>
<td>22</td>
<td>Chiplet Test Access Mechanism Test flow for 3D</td>
</tr>
<tr>
<td>23</td>
<td>Initialization and At Speed Interconnect Test Flow for 3D</td>
</tr>
<tr>
<td>24</td>
<td>Debug Hooks for Initialization and Interoperability. Isolation is key.</td>
</tr>
<tr>
<td>25</td>
<td>Need 100% package equivalence for test/debug at bare-die</td>
</tr>
<tr>
<td>26</td>
<td>Debug hooks for test access</td>
</tr>
<tr>
<td>27</td>
<td>ATE testing of Multi-Chip Devices</td>
</tr>
<tr>
<td>28</td>
<td>Scan &amp; S/W Defined Functional Test over HSIO</td>
</tr>
<tr>
<td>29</td>
<td>Functional Test Download &amp; Execute w/HSIO</td>
</tr>
<tr>
<td>30</td>
<td>Extended Binning for Advanced Packaging</td>
</tr>
<tr>
<td>31</td>
<td>Example Package Definition</td>
</tr>
<tr>
<td>32</td>
<td>Higher Bandwidth can be realized with either a wide IO bus or a high speed serial implementation</td>
</tr>
<tr>
<td>33</td>
<td>Example Package Definition</td>
</tr>
<tr>
<td>34</td>
<td>Amkor’s 2021 Heterogeneous Integration Path Options</td>
</tr>
<tr>
<td>35</td>
<td>A fraction of the logic products have emphasized on adding product health indicators, by adding process, voltage, temperature (PVT) and other sensors to their logic design</td>
</tr>
<tr>
<td>36</td>
<td>High Speed Digital Processor &amp; High Bandwidth Memory (HBM) Multi die Package</td>
</tr>
<tr>
<td>37</td>
<td>Logic/Sensor &amp; shielded multi-die packages</td>
</tr>
<tr>
<td>38</td>
<td>Molded multi-die packages</td>
</tr>
<tr>
<td>39</td>
<td>Multi-die power packages</td>
</tr>
<tr>
<td>40</td>
<td>Multi-die RF Front End Packages</td>
</tr>
<tr>
<td>41</td>
<td>Multi-die Silicon Photonics IC</td>
</tr>
<tr>
<td>42</td>
<td>Case Study Debugging chiplet 2 failure</td>
</tr>
<tr>
<td>43</td>
<td>Case study dealing with field returns</td>
</tr>
<tr>
<td>44</td>
<td>Case study for package-level TSV or IO failures</td>
</tr>
<tr>
<td>45</td>
<td>NELB path at pad &amp; NELB path isolated from pad</td>
</tr>
<tr>
<td>46</td>
<td>Failing path illustration based on test coverage</td>
</tr>
<tr>
<td>47</td>
<td>Failing vs passing frequency sweep</td>
</tr>
<tr>
<td>48</td>
<td>Failing schematic and signal waveform showing faulty path</td>
</tr>
<tr>
<td>49</td>
<td>(a) Microbump damage photos and measurements from direct microbump probing test. (b) Test specifications</td>
</tr>
</tbody>
</table>
Table of Tables

Table 1: IEEE Referenced Standards in BKM Document 9
Table 2: Packaging Integrator Disaggregated Product Deployment Checklist 60
Table 3: Fault Diagnosis Model 72
Table 4: Fault Diagnosis Model with Limited DVT Coverage 73
### Revision History

<table>
<thead>
<tr>
<th>Revision</th>
<th>Description</th>
<th>Edited By</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>Initial Document Revision</td>
<td>Abram Detofsky</td>
<td>6/10/2021</td>
</tr>
<tr>
<td>0.5</td>
<td>Draft Release to IEEE EPS Test TC</td>
<td>Abram Detofsky and Zoe Conroy</td>
<td>9/21/2021</td>
</tr>
<tr>
<td>1.0</td>
<td>1st Release to the public</td>
<td>Abram Detofsky and Zoe Conroy</td>
<td>1/14/21</td>
</tr>
</tbody>
</table>
1. Introduction

1.1. BKM Document Purpose

The purpose of this document is to consolidate many of the very best Testability practices and techniques which are needed to ensure a successful 2.5D or 3D product deployment. This document does not attempt to replace existing conferences, standards efforts or roadmaps, but rather supplements them by being a cohesive “one-stop-shop” landing-zone that all interested parties can come to for guidance. This document contains both original content as well as references to previously published materials. This document is a “practitioners guide” to the art of making and testing heterogenous products.

1.2. Audience

- Existing Standards Bodies (CHIPS Alliance, OpenHBI, HIR, etc.)
- Product Architects and Designers
- Test Engineers
- Tooling Engineers and Suppliers
- EDA suppliers
- Anyone interested in heterogeneous integrated product Design, Manufacturing and Test.

1.3. Inputters

Inputters are those individuals who are domain experts or have a strong vested interest in the high level design of the various capabilities defined in this document. Inputters will be co-authoring the content within this document.

<table>
<thead>
<tr>
<th>Name</th>
<th>Role/Focus</th>
<th>Sections led/authoring</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abram Detofsky</td>
<td>BKM document integrator; tester and methods architecture focus</td>
<td>High-level DFx Descriptions (Lead); test economics; co-editor</td>
</tr>
<tr>
<td>Vineet Pancholi</td>
<td>OSAT Test Architecture</td>
<td>Background (Lead) and Case Study</td>
</tr>
<tr>
<td>Terrence Tan</td>
<td>Debug and fault isolation</td>
<td>Debug and Fault Isolation (Lead); Fail Modes; Case Study (Die-2-die)</td>
</tr>
</tbody>
</table>
1.4. **Assumptions**

The intent is to capture the best-known thinking at the time of publication with the assumption that this document will be rev’ed on an annual basis and coincide with the annual release cadence of the IEEE EPS Heterogenous Integrated Roadmap (HIR).

1.5. **Scope**

By the end of each chapter you can understand the role of the chapters’ technology area in the overall Heterogeneous Integration Test Process. What is leading-edge and general practices in the industry, with BKM’s on how to do testing, or solving potential difficulties or issues that may come up. Each chapter can be read ‘standalone’ without having read all prior or subsequent chapters. Chapters may refer to prior chapters if background explanation is needed. This is not a comprehensive list of solutions or coverage of all technology available.
1.6. **Terminology**

This document follows a convention of spelling out a name completely and following it with an acronym in parentheses the first time they are found in the document.

1.7. **Reference Documents and References to IEEE Standards**

References are embedded within each chapter below.

<table>
<thead>
<tr>
<th>IEEE Standard</th>
<th>Referenced in these Chapters</th>
</tr>
</thead>
<tbody>
<tr>
<td>IEEE 1149.1</td>
<td>3.4, 9.10</td>
</tr>
<tr>
<td>IEEE 1500</td>
<td>3.4</td>
</tr>
<tr>
<td>IEEE 1687</td>
<td>3.4, 5.4</td>
</tr>
<tr>
<td>IEEE 1838</td>
<td>3.4, 5.4, 6.8, 9.10</td>
</tr>
<tr>
<td>IEEE 2929</td>
<td>3.4, 6.7, 6.8</td>
</tr>
</tbody>
</table>

*Table 1: IEEE Referenced Standards in BKM Document*

1.8. **Approval / Deciders**

We will use the Recommender/Approver/Decider template (RAD) for approval and review. Direct approval of this of this document is owned by the EPS Test TC. After review comments are taken and review a decision will be made to promote the document or why it should not be promoted. Parties there have access to this document should they have specific inputs.

Recommenders: Abram Detofsky and Zoe Conroy
Decider: by Pooya Tadayon
Approving body: EPS Test TC
2. Heterogeneous Integrated Products and Testability Overview

1.9. Summary Abstract
By the end of this chapter, you will learn why product Testability is a critical aspect that needs to pervade throughout the entire heterogeneous product life-cycle from initial product architecture definition to product resiliency and security in the field.

1.10. Background
In recent years there has been a sharp rise of heterogeneous system designs. Numerous publications targeting a large variety of applications exists in the public domain. Design and Analysis of Chiplet Interfaces for Heterogenous Systems [1] presentation on the IEEE’s website does a good job of detailing the anecdotal path of heterogeneous systems by way of chiplets building blocks integrated within a single package. The presentation includes references to a handful of example heterogeneous systems that include CPU, FPGA fabric, Memory, RF, IO, Analog. It describes advantages and dis-advantages of working with chiplets. In addition, it includes package design considerations, functional and performance aspects as well. While it serves as a good reference material as a background, it does not provide any detail for testability.

Over the past three plus decades, the number of packages and the packaging technologies have evolved, to accommodate the increasing I/O density.

Figure 1: High-end packaging roadmap: application-technology

Figure 2: Mapping of high-end packaging players based on technology categories
The purpose of this document is to describe testability best known methods (BKMs) for the class of products that have heterogeneous die integrated within a single package. Heterogeneous die, as the name suggests, are die that may have been fabricated on different substrates and using completely different fabrication technologies. Each die within the single die ensemble, in most cases, may represent different functional blocks of the platform or the end application. The end application may be requiring the collection of die to be integrated within a single package for a variety of reasons, such as the mechanical form & fit in the application or for the overall inter-IC performance, security or efficiencies.

1.11. References

1. Design and Analysis of Chiplet Interfaces for Heterogenous Systems
2. Heterogeneous Integration Roadmap – 2019 Edition
3. Testability Ecosystem and Heterogenous Integrated Test Economics

3.1. Summary Abstract
In this chapter, you will learn about the role of Test throughout the product ecosystem and how the economics of test influences various design and manufacturing choices.

3.2. The Impact of Test and Testability on Heterogenous Integrated Products Partners
Monolithic System-on-chip (SoC) silicon designs have been mainstream for many years, and a relatively mature supply-chain exists of IP-providers, EDA tool-chain suppliers, manufacturing integrators and related partners. The rise of Heterogenous Integrated products that split a monolithic design into discrete silicon chiplets/tiles brings with it a host of both traditional and non-traditional technical and commercial testability challenges.

The EDA supplier tools and methods are challenged to “stretch” into ensuring support of the latest agreed-upon chiplet strategies are implemented within their design flows. Multi-physics design and modeling flows are critical in the design of heterogenous-integrated products where the product designer can no longer be successful with a narrow monolithic view of the final design. Things like cross-chiplet power plane noise, thermal crosstalk and others can only be comprehended when comprehending the package system as a whole.

For the chiplet IP Designer, the testability challenge is several-fold. First, the chiplet needs associated design collaterals created for downstream partners including a testability strategy with test ports which are compatible with neighboring chiplets that it will interface with. On-board PVT sensors and cross-chiplet interconnect BIST DFx engines need to be considered and validated. The chiplet IP supplier needs to provide design collaterals including estimated worst-case thermal and power maps for their IP under both Test and Mission-mode conditions. Lastly, the IP provider must provide a secure way for the downstream integrators and customers to execute test content targeted to their specific chiplet to ensure quality of the chiplet in high-volume manufacturing (HVM), power-on debug fault isolation, and for customer-return diagnostics.

The chiplet manufacturer has the challenge of providing both relevant PVT-related fab Test data, defining a “good-enough” DPM criteria for known-good-die with the customer as well as a data-based long-term reliability guarantee for these passing units. Chiplets which
historically are “known-good”, may not actually have sufficient margin when combined with neighboring die from different manufacturers or from different process nodes, yielding an integrated product which may either fail outright or exhibit failure when assembled or exhibit intermittent failing behavior that are time-consuming to reproduce and diagnose. Transistor aging in the field is also a known phenomenon which can affect product health and this likelihood increases with the number of total transistors in the assembly. High-quality chiplet reliability models are critical in predicting how a chiplet will age in the field and deploying sufficient margin and counter-measures in the design for those realities.

All roads ultimately lead to the product package integrator who has the task of combining multiple chiplets onto a single substrate and guaranteeing the packaged product to the platform customer. Testability features are critical to ensure all chiplet-to-chiplet and chiplet-to-substrate IO and power interfaces are working as-intended with margin post-assembly. The quantity of these connections can easily exceed 100k connections package-wide. This demands that each of these chiplets and their interfaces have exceedingly low DPM requirements, else the full assembly will not yield acceptably. The integrator will typically employ techniques like onboard sensors and interconnect BIST engines to gauge interface health, and be able to repair and tune low-margin/failing IO lanes with redundant resources on a shared chiplet boundary.

3.3. Testability Gaps Assessment

Figure 3 illustrates a representation of a generic disaggregated product and its associated testability infrastructure. In this diagram, one will notice three chiplet die that have been assembled onto a common organic package substrate either directly or indirectly through a silicon bridge technology offered by various suppliers. The exact technologies and blend of technologies used are largely unimportant for the purposes of this discussion. The test infrastructure can also be seen here as a collection of a “Test Host” in the form of a piece of ATE equipment and associated tooling or a more lightweight System Level Test platform as an example. The Test host executes Test content defined in a number of standards such as Open architecture Test-plan Programming Language (OTPL) or Standard Test Interface Language (STIL) and sends resulting data to one or more databases and analytics engines in a format like Test Data Management System (TDMS), Standard Test Data Format (STDF) or a supplier-proprietary format.
One of the requirements that a product development and tester tooling engineer will want to know is the 2D power consumption and thermal dissipation maps that each of the chiplets will be requiring both in “mission mode” customer end-use functional operating conditions, but also and more importantly during test modes of operation. It is common for test content such as parallel scan chain execution to exercise much of the logic content simultaneously than what would normally be seen in a customer system. As such, the 2D Test-content power maps can be in excess of an order of magnitude greater in magnitude than what you would typically see in a power map executing mission-mode content. As multiple chiplets are combined with their own unique mission-mode and test mode power consumption characteristics, the designer must be able to use blended expertise and multi-physics and multi-design co-optimization tools to determine if the entire die complex is both manufacturable and usable in a customer system. This “blended domain” (thermal, mechanical, DfX, Signal-integrity and Power integrity) and the ability to co-optimize an entire complex of chiplets is lacking in industry EDA tools today.

Also missing today from EDA suppliers is a full end-to-end tools, flows, methods, standard cells and coherent test methodologies for standards such as for example the IEEE 1838 die-to-die interconnect. This lack of EDA “completeness” makes it difficult for designers to guarantee that what they are implementing is truly standard-compliant and also ensure they are interoperable across different designs created by different design suppliers or using different EDA toolchains.
A number of chiplet-to-chiplet “plumbing” standards exist including:

- AIB [https://chipsalliance.org/; https://github.com/chipsalliance/AIB-specification]
- BoW [https://www.opencompute.org/wiki/Server/ODSA]
- OpenHBI [https://www.opencompute.org/wiki/Server/ODSA]

Most of these specifications only describe how to assert test modes to enable boundary scan and similar modes to get test data in and out of the devices, but most often do not describe more critically how to guarantee the interface is good. For example, when determining that a chiplet-to-chiplet interface is good, it is common for either one chiplet to contain DFx that generates pseudo-random transmit traffic and the companion chiplet would contain DFx to receive and check and/or loopback this incoming traffic. If for example Chiplet1 is designed to generate PRBS7 traffic and Chiplet2 is designed only to check for PRBS9 traffic, then the combined DFx of two chiplets will not be able to verify the health of the inter-chiplet IO interface as they are “speaking different languages”. This kind of testability mis-match is quite common and largely inevitable without proper planning, and this will lead to products which are utterly non-manufacturable unless corrected.

The debug flow in a chiplet-based product (see chapter 6) is also quite different from monolithic designs. Multi-chiplet assemblies have a particular challenge of fault isolation to determine root-cause in the presence of thermal, electrical and often functional crosstalk. These sometimes-subtle interactions make it imperative that there is a means to isolate and deconvolve sub IP blocks under Test in as much isolation as possible. Test content from suppliers must often be shared as well to the debug partner which may come from die-level testing from an IP supplier or manufacturer and systems for encryption of sensitive test content must be put in place.

Yield analytics is yet another area where there needs to be significant attention for a manufacturable product to exist. A chiplet-based design allows for the impact of defects on die yield to be mitigated to some degree by constraining individual chiplet surface area. This creates a chiplet pairing challenge in a multi-chiplet ensemble where chiplet-to-chiplet process variation and manufacturing sources will both impact the final assembly’s performance to spec, margin, yield and overall power consumption. A software-based yield analytics engine is essential to best optimize the best pairings to meet specific customer performance and power product volume targets and ensure minimal chiplets are scrapped.
3.4. High-level Heterogenous Integrated DFx Strategies and IEEE Standards

There are a number of standards that are particularly useful for heterogenous-integrated product design and manufacturing.

IEEE 1149.1 has been a foundational “workhorse” standard for many years in the digital logic product arena, and a vast number of component and board designs support some subset of this standard as a primary test access port for command and data transport. This standard was created before heterogenous-integrated logic products, but many of its core elements have been expanded in other subsequent standards such as IEEE 1687. [https://standards.ieee.org/standard/1149_1-2013.html]

IEEE 1500 defines methods for testing IP core designs within a system-on-chip (SoC) design. It describes the core under test (CUT) but says nothing about a hierarchy of multiple cores where IEEE 1687 defines these well. This methodology was extended by IEEE 2929 (below) to include scan and memory array debug. [https://standards.ieee.org/standard/1500-2005.html]

IEEE P2929 standard extends existing access mechanism standards to define an architecture for scan and memory array debug as applied to complex system-on-chip (SoC) products. It provides a consistent method for trigger-based freezing of SoC scan and array state. [https://standards.ieee.org/project/2929.html]

IEEE 1687 is a specification that describes an architecture for the configuration, operation and data transfer for on-chip networking of embedded instrumentation building on the TAP communication protocol defined in IEEE 1149.1. [https://standards.ieee.org/standard/1687-2014.html]

IEEE 1838, one popular standard, is sometimes known as “DFx for the 3D IC”. It is a die-centric standard that describes the plumbing and access mechanisms to allow multi-die within a die stack ensemble to allow for the transport of both control and data signals to the outside world. It defines a die wrapper register to enable modular testing, a serial control mechanism for low-speed configuration and data transfer, and an optional flexible parallel port for high-bandwidth test access. [https://standards.ieee.org/standard/1838-2019.html]
3.5. Heterogenous Integrated Product Unique Test Economic Considerations

Moore’s law is an economic law that observes a doubling of transistors on a chip at a regular cadence. The industry has worked hard to ensure that silicon cost scaling has kept pace despite profound increases in manufacturing complexity, and this permits the selling price of a CPU today to be roughly flat generation over generation despite increases in transistor count. To allow Test cost to keep pace with the silicon cost scaling, a strong industry push to deploy new DFx techniques, higher levels of test parallelism and enhanced structural testing were deployed.

Heterogeneous integrated products have been the next-enabler for achieving higher and higher transistor counts. Smaller chiplets, versus large monolithic ICs, allow for higher chiplet individual yield which helps reduce product cost. These chiplets can be mix-and-matched across multiple manufacturing suppliers and process nodes to allow an optimal economic balance by tailoring chiplets to exactly the process nodes that will deliver the desired performance and capability.

Die disaggregation has a profound impact on sort probe module cost. Unlike in a monolithic design which is only probed in a single touchdown by a single set of probe equipment, a disaggregated product design contains multiple chiplets, each of which will require separate probe cards, test equipment, and replicated test content (setup time, shorts/opens, etc..) overhead. This leads to an aggregate cost of chiplet-based product wafer test that is significantly higher than equivalent monolithic IC design. The disaggregated wafer test cost further increases with the number of die in the heterogeneous chiplet complex. Furthermore, the push for ever-higher interconnect densities between chiplets is driving the probe density limits of current probe card technologies that now need to contact chiplet die with advanced die-to-die processing techniques such as through-silicon vias (TSVs) and Hybrid-Bond Interconnect (HBI) at contact pitches of <10um.

Yield considerations are very large drivers in test economics. For a disaggregated chiplet ecosystem to take-off, the test infrastructure supply chain must deliver high-quality characterized Known Good Die (KGD) to packaging integrators. It is often said that there is no such thing as “Known Good Die”, but rather “known not-bad die”, and this is a pragmatic perspective for a test engineer to take. As more and more chiplets are co-assembled on a substrate, the overall yield for the packaged part drops. To combat this and prevent an outgoing quality free-fall, increasing pressure must be made to secure improvements in KGD outgoing characterized chiplet-level quality. But to deliver better and higher quality tests at the chiplet wafer level, one requires more expensive equipment, testmethod and test time investment. This challenge grows worse in multi-die stack assemblies where the interface
tested at wafer sort as “known good” is now paired with its companion die and thus needs additional DFx and test time to validate again as a paired chiplet-to-chiplet interconnect. Additionally, die area needs to be consumed for these DFx test engines and repair and redundancy strategies and the test time to execute them, both adding to cost.

Die disaggregation also demands that we also pay attention to what chiplets get paired together in a final chiplet complex, as all KGD chiplets are not all equally performant. For instance, in a multi-chiplet complex that contains higher and lower performing logic die as a function of their normal process variation, the assembled product will usually need to sell as the performance matching the lowest performant chiplet in the assembly even though other die may perform better. Often it is not easy to pre-determine which chiplets will have sufficient margin to perform better until post-assembly. In this way, it’s not sufficient just to know a chiplet is known-good, but it must also be accompanied by detailed characterization data to allow for higher-quality intelligent die pair selection by the assembly team.

Process variation within multiple chiplet die is much larger than what one would see within a single monolithic die. The use of chiplets then adds assembly yield risk and cost unless chiplet die are paired intelligently on a package. However, the pairing strategy may depend strongly on one’s product end-goals. For instance, combining chiplets in a purely random fashion adds risk of meeting the minimum frequency performance target for a product. Conversely, combining chiplets of all the same performance may meet the frequency performance needs, but may lead to a product that exceeds the maximum power requirements defined for the customer. As such, an intelligent and nuanced pairing “balance” must be achieved to simultaneously meet all product criteria.

Disaggregation also drives-up the cost of product debug as chiplets may drive the need for chiplet and chiplet-stack custom debug packages to allow for sub-assembly logical or physical debug that may not be possible in the full product package assembly.

**3.6 References:**
4. Die and Probe Layout Strategies to Enable Probing Best Practices

4.1. Summary Abstract
The steady march of Moore’s Law has diminished in terms of transistor density, but due to heterogenous integration and advanced packaging, wafer and singulated-die test (which we’ll call ‘probe’ from here on) has seen no relief in the demand for increased density or improved electrical performance. The once straight-forward challenge of contacting bond pads accurately has been complicated by the introduction of solder microbumps and the questions of how to contact them or whether to even try. Some chip designers get around probing microbumps literally, by placing special test-only pads or bumps amid the array. Another designer might try to contact several bumps with one probe or to use multiple probe types within one probe card. These and other design decisions are not purely technical but also financial. Every packaged product requires different considerations, but the optimal use of probe test is nearly always a compromise of what is possible and what makes sense cost-wise.

4.2. Industry’s call for KGB from Probe Test
With integration, and consequently value, shifting downstream from the front-end of the wafer fab to packaging and assembly, the role and importance of probe in the semiconductor manufacturing process is also transforming. Ideally, to maximize the yield of the final composite part, each of the component die going into the advanced package needs to be defect-free, and the sobering exponential arithmetic of composite yield loss has led to various calls for Known Good Die, or KGD, coming out of probe. However, it’s also important to keep in mind the fundamental reason for probe: cost reduction. At its essence, probe exists to screen out and eliminate bad die from downstream assembly processes. For leading-edge nodes, verification of a Known-Good state of every die on a wafer can be very expensive, but this can often outweigh the significant cost of downstream yield loss and scrap when aggregating multiple die into advanced packages.

4.3. Cost-savings and Compromises in Probe
Obstacles to Known Good Die. Even for classical monolithic die, KGD gets expensive for two reasons: first, to test a leading-edge chip at speed while physically contacting each I/O and power pin on the die requires a very complex and costly tester and probe card; and second, to verify a 100%-fault-free chip often results in long test times, which means the throughput of that expensive tester and probe card is not very high.
**Decouple packaging pitch and probe pitch.** The newest MEMs-type vertical probe arrays can accurately touch down on fine-pitch microbumps, that can often be economically unfeasible and risks the ability of the bumps to bond post-test. To ensure that probe does not restrict advancements in packaging, it is important to probe in ways that don’t require probe pitch to keep up with packaging pitch.

**Reducing the compliance for overtravel requirements.** One of the things that makes probe cards challenging is the fact that they are springs that have to operate not just electrically at fine pitches but also that they have to do so as compliant independent suspension since the probe card must make up for all of the topology and planarity differences across the die and associated with the test cell. That turns out to be an expensive choice to make. Achieving high compliance and serving all the requirements drives a lot of the cost associated with developing and building the probe card. If one wants a 40 um pitch, does it really need to have 100 um of overtravel or can we compromise on what compliances are required? In finding ways to compromise on certain requirements, chip manufacturers can get lower cost probe cards, and lower cost probe testing.

**Next generation probe cards.** Until recently, innovation in probe had been primarily focused on the mechanical challenges: pitch, alignment and probe count. Modern developments, particularly in advanced packaging, have challenged probe cards to push the limits of wiring density for increased bandwidth and improved power efficiency. R&D answers that call with continuous process improvement and creatively utilizing materials at scales they may never have been used before. The good news is that probe cards are technologically capable of such difficult tasks; probe can test HBM$^{1}$s through microbumps with KGD fidelity. The challenge is to do so at a cost point that makes the product economically feasible. Probe will always strive to hit the constantly moving targets of pitch and yield, but effective use of the product will continue to depend on design compromises that enable probe to fulfill its primary role: lowering cost.

### 4.4. Microbumps and Probe Cards

**Why not probe the microbumps?** To produce the required die-to-die interconnect performance, advanced packaging relies on exponentially denser chip layouts and a broad spectrum of interconnect/contact structures, from copper TSVs, to solder microbumps. With over 50,000 contacts in a one-centimeter square footprint, many applications have two to four

---

times denser contacting patterns than an equivalent monolithic device. Directly probing those structures would require driving electrical signals in and out of delicate micron-sized copper and solder structures, structures that also have stringent probe-induced damage criteria to prevent downstream yield loss.

**Availability of probe card vendors to contact microbumps in the current market.** A microbump array might have a pitch of 55 um with bump diameters of 25 um. The engineering challenge of contacting the bumps demands precise probe placement and planarity from the probe card and test cell. These requirements are quickly outpacing the demands for pitch and size that the probe industry had adapted to for monolithic devices. Only the three or four leading probe card manufacturers can reliably contact the tightest pitches seen on the latest SoC devices. As the microbump pitches continue to tighten, effective use of probe will likely incorporate strategies that avoid the need for probe cards at minimum pitch.

4.5. **Employing Test Pads Among Microbump Arrays**

**What are “dummy” test pads?** Instead of directly probing the TSVs that will form the die-to-die connections in the multi-die stack, dedicated probe pads are included in the component die design, providing electrical access for the required level of testability. An example of this setup might see 40 x 50 μm aluminum pads distributed amongst the array or microbumps. These “dummy” pads won’t be used for wire-bonding, so the size and pad damage constraints can be relaxed.

**Why use test pads?** Test pads enable standard DRAM probe cards and testers to probe the component die, leveraging decades of development of high-parallelism wafer test to increase throughput and reduce cost. When paired with Design-for-Test (DFT) strategies and some degree of repairability and/or redundancy, this approach provides an economically effective solution for HBM test and manufacturing.²

---

Tradeoffs to using test pads. Test probe pads are the current standard practice but the tradeoffs that they force ensure that the industry will seek affordable ways not to use them. Marinissen and Kiesetter\(^3\) note that, “although these dedicated probe pads achieve the job, they come at the expense of extra design effort, extra silicon area, possibly extra processing steps, extra test application time, extra load on the microbump I/Os during post-bond functional stack operation, and still leave the microbumps themselves untested.”

Figure 4: Bottom of SoC device in the stack provides test pads in the field of microbumps

Figure 5: (a) SEM image of MEMS cantilever spring probes. (b) Al pads are distributed among the microbumps to enable test using conventional MEMS spring probe technology. (c) Detailed side view of the spring tips contacting the Al pads in relation to the microbumps in both scrub and non-scrub directions.

4.6. Hybrid Bump and Probe Layouts

Use different probes in different areas of the die to do different jobs. There are some applications where there is no extra space for testing pads within microbump arrays. If there is some flexibility in layout, the chip designers can use a hybrid approach. This means allowing space for a bigger probe where large currents or high-power densities are required, while still using very fine-pitch structures to bring signals in and out of the die. By separating probes that do different jobs into different areas of the die, one can decouple the probes’ requirements.

Partitioning the real-estate on the die this way makes the job of probe a little easier which keeps the cost of the probe card down.

**An example of a hybrid probe array for application-processor test.** Figure 6 shows a probe array that uses two different probe types. The I/Os around the perimeter are very fine pitch, carry relatively low current, possibly at high frequencies. The powers and grounds are in the center of the array with more spacing to allow for a larger probe with 60% higher Maximum Allowable Current (MAC) and better power impedance (PI) performance. Probes that have their MAC exceeded can deform and fall out of plane with neighboring probes which can lower yield. When these “burnt” probes need to be replaced, the probe card has to be removed from the tester which is a potentially costly downtime event. Probe card manufacturers utilize composite metal technology to allow MEMS probes with different cross-sectional areas to have approximately equal wear rates, over-travels and stress fields.

---

Figure 6: Schematic of a hybrid probe array for application-processor test. I/Os at fine pitch are shown in red; powers/grounds with larger pitch (and current requirements) are shown in blue.

Figure 7: Power impedance (PI) comparison for critical power supply with 17 bumps. Utilization of higher current, 130 µm-pitch probes for PWR/GND allows 40% lower PI than with 80 µm-pitch alone.

Future opportunities for hybrid arrays. The hybrid array approach can be extended to enable other creative ways of achieving economical wafer testing. For example, a probe card might be designed to touch multiple bumps with one probe, or to skip some bumps while employing DFT and BIST to make up for the loss in test resolution. These and similar compromises are ways to
relax the basic requirements of the probe card. This allows for a lower cost probe card, reducing the cost of probe and reenergizing the ROI associated with it.
5. Interconnect BIST Best Practices including Redundancy, Repair and Rework

5.1. Summary Abstract
In this chapter you will learn the importance and role of interconnect BIST for Test and Repair of die-to-die interconnects.

5.2. Background
The increasing trend to integrate and package multiple dice into one SoC has been driven by yield, i.e. cost considerations, the need to integrate proprietary functionality from different sources into the SoC, etc. Semiconductor manufacturers responded by developing innovative interconnect and packaging technologies, including 2.5D and 3D variants. In 2.5D, dice are placed next to each other on a substrate and interconnected using dedicated IOs.

Figure 8: Cartoon of a 3D Stacked SoC showing the use of Die-to-Die communication

In 3D stacking, dice are placed on top of each other. Figure 8 shows an example SoC where two dice are stacked on top of a third die. Signals run between each of the two top die and the bottom die as well as between the two-top dice. To carry these signals, the collection of dice is interconnected, as shown, using 3D die-to-die interconnect. In this article we use the term die-to-die interconnects to refer to 3D die-to-die interconnect and focus on the test and repair of such interconnects.
5.3. **Logical Model of the Die-to-Die Interconnect**

The definition of the test and repair problem for die-to-die interconnects can be based on the notion of a “cluster” which is defined here as a functional Tx or Rx IP block that may be duplicated many times across a design to form a scalable parallel interconnect.

Figure 9 shows a 3-die stack. Standard buses run between these dice. A CXL bus connecting Die1 and Die2 is shown. CXL consists of signals from Die1 to Die 2 and vice-versa. Each signal uses a physical interconnect between Die1 and Die2. Designers split such standard functional buses into clusters. As shown in Figure 9, a cluster of size N, carrying N signals, consists of: a transmit end (Tx); a receive end (Rx); N die-to-die interconnects; 1 or more redundant die-to-die interconnects; and a forwarded clock going from Tx to Rx end. Redundant interconnects are used for cluster repair. The clock is used for synchronous data transfer between the two dice.

There are many variants of the individual die-to-die interconnects. This includes, but is not limited to: (i) face-to-face bonding, using solder bumps; (ii) face-to-face binding, using direct cu-cu connection; (iii) face-to-back bonding, which includes TSVs, etc. These variations impact the physical and electrical characteristics like area, density, performance, power-dissipation, etc. The logical abstraction of such connections is shown in Figure 10.
**Figure 10: Logical abstraction of the Die-to-Die Interconnect**

Figure 10(a) shows flip-flop (FF) F1 and driver B1 at Tx end, which are part of the Tx die. Figure 10(b) shows FF F2 and receiver B2 at Rx end, which are part of the Rx die. The “Interconnect” box between Figure 10(a) and Figure 10(b) abstracts the solder bump, cu-cu bonding, TSVs etc. used for connecting B1 to B2. Figure 10(a) and Figure 10(b) assumes synchronous data transfer, where a common clock is used to transfer data from F1 to F2. Signals, like CLK in Figure 10(c) and Figure 10(d), do not have a common clock and are transferred asynchronously. In addition to clocks, other examples of other asynchronous signals are resets, power-good, etc. Synchronous connections and asynchronous connections are clustered separately resulting in: (i) synchronous clusters; and (ii) asynchronous clusters. Henceforth, cluster refers to either of these two cluster types.

### 5.4. Die-to-Die Interconnect Test and Repair Feature Description

Solutions to the following problems are required to implement the die-to-die test and repair feature.

1. **No touch testing using standard DFT interface.** The footprint of die-to-die interconnect is very small. The footprint, and spacing, between such interconnects started with 50 µm and trending to be less than 10 µm. The number of interconnects vary from 1,000, for small designs, to 100,000 for larger designs. Physical probing of such densely packed signals,
without sacrificing density, is not feasible. Therefore, a no touch test, repair, and debug method, using standards like IEEE 1687 to access the DFT infrastructure for these interconnects, is required.

![Diagram of Victim-Aggressor Tests (VATs) for die-to-die interconnects.](image)

**Figure 11**: Victim-Aggressor Tests (VATs) for die-to-die interconnects.

2. **Tests.** Specialized tests for such interconnects are needed. They differ from scan tests used for the logic area on the die. Test for logic target random defects which are modelled using single stuck-at or single transition faults. Tests for stuck-at faults are run at slow speed whereas tests for transition faults are run at-speed. Such tests are typically created using an ATPG tool and applied using scan DFT.

A better test for die-to-die interconnects is the Victim-Aggressor Test, abbreviated as VAT, shown in Figure 11. For a cluster of size N, VAT consists of N phases: Phase 1 targets lane 1 where lane 1 is the victim; Phase 2 targets lane 2 where lane 2 is the victim; and so on. In Phase 1, where Lane 1 is the victim, signals on lane 1 transitions in one direction whereas signals on all other lanes, referred to as aggressors, transitions in the opposite direction. The result of the test is read out after each phase. This is repeated for other lanes in subsequent phase. In addition to detecting single stuck-at and transition faults on individual lanes VAT detects failures caused by capacitive or resistive coupling between the victim lane and aggressor lanes. Detecting coupling induced failures is important since: (i) these signals run in very close proximity; and (ii) the signal frequency is extremely high, running into several GHz.

General purpose IO (GPIO) tests use pseudo-random tests, often referred to as pseudo-random bit stream (PRBS) tests, to accomplish the same goal. Simulation studies performed, using our latest interconnect models, show VATs to be at least as effective as PRBS with considerably reduced clock cycles. Thus, use of VAT reduces test time and test cost. Apart from test quality, another advantage of using VAT is that it enables implementation of an
autonomous on-die diagnosis and repair functionality. Implementing on-die diagnosis and repair functionality using PRBS is extremely costly and therefore not feasible.

3. **Analog Tests and Analog Observability.** GPIO test infrastructure includes extensive support for: (i) no-touch leakage (NTL) tests; and (ii) observing analog waveforms at SoC IOs. No such infrastructure is required for IOs driving die-to-die interconnects for the following reasons. NTL assumes that the difference in leakage current distribution between good and defective IO is such that an NTL threshold can separate good from defective parts. For IOs used for die-to-die interconnects, setting an NTL threshold is not feasible since there is a significant overlap of leakage current distribution resulting from defects and process variation. NTL is difficult to use correctly if there’s a wide process variation between the two companion die that are communicating to one another. As Figure 10 shows, IOs used for die-to-die interconnects contain negligible analog components. Therefore, VAT tests run at slow speed and at-speed are adequate.

4. **Debug and Failure Analysis.** After a cluster fails, root-causing the failure consists of two steps: (i) identifying the failing lane; and (ii) identifying if the defect is in the Tx die, the Rx die or the interconnect. In addition, fab houses use additional non-destructive probing techniques, like thermal sensing, to locate failures. Support for steps (i), (ii) and non-destructive probing is required.

5. **Repair.** Given the large number of die-to-die interconnects and the complexity of implementing them, defective interconnects are a fact of life. Significant yield improvements are achieved, in current interconnect generation, using die-to-die repair. An on-die autonomous hardware solution which tests the cluster, identifies the faulty interconnects, repairs, and retests the cluster is required. The repair solution must be fused into the die for subsequent use. An effective repair solution must comprehend the kind of defects, e.g., point, bridging or clustering as well as TSV failure rates.

6. **System level test and repair.** Analogous to power-on self-test (POST) for memories, system houses require a system-level trigger mechanism that tests, and possibly repairs, die-to-die interconnect. POST addresses infant mortality issues, and later in the product life cycle, improves system availability.

7. **Security.** Any feature providing access to die internals is a security risk. A requirement for any die-to-die test and repair solution is to mitigate such security risk.

8. **Burnin and Reliability.** Like all new processes, especially with shrinking geometries, it is important to understand the reliability of these die-to-die interconnects. This must be translated into burn-in requirements.

5.5. **Need for a new standard for die-to-die interconnect testing**

There is an increased need to integrate die from different die-vendors. To facilitate proper test and repair of the packaged parts the above list of test and repair problems must be adequately addressed. An IEEE standard is required to address this problem.
The IEEE P1838 standard [section 3.4], a block diagram of which is shown in Figure 12, has standardized the DFT access mechanism for 3D stacked die. A standard has been developed for test control, using standardized SCMs, and test data, using standardized FPP. However, for die-to-die interconnect, it assumes a die-wrapper-register (DWR) and assumes this die wrapper to be part of the scan DFT infrastructure of individual die. This is not adequate since scan is not a good approach to solve the list of problems listed above. A holistic view of the die-to-die test and repair solution is required and an autonomous, hardware test mechanism for die-to-die interconnects is needed. To facilitate effective die integration an IEEE standard that comprehends solutions to the above problems while understanding the various interconnect types is required.
6. Debug and Fault Isolation Best Practices

6.1. Summary Abstract
In this chapter you will learn some of the best practices, challenges and pitfalls to avoid when debugging and isolating faults in a complex heterogeneous-integrated product that extend beyond the challenges posed by traditional monolithic designs.

6.2. Debug Challenges: from Monolithic → 2-die Multi-chip Package (MCP) → 2.5D/3D
Physical debug is limited with such integration and needs good DFT that allows debugging at package / substrate level. Assembly time is lengthy, slowing down debug throughput. Also, SOC test and debug is not equivalent to die-level, as test access and content can be different between die level and package level test. Test content built is usually at die level and thought is not given to testability at higher levels of integration.
Physical debug / probing on the interposer (active or passive) is not possible

![Diagram showing test flow](image)

“30 bare dies, each 99% quality... final package yield loss: 26%”
- How do we effectively debug and isolate failures at final package test?
- What is needed in each chiplet (bare die suppliers) to provide effective debug and fault isolation at final test
- How do we make use of debug capability for each die at package level?

“How do we deal with field returns?”
- How do we debug system test failures which is a measure of DPM?

Figure 13: Yield and Quality are the main concerns

During test, it should be ensured each chiplet or die can be run independently. Mission-mode testing (full package bring up and functional test) is challenging on ATE equipment. Any system application failures are very difficult to port to ATE.
Thermal issues due to stacked packages (PoP package on package) may require specialized power control mitigation. Inter-die dependency post assembly tests need to be validated/reviewed in a stacked model, or they may result in post assembly test problem. Chiplet vs package testing has many differences, especially on bring up hence test re-use not likely, introducing more variables to debug. Known good die (KGD) does not mean all shipped bare die will pass at package and system.

6.3. Test Program Flow to Accelerate Fault Isolation

Will a known good die need additional testing at package level (Final Test)? A critical component of final test includes detecting packaging induced faults and performing any means of recovery and repair for chiplet interconnects and RAM repair.

The test program flow for a 2.5D/3D package needs to comprehend real-time in-situ fault detection and isolation. The flow shown in Figure 14 shows an example of how we can build the final test flow to allow inline fault detection. The very 1st step is to screen for assembly failures.

In monolithic die, assembly screening is relatively straightforward. For 2.5D/3D, more complex considerations are needed as shown below. It is worth to note that one should avoid complex dependencies (no PLL, no firmware, voltage regulator works by default without analog trimming) for the 1st two operations shown in Figure 14 (assembly screen and chiplet/active interposer test access screen). If your design does have complex dependencies as listed, successful assembly screen of a chiplet will be at higher risk due to the need for these dependencies to be functional, healthy and well-understood. All of these are unlikely to be well-behaved immediately after assembly.

### Package Level Test Flow

- **Assembly Screen**
  - Power rails open/shorts
  - Package pin open/shorts
  - JTAG access
  - TSV failures
  - Pin leakage
  - Die to Die open/shorts
  - Die to Die repair

- **Chiplet & Active Interposer Test Access Mechanism**
  - Aliveness for test access for each die component in package

- **Chiplet and Active Interposer Bring Up**
  - Clocks distribution
  - Reset distribution
  - PLL aliveness
  - Voltage regulators aliveness

- **Die to Die Performance Test**
  - At speed die to die test
  - Crosstalk
  - Voltage and Time Margining

- **Test Coverage**
  - All planned logic, array, analog coverage
  - Performance testing

Key data logging
1. Which bump failures during die2die testing
2. IO failures due to TSV or IO buffer itself?
3. Which chiplet unable to pass test mode access (TAP and/or Parallel Test Port)
4. Which chiplet or interposer unable to bring up
Section 6.4 walks through the inline fault isolation for 2.5D package topology and section 6.5 walks through the inline fault detection for 3D package topology. Section 6.6 covers the same for multi-stack (3 or more) configurations.

### 6.4. Strategy for Passive Interposer
- Passive interposer consists purely of TSVs and die-to-die interconnects to connect active chiplets.
- Passive interposers have no active transistors, thus no debug nor isolation capabilities up to and within its boundaries.
- One of the chiplets can take the role of an SOC (primary) die that manages the communication among other chiplets. In this example (figure 15), chiplet 3 is the primary die that manages functional and test data movement across chiplets.
- Each chiplet is a known good die and we need to ensure wafer level test is able to achieve equivalence of test coverage as package level.
- In many cases, the passive interposer won’t be a known good die. It’s often difficult or not possible to fully screen manufacturing defects for all microbumps, die-to-die interconnects and TSVs at wafer-level due to the interconnect and probing densities required.
6.4.1. Assembly Screen

Figure 16 shows the order of the tests that can be used to screen for any 2.5D package assembly failures.

- **Step 1 and 2:** Conventional power and package pin open/short test.
- **Step 3:** Since subsequent tests require the use of TAP, it is important we check the TAP access for each chiplet. A simple TAP IDECODE for each chiplet can be issued, as this will ensure all the basic JTAG connectivity to each chiplet. TAP connectivity to each chiplet can be direct from the package pin for each chiplet or a shared TAP access where the access is controlled by chiplet 3 (the SOC/primary chiplet).
- **Step 4 and 5:** Isolate TSV or IO failure as well as pin leakage. One option here is to use boundary scan DFT. We can put the boundary scan into loopback mode as well as extest mode to isolate TSV vs. IO failures as shown in Figure 17. Refer to case study 4 for details.
- **Step 6 and 7:** Open/Short test for Die-to-Die interconnects. Since speed testing isn’t done at this step, this means that marginal or resistive defects might not be caught. Speed testing can be done at a later stage. The die-to-die DFT should be able to generate basic patterns such as simple static 0, static 1 and checkerboard patterns and use the test access port’s clock to advance state instead of a functional mission-mode clock. One can consider repairing the die-to-die interconnects at this step if that is allowable and desirable.
6.4.2. Chiplet Test Access

After the assembly test is completed, the next step is to perform a health check on all critical test mode access mechanisms besides JTAG. Typically, this is a form of parallel test port used for ATPG and other logic/array/analog testing. As shown in Figure 18, the easiest approach can be to implement the test port loopback scheme and this should be done with test access port speeds and no mission-mode PLLs should be involved. Test port programming and setup is done in the SOC/Primary chiplet, in this example is chiplet3.

In this example as shown in Figure 18, chiplet3 acts as the main chiplet to interact with other chiplets for transporting test data. We need to first check the test port access for chiplet 3 followed by chiplet 1, chiplet 2 and chiplet 4. The access to chiplet 1, 2 and 4 is via chiplet 3 hence ensuring test access to chiplet 3 first is important.
At this point, we have fully verified basic package assembly and test port connectivity. The next step is to verify the clock, reset and power delivery distribution as well as at speed die-to-die interconnects as shown in Figure 19. Clock distribution can be done with the PLL lock test. Each chiplet would also have a basic bring up sequence test to ensure each block is out of reset and has healthy clocks. Observability can be done via some internal register read or some other alternative. Voltage regulators can be indirectly covered with clock and reset tests and we need to ensure voltage regulators have analog monitors to look for any voltage issues. At speed interconnect can be part of the die-to-die DFT. Case study 5 provides some details of die-to-die interconnect debug.

After this step, we are ready for the remainder of die testing (scan, functional, memory bist, IO, performance, etc.).
6.5. **Strategy for Active Interposer**

- Active interposer itself is an active component, contributing to overall stack functionally, and could also include minimal TSVs for top dies.
- Active top dies may be multi-instantiated dies, or multiple unique dies.
- We will consider the following active interposer topology shown in Figure 20. The active interposer is the SOC/Primary die (similar to chiplet 3 in 2.5D topology discussed above). The active poser has an IP block (green box in Figure 20) that integrates all the chiplets connectivity as well as provide DFT access to the chiplets during package test.
- Each chiplet has its own PLL as this is needed for wafer level tests.
- We won’t be able to have direct access to each chiplet from the package pin as this solution is not scalable. Test access at die level can be different from package level. In this example, chiplets 1 and 2 which are instantiated multiple times are accessed via die-to-die interconnect from the active base die. In the active base die, there can be a block which handles the functional and DFT integration of all the chiplets 1 and 2.
- Power delivery to chiplets can be from active interposer voltage regulators.
6.5.1. Assembly Screen

Figure 21 shows a test order to screen for any 3D package assembly failures. The setup is similar to 2.5D described in Figure 14.

- **Step 1 and 2:** Conventional power and package pin open/short test.
- **Step 3:** Since subsequent tests require the use of TAP, it is important we check the TAP access for each chiplet. A simple TAP IDECODE for each chiplet can be issued, as this will ensure all the basic JTAG connectivity to each chiplet. In this 3D stack example shown in Figure 21, chiplet1 and chiplet2 TAP will be access via active interposer DFT hub. We need to ensure step 3 can be done via only direct power and JTAG pins.
- **Step 4 and 5:** Isolate TSV or IO failure as well as pin leakage. One option here is to use boundary scan DFT. Refer to case study 4 for details.
- **Step 6 and 7:** Open/Short test for Die-to-Die interconnects. Since speed testing isn’t done at this step, this means that marginal or resistive defects might not be caught. Speed testing can be done at a later stage. The die-to-die DFT should be able to generate basic patterns such as simple static 0, static 1 and checkerboard patterns and use the test access port’s clock to advance state instead of a functional mission-mode clock. One can consider repairing the die-to-die interconnects at this step if that is allowable and desirable.
6.5.2. Chiplet Test Access

Figure 22 shows the test access mechanism test for a 3D package with an active interposer. One unique requirement for a 3D topology is to ensure the base die (aka active interposer) has full isolation capability (i.e. no dependency on handshake signals with chiplets 1 and 2). This is critical to maintain equivalence between die level and package level for the active interposer. We will start with testing for the active interposer test access in isolation mode (step 1, 2 and 3 of Figure 22). When testing chiplet 1 and chiplet 2, the active interposer isolation mode can be disabled (steps 4 and 5 of Figure 22).
Figure 22: Chiplet Test Access Mechanism Test flow for 3D

6.5.3. Bring Up Sequence, Voltage Regulators and At-Speed Interconnects

Figure 23 shows the clock, reset and voltage regulator distribution test and the at speed die-to-die interconnect flow. For clock, reset, and voltage regulator tests for the active interpose die, we need to be in isolation mode for wafer and package level equivalence. All die-to-die interconnect tests require all chiplets and active interposer to be up. When testing chiplet 1 and chiplet 2, the active interposer isolation mode can be disabled since the clock/reset/voltage regulator distribution for chiplet 1 and 2 may come from the active interposer as shown in steps 2 and 3 of Figure 23.
6.6. DFT Consideration for Debug and Fault Isolations
Initialization (Clock, Reset, Power Delivery) and Interoperability
Mission-mode operation for clock (including PLLs) reset, and power delivery distribution and functional interoperability will not be friendly for test and debug. Some debug hooks are in Figure 24. Here are some of recommendation of clock, eset and power delivery DFT needs:

- No dependency on PLL and Firmware for assembly screen and test mode access check.
- On die voltage regulators should work by default without any complex trimming. This is the key for the assembly screen and test mode checks.
- For active interposer, there should be 100% equivalence between wafer level and package level test in terms of bring up and test mode access. As such a special “isolation mode” should be allowed where all input signals from chiplets to interposer are replaced with a valid safe state during “Wafer Test Mode”; this mode can be used in package tests as well during content delivery.
- Ability to enable or disable each chiplet for debug and de-featuring in case any faulty assembly or non-recoverable interconnect issues are uncovered post-assembly.
- Care must be taken to avoid thermal spikes on untested logic. Controls should be integrated into bring up to avoid over stressing untested logic and risk damaging parts.
- Thermal sensors on all chiplets should be placed close to high-power logic where possible in order to monitor thermal gradients within the chiplet complex and assist with debug.
6.7. **Die Level Debug**

To avoid packaging faulty die (KGD does not mean all shipped bare die will pass package and system test), there is a need to perform die-level debugging in a multi-chiplet complex. To achieve this, one must be able to run test content suited for post-assembly testing at die-level as much as possible. Some recommendations:

- By maximizing test content reusability across test sockets, isolated die level faults can be detected and debugged early without needing to wait for a packaged product.
- A major barrier to test content reusability is the power-on sequencing which would vary across test sockets. One example is the need to bring-up the “base die” in order to test the “top die” in a stacked packaged config. This requirement is exclusive to post assembly testing as for die-level, the bring-up sequence would be independent for each die. Due to this, there may be test coverage gaps between the two test sockets which could be addressed via specialized test coverage to mimic the power-on sequence implemented on the packaged product.
- When debugging at die level, we need to maximize controllability and observability to isolate the faulty logic or circuitry. Wherever possible, scan-able sequential elements or
registers should be implemented. The objective is to be able to halt and dump register values of any digital logic anytime during the test. The infrastructure should be enabled with a deterministic logic state for accurate diagnosis. [IEEE 2929 in Section 3.4].

- For critical signals/nets (digital and analog) that are crucial for debug and test enabling, one can route these out to an external (high bandwidth capable) IO bump for observation. These pins or bumps should be accessible during die level testing as well, thus tester probe access to bumps at die level should be planned with debug in mind.

- Debugging mixed signal IPs as such is very tough without the aid of physical debug. In this case, having a debug toolset that can perform auto tuning of analogue circuits, like adjustments done during circuit design simulation would greatly help debugging such circuits to identify if one is faulty. Of course, to achieve this, all tunable circuit adjustments need to be accessible via DFX overrides. This would be in addition to having the signal observability infrastructure as well as isolated and direct circuit test suites as discussed in the previous section.

- Due to difficulty in active interposer debug, analog IP used should be stable known-good solutions with sufficient design margin.

### 6.8. Test Access Controllability and Observability

- Each die needs to be accessed directly or with minimum dependency from other dies.
- Per chiplet test port loopback at test port speed

![Figure 26: Debug hooks for test access](image-url)
It is critical that each die in 2.5D/3D package needs to have unique identification of an ID register that logs which die is failing. This means each die needs to be assessed directly or with minimum dependency from other dies.

- For scalability, it is expected that we won’t be able to have direct access to each chiplet at package level. It is also expected that the test access between wafer level test and package test can be different. It is important that we have a DFT hooks that allow us to test each DFT path.
- For TAP-based test access, the requirement is for each die in the package to have a unique JTAG ID which is public and can be read-out during production test. TAP access needs to be as simple as possible – only TAP clock is needed, no PLL dependency, no firmware and on-die voltage regulators work when power is applied without any trimming. All die-to-die testing should rely only on the test access port.
- Parallel Test port access (i.e., for ATPG), will typically need to go through a DFT hub in each die on the package. A good method is to have a loopback path from DFT input to DFT outputs to verify for test access integrity.
- Where possible (as it is impractical due to package pin constraints), all top chiplets should contain package-routed pins for critical debug ports.
- Areas needing possible standardization or at least industry wide BKM’s:
  - Direct JTAG connection to each of the top chiplet to maximize access survivability for stacked configurations. Similar concept to platform boot debug using TAP to manually bring up the part in the system.
  - Standardization of DFD/DFT interface for commodity parts in stacked die applications. IEEE 1838 currently does not address debug hooks. IEEE P2929 should take this up?

6.9. Die to Die Interconnects (applies to microbumps and silicon bridges such as Embedded Multi-Die Interconnect Bridge (EMIB))

- BIST/Interconnect Tests for connections between active interposer and top dies - single point of failure at interconnect junction at die boundary (die 1 bump -> die 2 bump)
- Simplification of bring up sequence
  - Avoid any complex firmware, fuse or on die VR dependency
  - Using external pin to initialize
  - Always on domain
- Since bring up for certain chiplet may rely on d2d interface, simple bring up for d2d dft which don't depend on these signals is recommended, else plant dfx override paths on these signals on the receiving chiplet
- No compression of data. Able to read out per die2die information in diagnostic mode, able to tell straight away which lane failed
- Differentiate between die2die connectivity (intra die) vs. Inter die failure
- Slow speed mode - static all0 and all1 data
- Catch open and short issues
- Diagnostic DFT which is able to identify failing die to die interconnects for recovery/repair.
- Need full suite of loopback & d2d cross test at speed and DC for debug
7. Database, Yield Logic, Data Sharing and Analytics Best Practices

7.1. Summary Abstract
Exploiting data analytics best practices is critical for successful manufacturing of multichip products. There is an excellent document created by the HIR industry group related to data analytics in the HIR Test Roadmap at the following link: (pages 70-88)
In this BKM document, we won’t repeat all the detail available in the HIR document. Instead, we’ll briefly describe the opportunities and benefits – and we’ll provide links to other industry emerging standards.

7.2. Background
An important capability for heterogeneous packages is to be able to integrate data for all components – and to perform data analytics to drive a range of optimizations. The benefits of this capability will enable cost reduction, component/chain management, quality root cause analysis and yield optimization. There are several challenges to enable this data integration and analytics including the definition of the required data, data identification and traceability standards and overcoming the business concerns of sharing a set of new types of data from suppliers to integrators.
There is the need for a unique Die Identification for each component. (e.g., processors, SOCs, chiplets, substrates, interposers, DRAM bare dies, HBM, etc.)

7.3. HIR Data Analytics Roadmap
The HIR data analytics section describes details of the benefits, requirements and solutions for exploiting data for heterogeneous integration optimization. The basic capabilities include:

7.3.1. End-to-End (E2E) database
Includes data for each of the components (bare die). Die ID (e.g., stored on efuse for each die) is required for traceability. Data for each bare die could be stored on-die (efuse) or could be provided through data feedforward database. This End-to-End (E2E) database is key to enabling data analytics & optimization.
- A key challenge to enable this E2E database is enabling traceability of all components through the supply chain. There is a SEMI committee that has defined a standard related to traceability. (SEMI T23 – link is below)
- Another E2E challenge is to ensure companies who own the basic data is willing to share the data that will be included in the E2E data. There are a few challenges including business (confidentiality) concerns.
  - One of the best ways to motivate companies to implement/exploit this E2E analytics is having many case studies / applications published in the industry that clearly shows the ROI of such a system.
- Today, most implementations of E2E databases & analytics are done in an ad-hoc way. There aren’t data format and content definitions and requirements.

7.3.2. **Analytics using the E2E database**
- Use the full data suite to optimize the test content from End-to-End.
- Correlation of quality data and yield/fallout to optimize yield/test/quality trade-offs
- Root cause analysis of quality/yield excursions.

7.3.3. **Data feedforward methods**
- To enable advanced test flows and adaptive testing
- Provide die-specific data supplied by die suppliers to optimize Final Package Test and to customize the die personalization at the Final Package level. (e.g., VDD settings, repair data, tuning parameters, etc.)

7.3.4. **Security**
- An emerging concern is how to ensure the supply chain and all components are secure. There are a number of concerns related to security – we don’t try to cover all in this document. Below is a link to the “Trusted IoT Ecosystem Security (TIES)” industry group – those links provide excellent background material and proposed directions.

7.4. **Other Industry Standards / Working Groups**

**SEMI CAST: “Single Device Traceability Task Force”**
- SEMI T23 - Specification for **Single Device Traceability** for the Supply Chain. This Document establishes a standardized approach for enabling traceable device ID throughout the IC manufacturing, test, and assembly processes to the point of use in the final system.
0Device%20Traceability%20for%20the%20Supply%20Chain, -
Volume(s)%3A%text=This%20Document%20establishes%20a%20standardized,\
use%20in%20the%20final%20system.

**Trusted IoT Ecosystem Security (TIES)**

- “The goal of TIES is to promote Trusted End-to-End solutions in the IoT value chain that accelerates the adoption, growth and field use of connected chips, devices, systems and IoT applications while enabling recurring services revenue streams and high value business models.”
- https://www.gsaglobal.org/iot/ties/
- https://www.gsaglobal.org/iot/ties/ties-introduction-presentation/
8. The Changing Nature of ATE and Content Delivery

8.1. Summary Abstract

In this chapter you will learn how heterogenous integrated products are driving the need for novel methods, tester hardware and software to provide test coverage that spans uniquely across multiple chiplets.

8.2. Background

For decades most semiconductor test ATE strategies have been driven by some form of Acceptance Quality Limit (AQL) vs coverage vs test cost. In recent years we see different drivers pushing the requirements for low defect per million (DPM) and while driving DFx innovation at every DUT insertion. A large push for zero defects has been driven by automotive electronic content at an all-time high while digitalization of many aspects of business and personal life drives the storage, security, and management of that data in the cloud. As automation density increasingly is embedded in everyday life, the need for secure high reliability bandwidth and processing is driving an evolution to HIR-class devices.

Traditional ATE material testing flows at wafer, final and SLT tests have driven system growth in terms of pin speed, AC & DC accuracy, memory (vector) depth and optional modes while lowering overall cost per channel. Typical HVM testers in the last decade have focused on logic and memory testing for a single die or package with as many as 64 sites being tested in parallel. The ATE test flows with these types of devices have used common Protocol Aware (PA) test interfaces with JTAG, SPI, I2C, MDO, physical layers running at <= 100MHz. The various test and target modes functional patterns for all or sub-groups or ports of device pins require test vector-based pattern data extracted from EDA simulations. We see for single device test flow from 300-4500 test suite executions per device for each insertion and data logging requirements for these devices can be in 100’s of megabytes per device. Driving this large volume of data logging is everything from chip health monitors, device calibrations, parametric measurements to redundancy data with hard and soft binning all with the requirement for zero or low overhead processing and storage.
8.3. ATE Testing In Multichip Devices

Evolution of Multi-Chip Integration Technology

With multi-chip devices we require the same level of testing for each die/package in the system plus some System Level Test (SLT) requirement for the various sub-systems and final product test. Modern ATE equipment has flexible configuration options that allow scaling of pins, speed, vector depth, test domains (Analog/MX, Digital, RF, Power) with test program controlled flexible tester licensing. Testing a device in DC, Digital/HSIO, PMIC and RF domains in a single insertion has been a typical test case for mainstream ATE platforms since HP93000 gen-1 era of the mid-1990’s.

For our heterogenous-integrated device it’s important for the device pins or ports to have direct or indirect access so we can execute the modern test lists. We have devices with high-speed physical layers (HSIO) and a trend toward software defined functional testing as well as scan or DFT testing over these interfaces for better throughput while generating and transferring large amounts of device logging data with zero or low overhead. Within multi-threading ATE software, we can execute setups on multiple cores and run DFT test suite containers with interfaces over several sets of HSIO or PA PHY pins in parallel.

8.4. Re-invent HVM ATE Functional Testing
Complex digital devices have many cores and in the device application these cores work together. It’s difficult to get test coverage that span across multiple cores as deterministic test content (SCAN, MBIST, IOBIST etc) are typically intra-core-based tests. We get good defect coverage of the individual cores but no interaction between core(s) and host. Usually there’s no target mode software (firmware) during a test insertion so the test coverage gap can be an issue for functional or target mode operations.

To close this gap, we need to run complex functional test cases across cores having system-like interaction with external host and be able to setup device firmware and the host software stack. We will define new ATE use cases for system-level test content(containers) that needs to be executed on the host and/or the DUT during ATE test. The HSIO interfaces like PCIe and USB will be operated in their native mode using the customer’s S/W and F/W drivers. We can also speed up scan test content delivery over functional HSIO interfaces emerges as new test methodologies (IEEE 1149.10 or other propriety interfaces).

![Diagram](image)

**Figure 28: Scan & S/W Defined Functional Test over HSIO**

8.5. **Software-Driven HIR Test**
The target system-level software, firmware, drivers and test cases do not run via classic digital test vectors as they need a host processor talking to the device under test. To enable this, one method is to have test-head-resident instruments to enable software driven functional testing, and SCAN over HSIO interfaces for structural and SLT test cases using local compute resources.

As these instruments develop, we see both USB-X and PCIe-X interfaces in use depending on if device application type is in the mobile or HPC space. The instruments use one or more Computer on Module (COM) pcb’s in tester card format but using the standard tester pogo mechanical and electrical interfaces. This new instrument type has no classical digital setups used (levels/timing/vectors) and is built to allow customers to create custom COM programs. The open COM based cards use the same tester control software and programming language and are synchronous with the other instrument domains in the ATE system. This adds expanded functional coverage as well as constrained random testing features to our ATE platform. This approach increases structural and functional coverage without requiring additional tester vector memory.

![Figure 29: Functional Test Download & Execute w/HSIO](image)

We generate a large volume of data from traditional test suites as well as HSIO enabled functional test containers and scan logging from each device under test. This data can be used to adaptively change test program limits or re-arrange or change test sequence. This also adds complexity to the binning process and with KGD targeted material many customers are doing post-test or extended binning. This requires state of the art point of use analytics or edge computation.

Edge HPC products at point of test allows running complex test workloads with pre-configured containers for machine learning (ML), complex demodulation, or other high-performance workloads. These systems have one to one encrypted private connections that are closely integrated into the test cell making it easy to implement ML in semiconductor test environments. The customer’s device data privacy and security are key requirements that a local ML/HPC solution provides as well at better latency and local
storage for offshore operations. For many devices and especially HIR targeted applications we see common use of post-test WS/FT binning based on data from the entire test insertion chain.

Figure 30: Extended Binning for Advanced Packaging

9.1. Summary Abstract

The business of building and deploying a complex heterogenous product is a challenging endeavor for any packaging integrator or IP supplier. There are many “moving pieces” and multiple stakeholders both inside and outside one’s organization which need to be satisfied, not to mention getting the IP blocks themselves to work together and perform seamlessly and reliably as a completed system in customers’ hands. A deployment checklist can be useful as a guide to help ensure that a disaggregated package assembly will be manufacturable and successful as a product and that sometimes subtle design factors were not missed.


Table 2 shows a non-exhaustive checklist that a packaging integrator may consider when deploying a disaggregated product. The “category” column describes the category of requirement that needs to be satisfied. The “responsible party” column names the entity who is solely responsible for delivering this item. The “consulted parties” column describes the different entities that need to be consulted to deliver this item but who themselves aren’t solely responsible. The last three columns describe the checklist item itself, whether it’s been completed or not and any comments that the supplier would like to make.

<table>
<thead>
<tr>
<th>Checklist Index</th>
<th>Category</th>
<th>Responsible Party</th>
<th>Consulted Parties</th>
<th>Item</th>
<th>Completed?</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Inputs</td>
<td>Chiplet Manufacturer</td>
<td>packaging integrator</td>
<td>Received usable performance metrics (static current, dynamic current, process oscillator data, etc..) and binning info for each chiplet within my assembly for the purposes of optimal die pairing and</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Inputs</td>
<td>Chiplet IP supplier</td>
<td>Packaging integrator</td>
<td>Chiplet Manufacturer</td>
<td>power/performance estimation at assembly.</td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td>Packaging integrator</td>
<td>Chiplet IP supplier</td>
<td>Received usable reliability models for each chiplet within my assembly to understand how each chiplet will age both in assembly reflow process steps as well as customer end-use conditions</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Inputs</td>
<td>Chiplet IP supplier</td>
<td>Packaging Integrator</td>
<td>Chiplet Manufacturer</td>
<td>Received usable 2D/3D power maps under mission mode and Test modes for each chiplet within my assembly. include both Dynamic and Static models at standardized customer cold, room, hot temperature conditions</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Inputs</td>
<td>Chiplet IP supplier</td>
<td>Packaging Integrator</td>
<td></td>
<td>All chiplet digital models received from IP supplier including ICL/PDL, CTL, BSDL, STIL, etc</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Inputs</td>
<td>Chiplet IP Supplier</td>
<td>Packaging Integrator</td>
<td></td>
<td>3rd party Chiplet embedded IP cores have been received (ie, PCIe Phy) including IP datasheets, Test access Strategy</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Inputs</td>
<td>packaging integrator</td>
<td>Chiplet IP supplier</td>
<td>Chiplet Manufacturer</td>
<td>Proper NDA’s (including any 3-way+) are in place with all my IP providers/partners who I may need to interact with during product debug and</td>
<td></td>
</tr>
<tr>
<td>Step</td>
<td>Area</td>
<td>Sub Area</td>
<td>职责</td>
<td>Description</td>
<td></td>
<td></td>
</tr>
<tr>
<td>------</td>
<td>------------</td>
<td>--------------------</td>
<td>---------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Co-Simulation</td>
<td>Packaging integrator</td>
<td>Chiplet IP supplier</td>
<td>customer return events.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Ensure full-assembly acceptable thermal densities throughout all 3D/2.5D ICs under both mission-mode and all Test scenarios under worst-case loading in Test and customer thermal solution conditions</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Debug and Manufacturing</td>
<td>Packaging Integrator</td>
<td>Chiplet IP supplier</td>
<td>Defined a Design for Debug Architecture and use model plan. Ensured that all chiplet end-points expose compatible testability features and DFT engines to ensure debug and manufacturing are successful.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Manufacturing</td>
<td>Packaging Integrator</td>
<td>Chiplet IP Supplier</td>
<td>Created a Unified BSDL model at package level for board testing</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Manufacturing</td>
<td>Packaging Integrator</td>
<td>Chiplet IP Supplier</td>
<td>Defined a package monitoring and sensor strategy for all partners who will interact with the product</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Manufacturing</td>
<td>Packaging Integrator</td>
<td>Chiplet IP Supplier</td>
<td>Created an overall test strategy for full assembly and partial assembly, and system level package integration document</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>Manufacturing</td>
<td>Packaging Integrator</td>
<td>Chiplet IP Supplier</td>
<td>Completed High-temperature Operating Life (HTOL) and Design</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Chiplet Manufacturer</td>
<td>Chiplet IP Supplier EDA Vendor Chiplet Manufacturer</td>
<td>Defined multi-chiplet product repair strategy and process within manufacturing flow and ensured compatible repair rules are implemented across chiplets.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---------------------</td>
<td>---------------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------</td>
<td>---</td>
<td></td>
</tr>
<tr>
<td><strong>13</strong></td>
<td>Manufacturing</td>
<td>Packaging Integrator</td>
<td>Chiplet IP Supplier Chiplet Manufacturer</td>
<td>Completed datasystem assessment to ensure full chiplet die level traceability is assured via E-fuses or similar</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>14</strong></td>
<td>manufacturing</td>
<td>Packaging Integrator</td>
<td>Chiplet IP Supplier Chiplet Manufacturer</td>
<td>Completed security threat assessment and implemented measures to ensure chiplet-to-chiplet data links are appropriately protected throughout the supply chain from malicious attack</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>15</strong></td>
<td>Security</td>
<td>Packaging Integrator</td>
<td>Chiplet IP Supplier Chiplet Manufacturer</td>
<td>Completed and implemented an agreement with IP suppliers for data and test content sharing to ensure products can be debugged expeditiously during both initial product bring-up, but also for customer return events.</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>16</strong></td>
<td>Security</td>
<td>Packaging integrator</td>
<td>Chiplet IP Supplier Chiplet Manufacturer</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table 2: Packaging Integrator Disaggregated Product Deployment Checklist**
10. **Case Studies**

**Case Study 1: Multi-Die Packages and Implication to Test from an OSAT Perspective**

Out-Sourced Assembly Test (OSAT) houses are in a unique position in the industry since they experience a wide sampling of customer products. Higher volumes and higher mix of products result in a unique perspective of key learnings and missed steps for product packages with multiple heterogeneous die.

As of the writing of this document, the number of product packages with two, three and four die is high. Product packages, by volume, with die count greater than four is relatively low. The testability gaps assessment section [Section 3.3] includes the high level block diagram that shows typical inter-die connectivity. There are two popular examples of heterogeneous integration of customer products.

The first class of products are heavy on digital content. These are products with die from different IP providers and different fabrication nodes and may still be CMOS process die. Mostly digital products routinely implement testability features. [Section 3.4]

![Example Package Definition](image)

**Figure 31: Example Package Definition**

Consider the case where Die1 & Die3 are two digital peripheral functional blocks or memory blocks and Die2 being the processor die. Implementing test logic on this logic IC may be relatively less complex. Die1 may have inter-die connections to Die2 or to Die3, as defined for the end application. Functional & structural tests in the production test list may be used to trap manufacturing failures. Die1 and Die3 may have electrical I/O that may require production testing. Depending on the data direction - input, output and bi-directional, these interfaces may be tested with on-die loopback or on-loadboard loopback or on-loadboard external components or with tester instrumentation. Parameters specified in the product datasheet may require to be tested under a handful of test conditions, like min, nom and max voltage limits, temperature limits etc.
Inter-die electrical connections that are not exposed to pins externally on the package add to the production test complexities.

The second class of products are products with heterogeneous CMOS and non-CMOS content. These are products with silicon for the digital die, SOI for the RF, GaAs (Gallium Arsenide) for analog and RF components or GaN (Gallium Nitride) & Silicon Carbide for wide bandgap semiconductor products for power conversion. One such heterogeneous integration example is the mobile phone applications processor. The die within the package include a power management integrated circuit (PMIC) and a memory die, in addition to the baseband modem/processor. The second such example that has recently been driving higher production volumes is the integration of sensors, (e.g. MEMs) with limited digital logic like a microcontroller unit (MCU). There are a variety of reasons why testability features may be absent on such an example. The absence of testability features may result in failure modes that may end up resulting in lower production test yield, and may not be recoverable. Adding testability features helps with rapid root cause identification and a corrective measure being taken.

Inter-die electrical connections that are not exposed to pins externally on the package present production test challenges due to complexities to add the appropriate test logic.
Heterogeneous integration of die within a single package suffers from non-isothermal performance and so-called “thermal crosstalk”. Non-uniform self-heat generation within test conditions may negatively impact the test results and product reliability in the end application use.

Figure 34: Amkor’s 2021 Heterogeneous Integration Path Options

FCBGA, MCM FCBGA, 2.5D TSV/S-SWIFT package types are in production, while the others may be in limited adoption and pre-production stage as of the writing of this document, depending on the specific customer product requirements and business case. Interposers up to 43 x 36 mm & ASICs up to 32 x 26 mm have been qualified. I/O performance metrics of the three categories of packages are discussed in other parts of this document, including the subsequent case study sections.

High volume manufacturing (HVM) test simplifications typically result from Design For Test (DFT) structures implemented on die. There are lots of examples, like the test access port (TAP) implementations described with IEEE-1838[1]. Primary TAP engines are implemented on the base die and secondary TAP engines are implemented on the additional die enable Built-In Self Test (BIST) and interconnectivity within the package.
A fraction of the logic products have emphasized on adding product health indicators, by adding process, voltage, temperature (PVT) and other sensors to their logic design.

The data read from the DUT is useful during each test insertion and also during the system platform’s end use. Another separate DFT example is the Converged Pattern Generator & Checker (CPGC) [2] implemented starting with 14nm Intel SOC.

High Speed Digital Processor & High Bandwidth Memory (HBM) Multi die Package
In most common sensor applications, the read sensor data is sensed as a proportional voltage or current which is sent to a logic controller block that processes and reacts to the sensor. Analog and RF IC are sensitive to terrestrial radiation and are protected within a metal (Faraday shield) enclosure.

System-In-Package (SiP) more often than not requires custom & multiple voltage rails. For instance, conversion of battery voltage into analog, RF and digital voltages in a hand-held application require numerous independent power rails. These packages are gaining popularity within the automotive market segment, where custom MOSFET switch topologies are demanded by the end application, in say for example an electric vehicle (EV).
With the increasing demand for miniaturization in the cellphone & communication market segment, the demand for higher orders of integration of the RF front ends of the RFIC has continually increased. Multi-die packages that are also RF application friendly require careful shielding of the RF signal path from aggressor signals.

Silicon Photonics ICs (PICs) in the data center markets has also been gaining traction. The key advantage of using light and a fiber optic cable for transmission is effective data transmission rates that exceed rates that far exceed what's possible in copper. PICs themselves allow for handling the higher data rates on either end of the cables with a wide parallel interface to copper. The packaging of the optical and electrical multi-die have challenges, as shown above. Suffice to say that there are test challenges that this multi-die package presents, both in terms of optical and electrical paths with data rates that have a roadmap to exceed >100Gbps.

The key message from the examples listed within this case study is that there is a call to action to IC architects to consider adding logic that enables manufacturing simplification. The investment of design and IC real estate may be small, and the impact to total product quality, yield and speed-to-market is relatively large.

References:
Case Study 2: Debugging Chiplet Failures for Passive Interposer Topology

Without proper planning of the test program flow with in-line fault isolation, product debug and isolation work will be slow if not impossible. For this example, assuming that there is nothing wrong with assembly, physical debug on the known-good-die (KGD) top die of a multi-die stack is possible to further root cause a logic failure. In the case below, KGD “chiplet 2” with proper pass-through DFx in “chiplet 3” allows for proper chiplet 2 debug to be caused by mismatches in wafer-to-package test conditions for that chiplet, temperature differences or power delivery differences in the full assembly relative to the singulated die test results.

**Case Study: Debugging Chiplet 2 Failure**

![Diagram of chiplet 2 and chiplet 3 with DFT and test access points](image)

**Passing:** All Assembly Screens
**Passing:** All Chiplet Test Access
**Passing:** Clocks, Reset, PLL, Voltage Regulators and At-Speed Interconnects

**Failing: Chiplet2** failure can be attributed
a. Mismatch between wafer and package level test such as coverage, frequency, test condition, power delivery
b. Temperature
c. Thermal
d. IR droops

**Figure 42: Case Study Debugging chiplet 2 failure**

Case Study 3: Dealing with Field Returns with a product using an Active Interposer

Consider a hypothetical product with an active interposer which is a silicon interposer containing various CMOS logic. Assume a customer field-return has a failure coming from the active interposer portion of the package assembly, which means a test escape exists from our automated test equipment (ATE) and system level test (SLT) test steps. How does one deal with root-causing this?

Typically, the system validation team will engage with customers to understand the failure mode and attempt to reproduce using in-house system platforms as a starting-point. Once field failure can be reproduced in-house, system debuggers will perform deep debugging to decide if this is a defect, software issue, a “smoking-gun” silicon bug or analog marginality.
System failures need to be isolated using atomic-based test cases and system debug tools (register dumps) to pin-point the offending block and determination if it comes from a logic, memory or analog IP failure.

If the failure is a defect, manufacturing needs to screen this, and efforts must typically be undertaken to re-create the failure on a production tester.

- If the failure is logic-related, typically one would end up doing a functional test, but need to ensure the functional test has no dependency on adjacent chiplets.
- If this is memory related, typically one would end up customizing a memory test algorithm or using a functional method to access the memory to expose the failure
- If this is analog related, we would mimic the platform recipe first.

In any of the scenarios above, once a tester is able to reproduce the failure and screen (containment achieved), how do we find the defect mode? This is where die level debug comes into picture. We will need to bring the new test that we built to screen the “new defect mode, based on field return” to die level debug and failure analysis since we can’t do any physical debug at package level for the active base die.

**Important DFT considerations:** Any functional test that we need to perform on an active interposer must have no dependency on the attached chiplets else it will be difficult to achieve root-cause, defect isolation and ultimately moving this new content to wafer-test to improve its KGD coverage. DFT planning needs to comprehend any chiplet usage to send functional cycles to active interposer and mimic this operation using active interposer local cache.

---

**System:** requires CPU/GPU/SRAM to co-interact with failing block to reproduce the failure

**Tester:** Active interposer need to have a functional DFT injection block which mimic the down stream injection from CPU/GPU and capture the respond from the IP of interest. A local SRAM is needed.
Case Study 4: Package Level TSV or IO Failures

This case study shows how we distinguish between TSV and IO buffer failures during IO parametric (open/shorts/leakage) testing. We can make use of boundary scan 1149.1 in two ways shown in diagram below.

We will make use of 1149.1 EXTEST instruction to send the data via EXTEST instruction. We capture the data in two methods i.e., (a) at the pad (b) loopback to the receiver and capture into boundary scan cell. The table below shows the failure mode diagnosis base on different modes of boundary scan. Using this approach, we can figure out if the IO failure is due to TSV or IO buffer itself.

![Diagram](image1.png)

Figure 43: Case study dealing with field returns

Case Study 5: Die to Die IO Fault Model and Diagnosis Capability

![Diagram](image2.png)

Figure 44: Case study for package-level TSV or IO failures
A fault diagnosis model is introduced here for internal IO interfaces irrespective of packaging technology with the aim of providing Failure Analysis and Debug engineers a faster path to root cause failures that could stem from assembly, fabrication, design or test.

The model presented here significantly improves the success rate to root cause failures and overcomes many limitations of doing so with existing fault isolation techniques. A test-to-defect mapping table based on a single defect mechanism was developed and currently serves as a standard guide for Failure Analysis, Debug and Test engineers in diagnosing internal IO failures. As internal IOs do not have contact with the Automatic Test Equipment (ATE), only “no-touch” tests are available to screen IO-related failures. The primary test used for such IOs is near end loopback (NELB) which is also commonly employed to test IOs that exceed frequencies ATEs can support. Considering this, the IO loopback test is feasible to be used for screening as well as debugging internal IOs as they do not have touch access. Using the existing IO loopback Design-For-Test (DFT) feature coupled with die-to-die interconnect testing, an effective internal IO diagnosis model can be formulated without much overhead to isolate the failing component of a packaged device with more than one die.

How the combination of loopback and die-to-die interconnect tests are used to derive an effective fault diagnosis model is discussed in this section together with how an FI model is derived based on the loopback path. Figure 45 shows two different NELB paths in the AFE circuitry that are commonly used in multi-chip packages where one loopback path is at the pad and the other is isolated from the pad.
For the scenario where loopback happens at the pad, the possible failing test combination is shown in Table 2 below along with possible failing modes for each combination based upon a single defect/fault mechanism. Apart from the first combination where we can’t tell precisely which component may be defective (package, ESD diodes or active circuitry), the rest of the combinations actually provides the debug or FA engineer with adequate information to identify the failing component and in some cases the failing circuitry as well. For the second combination in Table 2 where die-to-die test passes although both dies fail NELB, one may argue that this condition is not expected. However, this scenario is possible if there happens to be a resistive defect which may not be caught by a DC test due to minimal leakage or a marginal timing issue. Thus, this scenario is possible if the loopback test is run at-speed while the die-to-die test is not (i.e., in DC mode). Principally, by understanding the circuitry covered by loopback and die-to-die tests, it is possible to localize the failing location by combining and studying the results of these tests.
Table 3: Fault Diagnosis Model

The scenario for which the NELB path is isolated from the pad would result in a much different diagnosis outcome with the same set of tests as can be seen in the right most column of Table 2. With the consideration of a single defect mechanism, the first two conditions would not be possible as both the loopback paths are well isolated, unlike the scenario in previous section where a failure on either die or package can affect the other. It can also be seen that the remaining failing combinations would result in different targeted failing circuitry although the failing component remains the same as compared to previous scenario. Due to the loopback path being isolated from the pad, any package or assembly related defects will result in the loopback tests to pass. As a result, in the event where any of the loopback test fails, we can confidently isolate the faulty die due to its isolation from the bump/package.

Without the availability of NELB and die-to-die test on both dice, the accuracy and granularity of fault diagnosis is significantly compromised. One example to demonstrate this is a
low-speed internal IO design which only has NELB on one die and a unidirectional/simplex die-to-die test. The diagnosis model for such DFT coverage is shown in Table 3. It can be seen that except for one condition where die-to-die test passes, the other failing conditions are unable to provide enough granularity to identify the failing component. This would typically result in very long throughput time to root cause or unsuccessful attempts at localizing the defect.

<table>
<thead>
<tr>
<th>DieA NELB</th>
<th>Die-to-die (B to A only)</th>
<th>Failing mode(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>fail</td>
<td>fail</td>
<td>Short at A/B/Diodes/Bump</td>
</tr>
<tr>
<td>fail</td>
<td>pass</td>
<td>Die A logic</td>
</tr>
<tr>
<td>pass</td>
<td>fail</td>
<td>Open Bump OR Die B logic</td>
</tr>
</tbody>
</table>

Table 4: Fault Diagnosis Model with Limited DVT Coverage

An actual internal IO failure diagnosis on 3D stacked device was demonstrated using the tabulated model presented where the loopback path is at the pad with both NELB and die-to-die tests enabled on the product. The device under test (DUT) failed loopback test on die B whilst passing on die-A. The unit also failed the DieB-to-DieA interconnect test at-speed but passed in DC mode. This indicates that the timing critical failure is on the TX path of the die-B, according to diagnosis Table 1 as tabulated in the previous section. While NELB passing on die-A indicates that the die is fault-free, the die-to-die test from die-B (TX path) to die-A (RX path) is failing together with die-B NELB. Hence, by eliminating the die-A RX path, what’s left is die-B TX, as illustrated in Figure 46.

Figure 46: Failing path illustration based on test coverage
Figure 47 below displays the failing voltage vs frequency sweep results compared to a passing DUT, which shows that the failure occurs at higher frequency and passes at low frequency. This coincides with the failing of high-speed test while passing at low-speed. From signal probing on the TX path before and after the flop isolated through FI as shown in Figure 48, it can be seen that the output data of the flop on the TX path is compromised and based on these findings, subsequent analysis showed a timing violation bug on the clock path design.

Figure 47: Failing vs passing frequency sweep

With the help of the tabulated fault diagnosis model, the failing die was immediately identified and the path to root cause was significantly shortened due to this. In the event of lack of test coverage as according to the scenario described in the previous section, fault isolation would have been a daunting task for the engineer. In this circumstance where the product is of stacked die configuration, based on the limited diagnosis as in Table 3, we would not know wherein the failure lies. This could cause a situation of root cause not found or the failing die potentially removed during FA process to identify the failing component.
Case Study 6: Direct Micro-bump Probing - HBM2 KGD Test

It is possible to contact all the microbumps and determine KGD. A 2019 collaboration between Advantest and FormFactor teams was able to show that it is technically feasible to do production KGD test on the individual DRAM die in an HBM stack at 3.2 Gbps. The images in Figure 49 demonstrate that a probe card was able to successfully contact all of the 25 um microbumps in at 55 um pitch. The probe card was also capable of driving signals in and out of the DRAM die through those microbumps to functionally test the DRAM die and confirm if they are known-good. With known-good confirmation, these die will be assembled in an HBM stack and integrated into the final product.\(^5\)

Is contacting all the microbumps practical? While possible to contact all of the microbumps on HBM stacks, it is still prohibitively expensive to implement on a production scale. Not only are the probe cards expensive and lower-parallelism than typical DRAM wafer-sort probe cards, the 2019 HBM2 KGD test showed that microbump damage worsened with increasing the number of contacts and also with increasing the contact times, which could compromise the eventual assembly yield. To minimize pad damage and contact resistance, low force probes were chosen, which is a design compromise paid at the risk of higher contact resistance.

---

Cost efficiency of direct probing on microbumps. A similar study was performed in 2014 that explored the economic feasibility of direct microbump probing test on a JEDEC Wide-I/O Mobile DRAM interface with 40/50 µm pitches. Marinissen and Kiesetter\(^6\) found that the possible gains in pad real estate and yield loss were minor contributors to the cost of test, whereas there was a 10x decrease in test time. The savings of the latter effect led the authors to determine that direct probing on microbumps is economically feasible.

The practicality of direct probing on microbumps is case-specific. The choice of whether to probe microbumps is determined by the economics of the test. The factors that influence the decision include: availability of extra pad space, cost of probe card, opportunity cost of tester time and yield loss to damaged microbumps. For different manufacturers and product types, these inputs might vary greatly, which is why both methods are used in the industry today.

\(^6\) Marinissen, “Large-Array Fine-Pitch Micro-Bumps”