GUIDE TO INTEGRATION WORKFLOWS FOR HETEROGENEOUS CHIPLET SYSTEMS

Boon Chong Ang Intel Corporation, John Caka Siemens EDA,
Kevin Cameron Cameron EDA, Lihong Cao PhD ASE (US). Inc,
KK Lin PhD Applied Materials, Subramanian Lalgudi PhD Siemens EDA,
Chieh Lo ASE (US). Inc, Mark Malinoski Siemens EDA,
Tony Mastroianni Siemens EDA, Jawad Nasrullah PhD Palo Alto Electron,
Rajesh Pendurkar PhD, David Ratchkov Thrace Systems,
Joe Reynick Siemens EDA, Satish K Surana Intel Corporation,
James Wong Palo Alto Electron
Executive Summary

The field of heterogeneous chiplet systems, particularly in advanced packaging, has seen significant growth and interest recently. This surge is driven by the evolution of chiplet design and packaging manufacturing, which offers innovative solutions to overcome challenges posed by the slowing progression of Moore's Law and the limitations of large, monolithic System-on-Chips (SoCs).

This whitepaper delves into the entire lifecycle of designing a chiplet-based system. It begins with System Planning and Co-Design, moves through System Implementation, and concludes with System Verification and Signoff. The document serves as an essential guide for developing a comprehensive and robust design flow and methodology tailored for the new era of chiplet systems.

A key focus of this whitepaper is highlighting the unique subflows required for chiplet-based design, such as System Technology Co-Optimization (STCO), Thermal Analysis, Mechanical Analysis, and more. These are integral to addressing the specific needs of chiplet-based systems and expand upon conventional chip-level subflows like Power Modeling, Signal Integrity, Timing Analysis, and Design For Test (DFT).

This document stands as a crucial reference for anyone involved in the chiplet system domain. It provides fresh perspectives and methodologies for optimizing and modularizing systems at a broader level, particularly for domain-specific products, where Open Compute Project (OCP) Open Domain-Specific Architecture (ODSA) plays a pivotal role. The insights and frameworks presented in this whitepaper are instrumental in navigating and excelling in the rapidly evolving landscape of chiplet-based system design.
Compliance with Open Compute Project Tenets

Openness
In this white paper experts in their fields have contributed open workflows for the benefit of anyone taking the endeavor of integrating chiplets into heterogeneous systems.

Efficiency
This white paper strives to improve the efficiency of architects, engineers and manufactures involved in the integration of chiplets by providing complete reference to the various workflows required to implement heterogeneous systems.

Impact
This white paper is the first to describe the entirety of workflows employed in the integration of heterogeneous chiplet systems.

Sustainability
The contributors are hopeful that this white paper, being the first of its kind, will benefit the anyone interested in growing the Open Chiplet Economy.

This work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International
# Table of Contents

**Executive Summary** 1  
**Compliance with Open Compute Project Tenets** 3  
  - Openness 3  
  - Efficiency 3  
  - Impact 3  
  - Sustainability 3  
**Glossary of Terms** 7  
**Personas** 8  
**Partitioning and Chipletization** 11  
  - Introduction 11  
  - Purpose of flow 11  
  - Expected results 13  
  - Flowchart 16  
  - Input/output description 19  
**Architectural Planning and Analysis workflow** 21  
  - Introduction 21  
  - Product Specification Inputs 22  
  - Intellectual Property (IP) Inputs 23  
  - Output descriptions 23  
  - Architectural Planning Workflows 24  
    - Micro-architectural Specification 24  
    - SiP RTL Design 25  
  - Architectural Analysis Workflows 25  
    - Predictive Analysis Overview 25  
    - SiP Floor planning 26  
    - Predictive Thermal Analysis 26  
    - Predictive Mechanical Stress Analysis 27  
    - D2D Datapath Planning/SI Analysis 27  
    - Power Delivery Network (PDN) 28  
    - System DFT Specification workflow 28  
**Functional Simulation and Verification workflow** 30  
  - Introduction 30  
  - Purpose of flow 30  
  - Methodology with Chiplet Systems 30  
  - Defect and Fault Coverage 31
BER/CDC 32
Simulation and Verification Flow 32
Functional simulation and verification plan 33
Chiplet-level verification 34
System-level simulation and verification 35
Hardware Validation 37
Tape-out 38
Input/output description 38

Physical Planning workflow 41
Introduction 41
Physical Planning Background 41
Conventional IC Physical Planning and Design 41
Physical Planning for Chiplet Integration 43
Input/output description 45

Power Planning and Analysis workflow 47
Introduction 47
Purpose of flow 49
Power modes 50
Inputs/Outputs 50

Thermal Modeling and Analysis workflow 52
Introduction 52
Purpose of flow 52
Inputs/Outputs 54

Signal Integrity and Power Integrity workflow 56
Introduction 56
Purpose of flow 57
Expected results 57
Input/Output Description 61
Flowchart 62
Input/Output Description 66
Flowchart 66

STA and Extraction 70
Introduction 70
STA Overview 70
SiP Level STA Overview 71
SiP Level STA Workflows 72

This work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International
Input/Output Description 73

**Mechanical Analysis workflow** 75
   Introduction 75
   Purpose of flow 76
   Flowchart of Mechanical Analysis 77
   Input description 79
   Output description 80

Figure 23 shows the thermomechanical stress analysis flow chart for chiplets integration by using FEA (Finite Element Analysis) 81

**Test Workflows for Multi-die System-in-Package** 83
   Introduction 83
   2.5D and 3D Overview 83
      2.5D Design Overview 85
      DFT in 3D Designs 89
   Purpose of the multi-die flow 92
   Expected results 92
   Flowchart 93
   Types of manufacturing test 94
      Wafer probe 94
      Package test 96
      IP Test 96
      Other tests 97
   Inputs/Outputs 98

**Conclusion** 100

**References** 101

**License** 102

**About Open Compute Foundation** 102

[Creative Commons Attribution-ShareAlike 4.0 International](http://creativecommons.org/licenses/by-sa/4.0)
Glossary of Terms

ATPG: Automatic Test Pattern Generation. EDA tool-generated test vectors that detect logic faults in scan based designs.

BIST: Built-In Self Test. Design technique which adds pattern generator and response evaluation internal to the chip either to test random logic (Logic BIST) or regular logic such as Memories, Register Files (Memory BIST)

CDXML: Chiplet Data Exchange Markup Language

SoC: System-on-Chip. Expands IC's from simple devices composed entirely of general purpose IO (GPIO), logic, and memories, and may include 3rd party IP cores like SerDes, DDR Phy's, and special IO's like LVDS and PECL.

SiP: System-in-Package. Multiple dies including SoC's and Chiplets are instantiated inside a single chip package.

Chiplet: a chip from a third party that usually has a specific function. For example, an Ethernet subsystem that is comprised of a SerDes and MAC, but also includes a PLL. Chiplets are supplied as fully tested bare dies and assembled into the SiP with one or more SoC's.

2.5D: Essentially this is like a PCB (2D), but more complex. With 2.5D, the PCB is called an Interposer, and is made in a fab from silicon. 2.5D technology has a much finer bump pitch than standard IC bump pitch by using microbumps. These microbumps allow many thousands of interconnects between IC's and achieve better performance than standard PCB's. The devices in a SiP use IO buffers to communicate between the IC's that are assembled to the interposer.

3D: 3D devices are individual dies that are vertically stacked upon each other. The dies are connected to each other by using Through-Silicon Via's (TSV's) which may or may not go through a simplified IO buffer with ESD protection. The advantage compared to 2.5D is that even more die-to-die (D2D) interconnects and better performance can be achieved with 3D interconnects. The die that communicates to devices or package pins outside of the 3D stack is called the base-die. The base die contains full IO buffers like GPIO, LVDS, or even SerDes. The base-die is usually the bottom die in the 3D stack, but that is not a requirement. See Figure 24.

5.5D: A combination of 2.5D and 3D technologies per IEEE 1838. For example, an ASIC on an interposer, with four 3D HBM DRAM stacks that are also mounted on the interposer and communicate with the ASIC. See Figure 24.

Defect Density (D0): Silicon defect density is a major component of yield calculation, and is expressed in defects per mm2.
**MCM**: Multi-chip Module. An older technology that is similar to 2.5D, but instead of silicon, the substrate material is based on FR-4 (fiberglass), PTFE (Teflon), or ceramic. The MCM bump pitch and wire pitch are like printed circuit boards, so the complexity is less than 2.5D.

**PCB**: Printed Circuit Board. typically, an FR-4 (fiberglass) substrate board with 1 or more layers of conductive traces that are used to connect multiple IC’s, active, or passive electronic components.

Interposer: A silicon-based substrate that is used like a PCB to connect multiple IC’s with much finer pitch and higher connectivity. Interposers are manufactured in an IC Fab process line. See also 2.5D.

**Process Complexity (N)**: is a component of the silicon yield calculation, that is based on the number of process layers and layer types.

**KGD**: Known Good Die testing. A wafer sort test strategy where the full die-level test program is run. This can be challenging due to the finer bump pitch for 2.5D/3D devices, along with the noisy and high RLC test environment, and reduced power integrity due to a sampling.

**Unified BSDL**: This boundary scan description language (BSDL) is used for board or system testing. 2.5D SIP’s have 2 categories of IO. The first category is IO that connects to other dies. The second category is IO that connects with package pins. With Unified BSDL, we build a single package-level BSDL comprised of IO that talk to package pins. Also, we use the TAP BYPASS register, and 32-bit DEVICE registers from the primary TAP for IEEE-1149.1 compliance. We also use the combined instruction registers from all the TAP’s in the 2.5D package.

**Yield**: The percentage of good parts with respect to all parts, either tested or untested. Silicon yield is a function of area, process complexity, and defect density. Yield has a direct impact on the profitability of a SiP project.

**Defect Density**: The inherent defectiveness of a silicon process which is expressed as defects per mm2.

**Personas**

Chiplet Integration is an intricate and multifaceted process. It necessitates a collaborative effort among engineers, architects, vendors, and various other stakeholders. To effectively depict the data flow within this process, we have compiled a comprehensive list of Personas. Each Persona symbolizes an individual who plays a critical role in the workflow, either by contributing essential information to a sub-flow or by receiving the output from that sub-flow. Detailed descriptions of these Personas are provided in the table below, and they will be referenced throughout the subsequent chapters.
<table>
<thead>
<tr>
<th>Persona Definition</th>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
</table>
| System Architect   | SA      | Involved in partitioning  
System/product level PAPLD, ALDPAC trade off analysis  
Maybe UPF designer  
Specified d2d protocols and speed and i/f requirements, lower speed interfaces  
Generates functional specs |
| DFT Architect      | DA      | Closely works with SA and the marketing requirements to study the use case and define and specify SiP level DFT architecture including ASIC DFT requirements |
| Structural Netlist Designer | SND | Generates connectivity  
Could be SystemC/TLM/RTL/Structural Verilog  
Maybe UPF designer  
Creates constraints |
<p>| Material Property Specification/suppliers | MPS | Provide package material (UF, Molding, PI, Substrate. etc) property (Modulus, Possson's ratio, CTE, Tg, Dk, Df, etc.) |
| ASIC physical design lead (APLD) | APLD | Top level floorplan of ASIC/chiplet and IO/pad ring design Interfaces with Package architects and Block engineers |
| ASIC logic designers | ALD | Generate RTL, runs synthesis, LEC, STA/SDC/UPF constraints, maybe running gate level regression |
| ASIC DFT           | ADFT    | DFT planning and logic insertion, test pattern generation |
| ASIC Physical Verification | APV | Routing, DRC/LVS, STA |
| System DFT engineer | SDFT    | SiP DFT planning, test interconnect, test pattern generation |
| System DFx         | SDFX    | Thermal, stress, mechanical, system yield, IR drop, EM, Bringup/Debug, Diagnosis |
| System verification engineer | SVE | Functional verification of complete system, logic, optional: formal, timing (CDC), RF and power/thermal |</p>
<table>
<thead>
<tr>
<th>Role</th>
<th>Abbreviation</th>
<th>Responsibilities</th>
</tr>
</thead>
</table>
| Package architect           | PA           | SiP level planning  
Support early SiP level analysis  
Makes choice for packaging technology |
| Package layout/verification | PLV          | Substrate and organic interposer layout, DRC,                                   |
| Si Interposer layout/verification | SILV   | Si interposer PNR, DRC,                                                          |
| PI/SI engineer/designer    | PISI         | Design and verify for PI and SI                                                  |
| Mechanical designer/engineer| MECH         | Design and verify all aspects of Mechanical and Reliability  
Design, including stress, warpage, and corresponding  
mitigation strategies, such as stiffener ring design, system  
assembly components, such as heat sink pressure forces, etc. |
| Material Engineer           | ME           | Selecting, developing, testing, and improving materials used in semiconductor  
manufacturing and integrated circuits. This includes working on materials at the atomic/molecular  
level to achieve specific electrical and physical properties. |
| Power Engineer              | PE           | Design and layout power distribution networks, perform power  
analysis, power domain partitioning, power integrity, power pin location |
| Thermal Engineer            | TE           | Analyze thermal behavior, temperature distribution, define thermal management |
| Chiplet vendor              | V            | Chiplet vendor, manufacturer, or supplier                                       |
Partitioning and Chipletization

Introduction

Chiplet based designs provide a variety of options to improve performance, power and area tradeoffs in systems.

- Disaggregation for relieving limits related to area/reticle size and power dissipation, as well as design cost, mask cost and wafer starts

- Increased integration related to interface lengths and drive strength, near memory size, programmable logic capacity, heterogenous combinations of fab processes, combining analog and digital on one substrate/interposer, etc.

- Supply chain tradeoffs as off-the-shelf chiplet availability increases, vs. ASIC chiplet designs.

This paper suggests a design and verification flow enabling chiplet based designs to meet system requirements.

Purpose of flow

S/W Defined Functionality of systems is leading to a “Cambrian-like” explosion in workload and complexity. Implementation of functions must fit within an electronic stack often realized by the involvement of several suppliers. Co-architecture, co-design, co-optimization and co-verification requires advanced allocation and partitioning methods and tools, referred to as cybertronics design automation.

A typical cybertronics stack will consist of elements such as enclosures (LRUs, ECUs), PCBs within the enclosures, IC and SiP component packages mounted on the PCB, and silicon chips/chiplets within the IC component packages. (Not shown is the cable harness that connects the enclosures.)

Currently, these cybertronic stack elements are developed by different supply chain participants, with disconnected design and verification flows. The software functionality desired is realized when chips and chiplets execute the code and algorithms. Verifying proper systems behavior is increasingly becoming more complex than humans can handle. We need advanced methods and tools to digitally engineer the cybertronic stack, across supply chain boundaries.

The challenge to designing a cybertronics stack which the functionality can “fit” into, is determining how hardware dependent the functionality is. We break down hardware dependency in 5 levels, with level 1 being the most hardware independent software, and level 5 being so hardware dependent that the algorithms must be implemented in discrete circuit blocks without software being executed, typically
referred to as “accelerators”. Our focus is on a partitioning method that can engineer at levels 3 and 4, that is the threshold of software meeting hardware to realize system functionality.

Figure 1: Contractual Functional Allocation Levels. Courtesy Siemens EDA

The desired system or software defined functionality is provided to development teams as requirements, functional representations, and preliminary architectures. Before engaging the development teams, the notional architecture must be analyzed to assure its risk of realization is minimal, while achieving the optimal architecture to realize system requirements. Increasing functionality and time sensitivity requires analysis of performance, power and size/area to assure system requirements can be met. This analysis needs to precede the start of development of the element layers. But it must also fully leverage the subject matter expertise of the design teams to be involved. That will entail continuous decomposition and refinement within each layer’s design teams, while continuously verifying that the resulting architecture is optimal.

The development teams need to refine requirements to meet enclosure sizes and space constraints, PCB interconnection within and thru the enclosure, component packages designed to be mounted on the PCB, that contain chips and/or chiplets capable of executing the s/w and realizing the system functionality.
There must be commonality and interoperability in how each layer in the cybertronic stack performs this decomposition and refinement, in a verifiable manner.

Figure 2: Verification Capture Point (VCP) Solution Pattern. Courtesy Siemens EDA

Area requirements need to be pushed down to the SIP/SOC level. Form factor is an important consideration in driving a monolithic vs heterogeneous approach and that could further drive 2.5D vs 3D implementation. 3D ICs will have a smaller form factor, but much larger cost, and does not offer product configurability; E.g. including/excluding different chiplets in a SIP is much cheaper and easier/quicker than doing different 1D (single monolithic) or 3D (stacked die) implementations. Once performance requirements are met by design solutions, further selection and optimization can be done to meet or revise costs and schedules.

Expected results

Before engaging the development teams, the notional architecture must be analyzed to assure its risk of realization is minimal, while achieving the optimal architecture to realize system requirements. Increasing functionality and time sensitivity requires analysis of performance, power and size/area to assure system requirements can be met. This analysis needs to precede the start of development of the element layers. But it must also fully leverage the subject matter expertise of the design teams to be involved. That will
entail continuous decomposition and refinement within each layer’s design teams, while continuously verifying that the resulting architecture is optimal.

**Dynamic analysis is the only way to explore the solution space**

![Diagram](image)

**Figure 3: Solution Space Exploration. Courtesy of Siemens EDA**

Enabling this requires a consistent and interoperable method of modeling the system. That method must explicitly define functional, logical and physical element layers. It must also automate the transition of a sub-system for each element type’s design and verification flow. Finally, verifiable requirements must be allocatable to elements in system and subsystem models. With these requirements met, the system and subsystem models can be optimized analytically, and continuously verified through the realization cycle.
Early performance analysis provides insights into how the system will operate. This information should be used to perform early power analysis, thermal analysis and mechanical simulations.
Figure 5: High level power model simulation of nRF52832. Image courtesy Thrace Systems.

Flowchart

This is a design flow for an homogeneous ASIC/SoC that fulfills the requirements and functionality of the system-of-interest, expressed in the optimized architecture.
Figure 6: Homogeneous Product Design Process. Courtesy Siemens EDA

The advantages of chiplet based 3DHI options are becoming increasingly viable. Lithography reticle limits are driving disaggregation. Increased cost and volume flexibility will follow. A key advantage of SiP is heterogenous fab processes for the chiplet interoperation becomes a co-design option. Initially this enables portions of what was a PCB design to be collapsed into the SiP. As off-the-shelf chiplet availability expands, SiP will offer higher integration and become a preferred design option.
Here's a design flow for 3DHI packaging that will meet the requirements of the system-of-interest. Multiple concurrent chiplets need to be co-architected and co-designed. Although more complex, it offers compelling advantages.
Future work will go into the details of how EDA suppliers are providing methodology and tooling to realize this design process.

Input/output description

<table>
<thead>
<tr>
<th>Inputs</th>
<th>User Persona</th>
<th>Description/notes/comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Target System Requirements</td>
<td>SA</td>
<td>Source for the target system PPAC (performance, power, area, cost) requirements</td>
</tr>
<tr>
<td>Target system Concept of Operations (CONOPS)</td>
<td>SA</td>
<td>Operational information and use cases the target system must perform, which gives necessary context for the Target System Requirements.</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td><strong>Outputs</strong></td>
<td><strong>Supplier Persona[table]</strong></td>
<td><strong>Description/notes/comments</strong></td>
</tr>
<tr>
<td>Refined and Formalized Requirements Doc (pdf)</td>
<td>SA</td>
<td>Product specification defining high level functional, power, thermal, cost, reliability, and physical footprint requirements</td>
</tr>
<tr>
<td>Architectural Specification (pdf)</td>
<td>SA</td>
<td>System block diagrams of the functional, logical and physical architecture, including parameterization of the elements and interfaces.</td>
</tr>
<tr>
<td>Architecture Analysis (pdf)</td>
<td>SA</td>
<td>Results of the architecture analysis that illustrate the results of PPAC tradeoffs, and why the physical architecture described is expected to meet them.</td>
</tr>
</tbody>
</table>
Architectural Planning and Analysis workflow

Introduction

Traditional IC design scaling has been accomplished primarily through IC technology scaling. This process is referred to as design-technology co-optimization (DTCO). As the pace of IC technology scaling has dramatically diminished over the past decade, a new process named system technology co-optimization (STCO) is extending design scaling. STCO enables architectural and technology trade-offs early in the system design process to achieve high-performance, cost-effective solutions in a reduced timeframe. Predictive analysis is a fundamental component of STCO that leverages high-level modeling and analysis during the planning phase.

In defining new electronic systems and products, a system designer or design team defines the high-level product requirements and objectives. The system objectives or priorities may consider power, performance, physical formfactor, non-recurring engineering cost, unit cost, time-to-market or reliability and manufacturability. The electronic system is then partitioned into software and hardware. The hardware is further decomposed into board level components which may include a custom system on chip (SoC). With the advent of heterogeneous integration, the custom SoC may be further partitioned, or decomposed into multiple chiplets, integrated within a system in package (SiP) that deploys 2.5 and/or 3D packaging technologies. This is referred to as heterogeneous integration, or HI.

While some companies design at the bleeding edges of technology, others seek a LEGO® block approach to optimize the overall cost and schedule of their complex solutions. Both need to consider going the chiplet route as an additional design paradigm. To determine which microarchitecture best meets a product’s requirements and objectives, high-level predictive analysis is required to assess a multitude of different chiplet level SoC decompositions which we refer to as viable design scenarios. This high-level analysis is typically performed by the system or RTL design architect, but it also needs to consider packaging technologies, available ASIC IP, and off-the-shelf chiplet components. These considerations drive the need for collaboration between the system, RTL, package, ASIC, and test teams early in the system design planning process before the detailed implementation process begins.

This chapter will outline the key components of the Architectural Planning and Analysis workflow including the inputs and resultant outputs, the persona responsible for generating the inputs and outputs and the target users of the flow outputs. This section excludes the system design activities outlined in the executive summary including the input product specification documents defined in the table below.

A flow diagram will be provided to illustrate a typical Architectural Planning and Analysis workflow.
Product Specification Inputs

During the system design process, the System Architect(s) is responsible for defining the following product specifications for the overall system or subsystem. These inputs in conjunction with optional IP inputs will be used to create the outputs for the Architectural Planning and workflows.

<table>
<thead>
<tr>
<th>Inputs</th>
<th>Supplier Persona</th>
<th>Description/notes/comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requirements Doc (pdf)</td>
<td>SA</td>
<td>Product specification defining high level functional, power, thermal, cost, reliability, and physical footprint requirements</td>
</tr>
<tr>
<td>Architectural Specification (pdf)</td>
<td>SA</td>
<td>High level system block diagrams including data flow, register maps, boot up sequence and summary of external interfaces of the SiP</td>
</tr>
<tr>
<td>Verification Specification (pdf)</td>
<td>SA</td>
<td>Detailed documents defining functional verification requirements of the system. Includes detailed functional verification plan and methods</td>
</tr>
<tr>
<td>DFT Specification (pdf)</td>
<td>DA (DFT Architect)</td>
<td>Detailed documents defining system level usage of DFT techniques. Includes DFx and debug access and system DFT plans</td>
</tr>
<tr>
<td>Package Requirements Doc (pdf)</td>
<td>SA</td>
<td>Document summarizing high level package requirements including package type, target package dimensions, external ambient temperature, air flow, heatsink and target IP assignments</td>
</tr>
</tbody>
</table>
Intellectual Property (IP) Inputs

Varios SOC/Chiplet IP is used throughout the various workflows. Those applicable to the Architectural Planning workflows are summarized in the following table:

<table>
<thead>
<tr>
<th>Inputs</th>
<th>Supplier Persona</th>
<th>Description/notes/comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>C2C IP</td>
<td>Third party IP Provider</td>
<td>IC design Intellectual Property (IP) for external, Chip-to-Chip IO and high-speed interfaces targeted for the integration within an ASIC or custom chiplet device.</td>
</tr>
<tr>
<td>D2D IP</td>
<td>Third party IP Provider</td>
<td>IC design Intellectual Property (IP) for internal, high-speed Die-Die (D2D) interfaces targeted for the integration within an ASIC or custom chiplet device.</td>
</tr>
<tr>
<td>Chiplet CDK</td>
<td>Internal or third party Chiplet Providers</td>
<td>Design models as defined in the CDX “Proposed Standardization of Chiplet Models for Heterogeneous Integration” white paper</td>
</tr>
</tbody>
</table>

Output descriptions

The product specification inputs in conjunction with optional IP inputs will be used to create the outputs for the Architectural Planning and workflows.

<table>
<thead>
<tr>
<th>Outputs</th>
<th>User Persona</th>
<th>Description/notes/comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Micro-architectural Specification (pdf)</td>
<td>SA, SND</td>
<td>Internal block diagrams of SOC and/or SiP. Summary of external interfaces and internal SiP Die-to-Die (D2D) interfaces including protocols, configuration, and performance requirements</td>
</tr>
</tbody>
</table>
Architectural Planning Workflows

Micro-architectural Specification

Assuming a custom SOC is required within an electronic design, a heterogeneous solution should first be considered as a viable implementation strategy. A monolithic/homogeneous ASIC design will often lead to the highest performance, lowest power, and lowest unit cost, but the additional product requirements such as NRE cost, design resources, schedule and form factor may warrant the need to consider alternative heterogeneous implementations. Assuming a homogeneous implementation is not feasible and/or does not adequately meet the key project requirements, the SOC design should be further decomposed into 2 or more chiplet components to be interconnected within a package. This decomposition would generally be driven by the System Architect/Designer (SA). The first step in this decomposition process is to take the Architectural Specification and refine this into more detailed block diagrams that partition the design into multiple chiplet components or sub-blocks. At this level, it is very difficult to assess the merits of a particular partitioned SiP design, so multiple micro-architectural scenarios should be considered. During the exploratory process, a detailed micro-architectural spec is not required, but once a suitable configuration is determined a more detailed specification may be considered. These high level micro-architectural specifications should include the physical partitioning of the SOC blocks into chiplet components and include the internal (to package) die-to-die (D2D) chiplet to chiplet interfaces and external SiP interfaces. These chiplet blocks can be further refined into detailed ASIC specifications for implementation as custom ASIC-chiplet devices. Alternatively, as an ecosystem of commercially available chiplet components become available in the market, those commercial, off-the-shelf chiplet components can be used directly as the SOC blocks within the SiP device. These micro-architectural specs can be used in creating chiplet and SiP level RTL structural models for each of the design scenarios.

SiP RTL Design

Once a set of viable design scenarios have been defined, the System Architect (SA) may develop more detailed micro-architectural specifications for each chiplet within each respective design scenario. Using

<table>
<thead>
<tr>
<th>SiP RTL (.V)</th>
<th>SND</th>
<th>Structural RTL including instantiations of SOC/Chiplet components of the SiP device, including D2D and external package IO interconnect</th>
</tr>
</thead>
</table>
these micro-architectural specifications, the System Architect (SA) and/or Structural Netlist Designer (SND) will capture a high-level structural description of each scenario in System Verilog (SV). If existing chiplets are available, the Verilog model from the associated chiplet CDK could be used for each respective, off-the-shelf chiplet component in the design. If a new or modified chiplet is to be considered for one or more of the chiplet components, a structural model for each custom chiplet is created for each design scenario. These chiplet models should instantiate critical top-level IO. The minimum IO requirement for the chiplet structural models necessary to support early STCO analysis should include the internal D2D interfaces and external C2C, high speed interfaces such as DDR, PCI, ect. These interfaces may be defined using simple IO declarations or leverage available generic or vendor/technology specific IP models which are instantiated into the respective chiplet RTL model. The level of completeness, structure and functionality of these models may vary throughout the design implementation process and by the SiP design team’s design process. If SoC IP for the high speed Internal/External interfaces are to be included in the design, the ASIC Physical Design Lead (APLD) may be included in specifying the target SOC IP.

Once each of the chiplet models are created, a top-level SV model is created declaring the top-level IO pins (again minimum requirement would be to include the top-level, high-speed interfaces) and instantiating each of the chiplet components. The model should also define the connectivity from the top-level IO pins to the respective chiplet instance pins as well as the chiplet-to-chiplet interconnect. A more detailed summary of the chiplet models required for SiP Level functional simulation is addressed in the ‘Functional Simulation and Verification workflow’ section of this document.

Once a structural model for each of the SiP Level Scenarios is created, predictive modeling can be performed on each of the scenarios to home in on an optimal micro-architecture that best meets the overall design objectives.

Architectural Analysis Workflows

Predictive Analysis Overview

Once a viable set of SiP level design scenarios have been established, each scenario should be evaluated to determine which scenario is best suited to achieve the system level requirements. The level of detail and types of analysis required will vary from product to product and company to company. In the following sections we will provide a brief overview of the different predictive analysis workflows that might be considered during the architectural planning phase of the product development process. The objective of this analysis is to identify a design micro-architecture that best meets the project requirements. Once the optimal micro-architecture is identified that design would proceed into the more detailed Functional Simulation and Verification and Physical Planning workflows defined in the following two chapters. A more detailed summary of these predictive analysis workflows in addition to the in-process and signoff analysis
will be provided in the respective workflow sections of this document. These workflows include SiP Floor planning/Routeability, Predictive Power, Thermal, Mechanical stress analysis as well as optional D2D Datapath Planning and Signal Integrity analysis. Additionally, the planning of the power delivery network (PDN) which is required to provide sufficient power to all the active components within the SiP design as well as the SiP level design for testability (DFT) strategy and methods require careful planning during the architectural planning phase of SiP designs.

**SiP Floor planning**

Once the SiP level RTL is defined for each design scenario, the placement of the individual chiplets should be planned. The chiplet placements or floor plan needs to consider both the external top level SiP IO pin placement requirements as well as the internal D2D connections. If a custom chiplet is included, the internal and external IO interface macros need to be placed and aligned within the chiplet in conjunction with the other adjacent chiplet IO macros for each of the respective design scenarios. This preliminary IO/chiplet placement should be done before the subsequent predictive analysis is performed. Iterations of the IO/chiplet placement will likely be required for each design scenario to identify an optimal placement to best meet the specific design requirements.

**Predictive Power Analysis**

Once target design scenarios are defined and preliminary floor plans are created, a preliminary power estimate should be performed for each scenario. This will require power models for each of the composite chiplet components. If existing chiplets are being deployed, the power model from the respective chiplet can be used. If a new chiplet(s) is to be included in the design scenario, a power model should be created. The level of detail, accuracy, and number of operational modes is left to the discretion of the designer. Once power models are available for each of the chiplets, a high-level power estimate for each design scenario can be performed. The methods in creating these power models and SiP level power estimates are defined in the power section of this document.

**Predictive Thermal Analysis**

Performing a thermal analysis for a SiP design requires thermal models for each of the composite chiplet components. If existing chiplets are being deployed, the thermal model from the respective chiplet can be used. If a new chiplet(s) is to be included in the design scenario, a thermal model needs to be created. The level of detail, accuracy and granularity is left to the discretion of the designer. An estimated or predictive thermal model can be created using the associated power models for the composite chiplets, chiplet/block dimensions, material properties and physical representation of the chiplets bump pins. If target IP power models are available for IP within the chiplet and/or if chiplet sub-block level power models are available, a higher fidelity power model can be created.
A SiP level thermal simulation can be performed using the chiplet thermal models, the SiP floorplan placement information and the material and physical dimension/placement of the package substrates, interposers and associated pin locations along with the mechanical package, ambient temperature target and target cooling mechanisms and constraints. Simulations for each of the design scenarios are run to access the thermal performance characteristics of each candidate design. A more detailed summary of the predictive thermal analysis workflow is provided in the Thermal Modeling and Analysis workflow section of this document.

Predictive Mechanical Stress Analysis

Mechanical warpage is an important consideration for package designs that include large die, interposers and/or substrates. Different thermal expansion characteristics of the different package components and materials can create mechanical strain on the metal interconnect between the component pins during the package assembly process which can result in immediate connectivity defects or longer-term reliability concerns. Traditional monolithic designs have often relied on designing, manufacturing, and testing representative test vehicles to assess the reliability of a target design. Due to the large number of design scenarios and complex structures that are varying throughout the package design process, it is no longer practical to design and evaluate a representative test vehicle in a reasonable time frame for HI designs. Although it is difficult to perform accurate mechanical simulations during the pre-layout design during the architectural planning phase, it is important to run early, predictive mechanical stress simulations to eliminate and gross connectivity reliability concerns of any candidate design scenario before proceeding to the implementation phase. More detailed mechanical simulations and possible test vehicles are required once a more complete design database is available to ensure a reliable package design. A more detailed summary of the predictive and detailed mechanical analysis workflows is provided in the Mechanical Analysis workflow section of this document.

D2D Datapath Planning/SI Analysis

The high-speed Die-to-Die (D2D) interface between chiplets generally operate at speeds too fast to analyze using traditional ASIC Static Timing Analysis (STA) models and workflows may be requiring the high speed Signal Integrity analysis tools and workflows used for PCB and package level analysis for high speed interfaces. This analysis requires IBIS models for the transceivers, a detailed channel model of the interconnect and a simulation test bench. It is important to consider these interfaces when selecting packaging technologies, trace widths and spacings and shielding strategies. A detailed analysis requires a detailed layout and parasitic extraction of the D2D interconnect channel. During the architectural planning phase, a floor plan exists which includes the physical placement of the D2D physical interfaces (PHYs) within the respective chiplets including the pin locations. Although a detailed channel route is not completed at this phase, the placement and pin information can be used to generate some predictive
trace routing of the channel which can be used to conduct some predictive Signal Integrity (SI) analysis. Different trace widths, spacings and materials can be evaluated to home in on an optimal package technology and associated channel routing strategy. More detailed SI analysis is required once a more complete design database is available to ensure the target speed and signal integrity requirements can be achieved. A more detailed summary of the predictive and detailed D2D SI analysis workflows is provided in the Signal Integrity and STA workflow section of this document.

Power Delivery Network (PDN)

During the Architectural Planning and Analysis phase of the SiP design process, a team comprised of system, power supply, power integrity and package engineers will review the system requirements, C2C/D2D IP specifications and integration guidelines to define an optimal power connectivity strategy which may include power supplies, voltage regulator monitors (VRM) and power monitor ICs (PMIC). The power delivery network (PDN) will span from the power supply, which is generally on the PCB, the package interposer/substrate to the chiplet die which could include VRM/PMIC chiplets. The PDN is designed to provide sufficient power to all the active components in the design to ensure that the system and IP power integrity requirements are met.

The design intent of this PDN can be defined using the IEEE Unified Power Format (UPF). The UPF file is then used to define the physical PDN netlist that will be implemented during the package planning phase of the design as defined in the physical Planning workflow section of this document. Analysis of the PDN will be performed once a package layout is available using the power integrity (PI) analysis workflows defined in further detail in the 3D IC analysis workflow section of this document.

System DFT Specification workflow

Chiplet based SiPs can benefit from the system level Design for Testability (DFT) specification and usage in system bring up just as any custom SOC required within an electronic design. DFT logic and test IO are required in each of the chiplets to facilitate system level and Automated Test Equipment (ATE) testing. This adds value in bring-up environments specially during booting operating system on the SIP and running application in system environment. The DFT Architect defines and documents the architectural features and sequence to access these features. For example, Logic BIST (Built-In Self-Test) can be triggered by the system controller using JTAG access. A more detailed summary of the DFT workflows is provided in the DFT, IP, ATE Test workflow section of this document.

A summary of the Architectural Planning and Analysis workflow is summarized in the diagram below.
Figure 9: Architectural Planning and Analysis workflow. Courtesy Siemens EDA
Functional Simulation and Verification workflow

Introduction

Although a lot of design work can be done formally from the software level, functional testing is always required to cover design aspects that cannot be guaranteed correct-by-construction, that includes timing and power issues that are subject to variability in semiconductor manufacturing, and validating that test patterns provide appropriate coverage and limit escapes. Likewise, combining a number of chips into a system adds new problems created by interactions that are not foreseen in designing the individual chips, like noise from power supplies and signal crosstalk in communication.

Purpose of flow

Verification that the physical design will behave as intended. For our purposes at ODSA, we want verification flows that confirm that the ICs will work together as expected rather than that the individual ICs perform correctly. ICs are designed somewhat independent of their use, so issues like thermal and power management are often not considered fully at that level, whereas they are important in an assembled Chiplet system.

In addition to making sure the designs fall within the environmental limits, verification is required to check that defective systems can be recognized, since parts may fail in assembly or while working. That requires propagating information used at IC-test up to system-test.

At OCP we focus on large software (digital) systems, and how those are allocated into hardware, the components mostly communicate through serial links and parallel data buses, so a lot of the verification is looking at the integrity of that communication, and adequacy of power supplies and cooling. Given the scale of the systems, behavioral modeling or virtualization needs to be used for the computational elements, and “transaction level” methodologies for communication modeling. HiL (hardware in the loop) is also an option when dealing with Chiplets that already exist.

Methodology with Chiplet Systems

Since EDA companies have mostly focused on chip design in the last three decades, there are not many standards that apply to assembling and simulating (Chiplet) systems in their ecosystem, however the issues of a Chiplet system are much the same as for a system built from boards and modules. Systems used in vehicles (aircraft and cars) require validation for standards of air-worthiness and safety, so EDA companies have been developing more tools for things like ADAS [6][7][8]. Key to validating those systems is using HiL, and applying a HiL approach to validation of interposer based Chiplet systems, will require leveraging technology like custom hardware configuration which has been proposed with the HPC-Module in the OCP HPC subgroup. I.e. Chiplets available for use on interposers should also be available for rapid prototyping in non-custom forms - PCIe cards, M.2 modules, SATA drives.
Simulating analog systems at scale has long been a problem, and has been addressed by tools like Xyce (designed for parallel simulation on server farms), but even using IP level behavioral modeling such tools are likely too slow for validating systems in reasonable time. Fortunately, it is possible to use AI techniques to "digitally twin" analog behavior into neural networks and use neural network accelerators to get more speed.

Standards for mixing analog and digital behavior in simulation are currently being worked on at the IEEE as part of the P1800 SystemVerilog effort, with the target of co-simulating SystemVerilog with analog and behavioral simulators like Xyce and SystemC-AMS, and it is expected that a lot of the initial design of chiplet systems will be a mix of SystemC-TLM, SystemC-AMS and regular C++ software. P1800 probably won't deliver Chiplet modeling solutions until 2028, but a parallel effort has been launched at Accellera to create a standard for "federated simulation". Chiplet systems are intended to be nearer optimal ways to implement particular software algorithms, rather than general purpose hardware, so can be engineered with correct-by-construction approaches. The time to verify a complex system using functional simulation (alone) scales exponentially with the size of the system and the amount of software being run, it is advisable to fully debug software before trying to run any of it on a purely simulated platform, and to use HiL and virtualization as much as possible, e.g. if developing ARM based products, use ARM server boards with virtual processors (a time-sliced real processor), with simulation for only those components not available through HiL.

Defect and Fault Coverage

While Chiplets may all arrive 100% working, the process of making the system may damage them and the extra system components may have their own defects. Test patterns and corresponding models should be supplied with the Chiplets to enable developing system tests. Defects do not necessarily cause faults, and some hardware is fault tolerant, functional simulation may be used to explore the defect sensitivity of a design, and minimize the probability of failure. Given that most IP will be delivered as "black box", models of the IP in likely failed states are useful to have. Likewise a system will fail eventually in use, and being able to model with the mean times to a Chiplet dropping into a failed state, helps optimize system lifetime, MTTF information can be combined with the various models provided. Temperature sensitive models are
particularly important for co-design with cooling, and MTTF prediction is useful in optimizing the cooling of the system.

BER/CDC

As mentioned elsewhere, a lot of communication in Chiplet systems will be high speed SERDES, which naturally has a particular bit-error-rate due to noise, and potentially more errors when crossing clock domains. At a system simulation level one would use simple channel models and just inject occasional errors (as per BER spec) for validation. For better accuracy the Chiplet interfaces should be simulated with the actual channel models at a SPICE level (with Monte Carlo) to generate the simple models. Transistor driver/receiver models or equally accurate behavioral descriptions (Verilog-AMS) can be used, and are covered below in the D2D communication section.

Normal functional (logic) simulation cannot catch CDC errors; most CDC tools use static analysis, however that can provide assertions to be used in functional simulation. Statistical models that cover the timing spread can be used in functional simulation, but neither approach is particularly fast. For system level it is best to just capture the minimum information needed to recreate scenarios (in more detail) and use the assertions to pick the likely candidates for failure of signal integrity, those scenarios can be analyzed more thoroughly in isolation.

In the interests of power optimization, SERDES may be run at lower Voltage such that it operates close to an acceptable BER limit for the particular application.

Simulation and Verification Flow

In the chiplet-based design flow, the interconnects between the chiplets pose several new significant challenges. The interconnect needs to be simulated and verified to ensure that it delivers the necessary connectivity and performance. Here, the functional simulation is the process of testing the functionality of the system by executing a set of inputs and comparing the output with the expected results. This helps to validate the functionality of the system at the architectural level and find any bugs or issues before the hardware is built. While in the conventional SIPs the number of wires in these interconnects was manageable such that schematic reviews and netlist reviews were sufficient. With high-pin count chiplets, such verification needs to be fully automated. While circuit extraction tools are available for doing this work within an IC, the analysis in a Chiplet system requires that work to be redone using “field solver” software, since there will be unpredictable interactions between packaging (lead frames) and interposer tracks. Information about the interactions between the wires can be described in languages like Verilog-AMS and back-annotated into system simulations. There are currently no standards for back-annotating analog effects into IC models, so that analysis is often skipped and more work is done in error recovery and filtering to make the IC more robust.

The verification ensures the system meets the specified requirements and functions as intended. It typically involves functional simulation, hardware-assisted verification using prototypes, and formal
verification. In the simulation, the system is modeled using a hardware description language (HDL) such as Verilog. Its behavior is tested using testbenches. Prototyping involves fabricating a physical prototype of the chip on a small scale (multi-project wafers), or FPGA.

The goal of simulation and verification is to catch errors early in the design process and prevent costly redesigns and re-spins of chips. The functional simulation and verification steps are described in the following workflow. The further down the flow you are the more expensive it is to find and fix bugs, and the effort is exponential with the size of the system, so it is desirable to use pre-qualified chiplets, and correct-by-construction techniques.

**Figure 10: Functional simulation and verification flow. Image courtesy of Palo Alto Electron**

**Functional simulation and verification plan**

For the success of a chiplet-based design, a comprehensive functional simulation and verification plan is crucial. The simulation and verification plan will make sure that the chiplet-based design meets its desired specifications and is ready for fabrication and assembly. The plan should include:

- Requirements: Outline the specific functional requirements and objectives for the simulation and verification process, including the desired functionalities, performance targets, and interface.
• Methodologies: Specify the verification methodologies to be used, such as simulation, and physical testing.

• Schedule: Establish a timeline for the verification process. Identify all the verification steps that need to be carried out. Determine the dependencies between the verification steps and the order in which they need to be performed, such as the chiplet-level verification that should be done before the system-level verification.

• Environment and tools: Identify the tools and environment required for the verification process, including hardware and software components.

Chiplet-level verification

The Chiplet-level verification aims to ensure that each individual chiplet meets the required specifications and functions correctly. Each individual chiplet is simulated and verified individually. This can be done using various verification techniques such as functional simulation, testbenches, assertions, and formal verification. The functional simulation involves running simulations to verify that the chiplet behaves as expected for a given input. The testbench includes test cases that verify the chiplet's behavior for a range of inputs. Assertions and formal verification methods are used to verify the correctness of the design and to find any design errors. The steps for chiplet-level verification include:

• Chiplet-level models: For each chiplet, a model must be developed that accurately represents its behavior and interfaces. These models will be used in the simulation process to test each chiplet's performance. This can be done using an HDL such as Verilog and RTL.

• Testbench: A testbench is a simulation environment used to verify the behavior of the chiplet. The testbench includes test cases that exercise the chiplet's functionality and verify its behavior for a range of inputs. This also includes the test coverage.

• Simulation and verification: This helps to identify any issues with the chiplet's behavior and to confirm that it behaves as expected.
  ▪ Functional simulation and verification: Using the chiplet models and the testbench, run the simulations for each chiplet to verify its behavior for a given input, including its interactions with other chiplets in the system.
  ▪ Performance verification: This involves performing performance verification on each chiplet to validate its performance and ensure that it meets the desired performance goals.

• Formal verification or equivalent check: If formal verification is being used, mathematical algorithms are used to prove the correctness of the design. Semi-formal methods include the use of SAT-solvers and symbolic simulation. These methods prove functional correctness of a design, but do not verify the physical design is free from timing and power related bugs.
Debug and fix issues: If any issues are found during the functional simulation or formal verification, they are debugged and corrected. This may involve modifying the design, changing the testbench, or adjusting the assertions.

System-level simulation and verification

Once the individual chiplets have been verified, they are integrated to form the complete chip. A system-level simulation is then performed to verify the functional behavior of the entire system. This involves simulating the interactions between the different chiplets and ensuring that they work together as expected. The steps involved in system-level simulation and verification are

- **System-level model**: A system-level model is created that represents the interactions between the chiplets and the system-level interfaces. This model can be used for simulation and verification. This model can be created using a variety of tools and methods, including simulation tools, high-level models, and behavioral descriptions. This involves
  - Abstraction level: The level of abstraction used in the system-level model depends on the design requirements and the desired level of detail. In general, the system-level model should provide enough detail to capture the behavior of the entire system, but not so much detail that it becomes cumbersome or impractical to use.
  - Modeling language: The system-level model can be created using a hardware description language (HDL) such as Verilog or VHDL, or using high-level models in SystemC or Matlab/Simulink.
  - Modeling components: The system-level model should include models of all the components in the system, including the individual chiplets and any other components that are part of the system. This can re-use the chiplet models from the chiplet-level verification.
  - Modeling interfaces: The interfaces between the components in the system are a critical part of the system-level model. The model should define the interfaces between the chiplets and the rest of the system, as well as the interfaces between the chiplets themselves. This includes the communication protocols, data formats, and any other aspects of the interfaces that are relevant to the behavior of the system.
  - Modeling constraints: The system-level model should also include any constraints on the system, such as performance requirements, power requirements, and thermal requirements. These constraints should be defined in a manner that allows them to be easily tested and verified using simulation and verification tools.

- **System-level testbench**: A system-level testbench is created to verify the interactions between the chiplets. This testbench includes test cases that exercise the complete chip design and verify its behavior for a range of inputs under different scenarios and operating conditions. It involves
○ Input data: These input data should be carefully chosen to represent the range of conditions and scenarios that the system is expected to encounter in its intended operating environment. For example, the data may include different voltage levels, temperature ranges, or input signals.

○ Test Cases and coverage: The test cases should be designed to cover all relevant scenarios and conditions, and should be run against the system-level model and the testbench.

○ Output Analysis: The testbench should be designed to analyze the outputs of the system-level model and to compare them to the expected results. Any differences between the actual and expected outputs can indicate issues or problems with the system and should be analyzed and debugged as necessary.

• System-level simulation and verification: Functional verification involves executing the test cases against the system-level model and the testbench and comparing the results to the expected outputs. Any differences between the actual and expected outputs can indicate issues or problems with the system. This helps to identify any issues with the interactions between the chiplets and to confirm that the complete chip behaves as expected. This involves

○ Functional verification: This verifies that the system can perform the desired functions. Some examples are:

  ■ Boot up: Check boot up sequence using One-Time-Programmable (OTP) image
  ■ Routes checks: Checks if all connections drive their endpoints and no other interference occurs.
  ■ Drivers checks: Verifies if nets are unintentionally driven by multiple drivers at the same time.
  ■ IOs checks: Used IOs must be connected to the intended signal. Unused IOs must show their disconnected status (high z).
  ■ Power supply checks: Check if IOs are supplied with power as intended and have no unintended short

○ Inter-chiplet communication and protocol verification: This verifies that the chiplets can communicate with each other and exchange data and signals correctly. This also verifies that the system complies with industry standards and protocols, such as UCIe, BoW, USB, Ethernet, etc.

○ Error-handing verification: This verifies that the system can detect and handle errors correctly, such as memory errors, communication errors, etc.

○ Performance verification: This verifies that the system meets the desired performance goals, such as processing speed, memory bandwidth, etc. For example, performance can be modeled using netlist. The resistants are modeled in software delays based on the
interconnect protocol, materials, and structure. Large input data are streamed in and the output data timing is gauged.

- Stress verification: This verifies that the system behaves correctly under extreme conditions, such as high temperatures, high voltage, etc.

- Formal verification: If formal verification is being used, mathematical algorithms are used to prove the correctness of the complete design.

- Debug and fix issues: If any issues or problems are identified during functional verification, debugging tools and techniques should be used to identify the cause of the problem and correct it. This can include using trace files, waveform viewers, and code debugging tools.

Hardware Validation

In this step, a physical prototype of the chip is fabricated and tested to validate its performance and functionality in real-world conditions. This process helps to ensure that the chip design is working correctly and that it meets the desired specifications before mass production begins. The steps involved in this process are as follows:

- Chip prototypes: The chip prototypes are fabricated using a small-scale manufacturing process, such as maskless lithography/Multi-Project Wafer (MPW) or FPGA is used to allow for fast turnaround times and low-cost prototypes.

- Test board/Socket Board: The prototypes are assembled onto a printed circuit board (PCB) to create a complete system. This allows for testing and validation of the complete chip design.

- Environment, software, and drivers: The software and drivers are installed onto the prototypes to enable testing and validation of the complete system.

- System-level verifications: Functional tests are performed to verify the functionality of the prototypes. This includes verifying all functional blocks within the chip design.

  - Power-on and reset: Initial power-on tests are performed to verify that the prototypes are functioning correctly and to check for any issues, such as power-up failures or memory errors.

  - Functional verification: This verifies that the system can perform functions, such as booting up, loading software, and performing calculations.

  - Inter-chiplet communication and protocol verification: This verifies that the chiplets can communicate with each other and exchange data and signals correctly.

  - Performance verification: This verifies that the system meets the desired performance goals, such as processing speed, memory bandwidth, etc.

  - Stress verification: This verifies that the system behaves correctly under extreme conditions, such as high temperatures, high voltage, etc.
- Debug and fix issues: If any issues are found during the functional testing, they are debugged and corrected. This may involve modifying the design, changing the testbench, or adjusting the software and drivers.

Tape-out

This is the final step in the functional simulation and verification process. This involves preparing the design for manufacturing and producing the physical chips according to the rule decks provided by the manufacturer. The steps involved in this stage are as follows:

- Design data for manufacturing: The design data is prepared for manufacturing by creating production-ready data such as GDSII (GDS2). This data is used to manufacture the chips.
- Design Rule Check (DRC): Design rule checking is performed to ensure that the design meets the design rules and constraints set by the manufacturer in the form of rule decks. This is an important step to ensure that the design is manufacturable and that the chips will be of good quality.
- Layout Versus Schematic (LVS) verification: Layout versus schematic verification is performed to ensure that the design data matches the schematic. This helps to identify any errors or discrepancies between the design and the schematic.
- Design For Manufacturing (DFM) and Critical Area Analysis (CAA): Critical area analysis is performed to evaluate the areas of the design that are critical for manufacturing. This information is used to determine the process margins and to optimize the DFM.

Input/output description

<table>
<thead>
<tr>
<th>Inputs</th>
<th>Supplier Persona [table]</th>
<th>Description/notes/comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>CDXML [3] [4], chiplet</td>
<td>SA, APLD, ALD, V</td>
<td>Chiplet data: mechanical, electrical, IO, power, thermal, datasheet data, etc. in XML format.</td>
</tr>
<tr>
<td>datasheet</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Netlists</td>
<td>SA, APLD, ALD</td>
<td>Functional equivalence checks such as RTL and gate-level netlists</td>
</tr>
<tr>
<td>Functional requirements</td>
<td>SA, APLD, ALD</td>
<td>Outline the specific functional requirements and objectives for the simulation and verification process, including the desired functionalities, performance targets, and</td>
</tr>
</tbody>
</table>
interface.

<table>
<thead>
<tr>
<th>Simulation environment and tools</th>
<th>SA, SVE</th>
<th>Identify the tools and environment required for the verification process, including hardware and software components.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chiplet-level models</td>
<td>APLD, ALD, V</td>
<td>This can be done using an HDL such as Verilog and RTL.</td>
</tr>
<tr>
<td>System-level model</td>
<td>SA</td>
<td>A system-level model is created that represents the interactions between the chiplets and the system-level interfaces. This model can be used for simulation and verification. This model can be created using a variety of tools and methods, including simulation tools, high-level models, and behavioral descriptions.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Outputs</th>
<th>User Persona[</th>
<th>Description/notes/comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simulation environment or testbench</td>
<td>SVE</td>
<td>A simulation environment used to verify the behavior of the chiplet. The testbench includes test cases that exercise the chiplet’s functionality and verify its behavior for a range of inputs.</td>
</tr>
</tbody>
</table>
| Simulation and verification results | SVE | - Functional simulation and verification results: Using the chiplet models and the testbench, run the simulations for each chiplet to verify its behavior for a given input, including its interactions with other chiplets in the system.  
- Performance verification results: This involves performing performance verification on each chiplet to validate its performance. |
<table>
<thead>
<tr>
<th>Hardware validation results</th>
<th>SA, SVE</th>
<th>Results from the chip prototypes, test board/socket board, FPGA, Emulator, and etc.</th>
</tr>
</thead>
</table>

and ensure that it meets the desired performance goals.
Physical Planning workflow

Introduction

Modern packaging technologies across the industry offer a variety of 2D, 2.5D, and 3D solutions for heterogeneous integration of chiplets to form a product. Choosing the right integration solution will have a significant impact on cost, schedule, and ability to meet performance, functional, and use conditions for the product. A product architecture may consider trade-offs between silicon interposer, organic interposer, substrate, embedded die, die stacking, fan-out technologies, and PCB technologies. The trade-offs may also include combinations of the above technologies and each technology and associated generation may have different design rules and hence rapid prototyping and upfront assessment of critical XYZ parameters will be necessary during the technology selection, and product definition phase to ensure both (i) viability of the solution for a given product, and (ii) choosing the best solution to meet product requirements.

Physical design lead and the package architect must consider placement of the IP on the silicon device that will affect the ability to route through and out of the package depending on where the chiplet is placed on the package. Arrangement of the IO/PHY and bump out needs to be assessed to understand the impact to shoreline, number of routing layers required on the package. A package architect will take the individual die size, total die area, BGA pitch/technology, and assembly solutions to understand the thermal and mechanical performance. Package BGA pitch, pattern, and the arrangement of the signals will affect the board technology selection, and routing feasibility.

Physical Planning Background

Physical planning and design activity typically ensues once the early architecture reaches maturity. At this stage all of the collaterals have been identified and the structural floorplans are mostly clear. The planning phase of physical design primarily identifies how to supply power and clock such that power supply is robust, timing jitters are within control, placement and routing can be achieved in a given area. Once the top skeleton of the circuit is clear then design automation tools can be used to place all the IP blocks and gate only then the place and route ensues followed by analysis of the physical design to achieve architectural targets. There are remarkable differences between how physical planning and design has been done for silicon IC single chip as well as the IC package.

Conventional IC Physical Planning and Design
Physical planning and design of silicon integrated circuits is well understood. Following flowchart outlines a general flow used for that purpose. Typically there is planning phase to determine block boundaries, make a floorplan, IO placement, determine power and ground meshes, and top level clock distribution. Note that most of these physical flows are performance driven and there are constraints in place to time the between sequential elements for certain clock frequency. When the design progresses through various stages of the physical design, typically timing analysis is done at various levels to achieve performance. More detailed flows take into account effects of temperature, voltage, and process variabilities into account. After the planning stages, constraints are typically clear and the next stages include placement and routing which is the heart of chip design today. Placement at global and local level gets done mostly using automated tools followed by routing at global and local level. Along the way, a detailed clock tree gets synthesized. Finally the design goes through a comprehensive sign-off analysis. This physical design process includes many iterations and involves multiple teams during the performance optimization and verification steps. We expect that the workflows to design chiplets will be mostly identical to conventional chip design. The differences will come into picture at the stage of chiplet integration that we will discuss later.

<table>
<thead>
<tr>
<th>Physical Planning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Floorplan</td>
</tr>
<tr>
<td>Optimization</td>
</tr>
<tr>
<td>IO Assignment</td>
</tr>
<tr>
<td>Power/GND</td>
</tr>
<tr>
<td>Clock Planning</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Cell/IP Placement and Synthesis</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global Placement</td>
</tr>
<tr>
<td>Detailed Placement</td>
</tr>
<tr>
<td>Physical Synthesis</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Routing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Routing Constraints</td>
</tr>
<tr>
<td>Clock Tree Synthesis</td>
</tr>
<tr>
<td>Global Routing</td>
</tr>
<tr>
<td>Local Routing</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Sign Off</th>
</tr>
</thead>
<tbody>
<tr>
<td>Timing Closure</td>
</tr>
<tr>
<td>Physical Validation (LVS/DRC/ERC)</td>
</tr>
<tr>
<td>PI/PI/EMIR/Noise</td>
</tr>
</tbody>
</table>

**Figure 11:** Conventional IC Physical Planning and Design flow. image courtesy of Palo Alto Electron

This work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International
Physical Planning for Chiplet Integration

Physical planning for chiplet integration is a critical aspect of chiplet-based system design when incorporating multiple chiplets or IPs (Intellectual Properties) onto a single package or die. This process involves organizing, connecting, and optimizing the placement of these chiplets to ensure the overall functionality, performance, and reliability of the integrated system. Here's an overview of the key considerations and steps involved in physical planning for chiplet integration:

- **System Architecture Design:**
  - Define the overall system architecture, including the roles and functions of each chiplet or IP in the system.
  - Establish communication interfaces and protocols between chiplets, including buses, memory channels, and high-speed links.

- **Chiplet Selection:**
  - Choose the appropriate chiplets based on their functionality, performance, and compatibility with the system's requirements.
  - Ensure that chiplets are available in the required packaging formats (e.g., 2.5D interposer, 3D stacking, Multi-Chip Module).

- **Package Selection:**
  - Select the packaging technology that best suits the integration requirements, such as 2.5D or 3D packaging, flip-chip, or fan-out wafer-level packaging (FOWLP).
  - Consider factors like thermal management, signal integrity, and power distribution.

- **Thermal Analysis:**
  - Perform thermal simulations and analysis to determine heat dissipation profiles and ensure that the integrated chiplets do not overheat.
  - Implement thermal management solutions like heat spreaders, heat sinks, or thermal vias as needed.

- **Power Delivery Network (PDN) Design:**
  - Design the power delivery network to ensure that each chiplet receives stable and sufficient power.
  - Consider power gating, voltage domains, and power sequencing to manage power consumption efficiently.

- **Signal Integrity Analysis:**
  - Perform signal integrity simulations to analyze the behavior of high-speed data links and ensure that signals meet timing and noise requirements.
  - Implement measures such as signal equalization and error correction if needed.

- **Package-Level Routing:**
  - Plan and optimize the routing of electrical connections (traces) between chiplets, taking into account signal integrity, power distribution, and thermal constraints.
  - Use advanced packaging technologies to route high-speed interfaces efficiently.

- **Package Manufacturing and Assembly:**
○ Collaborate with packaging and assembly vendors to manufacture and assemble the package, ensuring that all chiplets are correctly placed and interconnected.
○ Validate the assembly process through testing and inspection.

- Chiplet and package interfaces:
  ○ Including interfaces consideration such as UCIe and BoW in the physical planning process ensures that advanced interconnect technologies are effectively integrated into the overall chiplet integration strategy. These interfaces enable high-speed communication and efficient data transfer between chiplets, contributing to the overall performance and functionality of the integrated system.
  ○ Ensure the interfaces align with the defined communication protocols and requirements between chiplets.
  ○ Determine the specific requirements, such as the number of wires and data rates, and incorporate them into the package design.
  ○ Optimize the placement and routing of BoW interfaces to meet signal integrity and power distribution needs.

- Testing and Validation:
  ○ Develop comprehensive test plans to verify the functionality and performance of the integrated chiplets.
  ○ Perform electrical and functional tests to ensure that each chiplet operates as expected.

- Reliability and Quality Assurance:
  ○ Conduct reliability assessments, including stress testing and failure mode analysis, to identify potential weak points in the integrated package.
  ○ Implement design and process improvements to enhance reliability.

- Software Integration:
  ○ Develop or modify device drivers and firmware to support the integrated chiplets.
  ○ Ensure that the software stack is optimized for the new system configuration.

Physical planning for chiplet integration requires close collaboration between hardware and packaging engineers, as well as thorough analysis and testing to ensure the successful integration of chiplets into a cohesive and functional system. It is a complex process that plays a crucial role in modern semiconductor design, enabling the creation of high-performance and energy-efficient chiplet-based systems.
Figure 12: Structural netlist diagram of 2 chiplets connected with BoW diagram. Image courtesy Thrace Systems

Input/output description

<table>
<thead>
<tr>
<th>Inputs</th>
<th>Supplier</th>
<th>Description/notes/comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>System requirements</td>
<td>SA</td>
<td>-Specifications and requirements for the overall system, including performance, power consumption, and form factor.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-Detailed architectural design, specifying the functions and interfaces of each chiplet.</td>
</tr>
<tr>
<td>Chiplet data and models</td>
<td>ALD, APLD, V</td>
<td>Chiplet data: mechanical, electrical, IO, power, thermal, datasheet data, etc. in XML format.</td>
</tr>
<tr>
<td>Package Selection</td>
<td>SA, PA</td>
<td>Selection of the packaging technology and materials suitable for the chiplet integration, considering factors like thermal management and signal integrity.</td>
</tr>
<tr>
<td>Netlist and schematic</td>
<td>ALD, APLD, SND</td>
<td>-For chiplets with programmable logic, the physical planner may need a netlist or schematic of the</td>
</tr>
</tbody>
</table>
intended logic design to estimate resource utilization within the chiplet. If advanced packaging techniques like 3D stacking are used, netlists could be necessary to evaluate interconnect options between stacked chiplets.

<table>
<thead>
<tr>
<th>Outputs</th>
<th>User Persona</th>
<th>Description/notes/comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Package layout</td>
<td>SA, PA</td>
<td>The physical layout of chiplets on the package or die, including their placement, orientation, and spacing.</td>
</tr>
<tr>
<td>Routing diagrams</td>
<td>SA, PA</td>
<td>Routing diagrams showing the electrical connections (traces) between chiplets and the power distribution network.</td>
</tr>
<tr>
<td>Structural Netlist</td>
<td>SND, PLV</td>
<td>Structural netlist describing connectivity at the substrate level. This is typically a Verilog netlist.</td>
</tr>
<tr>
<td>Power delivery network design</td>
<td>SA, PA, PE</td>
<td>Design plan of the power delivery network, including voltage domains, power routing, and distribution components.</td>
</tr>
<tr>
<td>Thermal models</td>
<td>SA, PA, TE</td>
<td>Thermal models showing the heat dissipation profiles and the effectiveness of thermal management solutions.</td>
</tr>
<tr>
<td>Signal Integrity Analysis Plan</td>
<td>SA, PISI</td>
<td>Plans for signal integrity simulations, indicating compliance with timing, noise, and jitter specifications.</td>
</tr>
</tbody>
</table>
Power Planning and Analysis workflow

Introduction

Heterogeneously Integrated ICs increase the complexity of power analysis and require a system level view of power. Complexity of power tree design increases substantially because of the number of power supplies for each component. Careful consideration is needed to determine how power will be delivered to the chiplets - some chiplets may require dedicated power supplies, some die to die interfaces may require power be supplied from another chiplet, and some chiplet’s power supplies may use common supply.

In addition, voltage regulator integration into the package creates additional constraints. Due to inefficiencies in voltage regulation, a significant amount of heat is dissipated in the Voltage Regulator ICs. This power dissipation leads to increased heating of the package. Additionally, power management through voltage level modulation needs to ensure that all chiplets can operate at the modulated voltage.

Power Modeling and Analysis workflow analyzes power dissipation under different use scenarios. A scenario is defined broadly as the mode of operation of each chiplet, power tree connectivity and voltage regulator settings. Chiplet operating modes can be considered as “major” and “minor”, where “major” corresponds to a particular power configuration, and “minor” to operating options within that, e.g. for an RF IC “transmitting” might be a major mode, the frequency band would be a minor mode. However, in many cases power consumption will be within a known range for any major mode of operation, so a list of major modes for the chiplets and the corresponding power characteristics may be sufficient if you know how the modes of the chiplets in the system correspond for possible system scenarios, e.g. boot-up, standby, normal operation. Given the chiplet power needs across the scenarios one can work out power supply needs and estimate the extra power lost due to power supply inefficiency. Chiplets that exhibit high power usage in (say) boot-up, may be powered up in sequence to avoid excessive system load.

In IC design UPF is often used to define power management intent, but that is a design input rather than a model of power behavior. Thermal models can be expressed in languages like Verilog-AMS, and would be a collection of models relating input voltage to output power for specific chip areas. The time constants in a thermal model are much longer than in the electronics, and can be run in parallel with behavioral simulation since the feedback of temperature is rarely immediate. As with thermal modeling, a list of sensitive points for temperature feedback should be supplied and feedback mechanisms supported - languages like Verilog-AMS and SystemVerilog support cross-module references, so a list of hierarchical names and associated physical location would suffice. Relaying the information as parameters for standard models makes it language independent. The ECXML for the hardware can be tagged with
connection points, and other models (like IBIS for pads) can also be indicated as sources of heat just by pad name. The details of thermal modeling are covered in the following section, co-simulation may be addressed elsewhere in OCP (e.g. HW/SW/AI codesign).

Statistical information about mode sequences and time spent in each mode is likely sufficient for determining power and thermal behavior, that would imply things like duty cycle for chiplets that are switching mode (e.g. transmitting vs receiving). Modeling tools can walk through legitimate sequences, and random possible sequences looking problems, and also with different chiplet layouts to see which work best. Designers generally spend their time making sure that particular modes of operation work as intended, they tend not to look at the mode transitions as much; in thermal analysis with power simulation the mode transitions are important because heat travels as a wave and transitory hot spots may occur around the mode transitions.

Chiplet power consumption may be used as an indicator of chiplet health, so extra components may be inserted to get access to key components. IEEE standards like P1687.2 address testing analog circuits (which includes power). Models of the analog test buses and key components can be supplied as behavioral (black box) or SPEF style, along with models of failing circuits. Complex switching circuits may be able to degrade gracefully as power devices (the big transistors) fail, which corresponds to the system dropping into a failing scenario. Many multi-processor systems can operate while missing cores, or have cores working only in lower power modes, so it may be desirable to track such information in systems running code that spans multiple machines that cannot stop and start easily. Likely failure modes should be included in chiplet mode information to help the tools model failure and help software manage it, complex chiplet systems are unlikely to come up fully working, or stay fully working when constructed from trillions of nm scale transistors, and most failures are electro-migration related (exacerbated by overcurrent and higher temperatures).

Aging of circuits is very dependent on temperature, including data on MTTF for a given temperature with the power and/or sensitivity models helps the tools adjust for optimal life span and maybe redundancy. MTTF information can be tied to the mode descriptions to indicate what modes will no longer operate.

Chiplet Power Models must account for the following factors:

- Voltage: Power dissipation has strong dependence on voltage and operating frequency
- Temperature: in the absence of thermal modeling for the system, models should indicate expected operating temperatures (and allowed excursions)
- Process: basic IC information (thickness, thermal conductivity)
- Modes of operation: all major modes, and (optionally) failing modes. Minor modes optional.

This work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International
Purpose of flow

Power analysis flow serves to provide necessary inputs for Thermal, Si/Pi and Mechanical analysis flows. Most electronic systems are powered by switch-mode power supplies, which tend to have an optimum efficiency of around 97% for a narrow band of power; simulation of the system is needed to provide the SMPS design requirements.

Figure 13: Example power tree of a chiplet system with substrate embedded voltage regulator. Image courtesy Thrace Systems.

Newer approaches to SMPS that can be done fully on-die with switched capacitor circuits may be more suitable for insertion in 3D-IC stacks, but are also harder to optimize. SMPS ICs are generally on special processes that handle higher voltages built on nodes above 45nm, and maybe hybrid designs which use GaN or SiC for switches (SiC is a particularly robust material). At the higher frequencies where GaN and SiC have an advantage over Si, interaction through radiated switching noise can be another effect that has to be taken into account in simulation since there is less opportunity to add shielding in a chiplet system. Knowing the sensitivity of chiplets to electrical noise is useful in optimizing the SMPS to avoid unwanted interactions, a tolerable spectral envelope should be provided, noting that orientation may be
important, so multiple of those may be required. Placement of SMPS IP within the system is also important; areas of Chiplets sensitive to noise should be marked up in the ECXML.

Power modes

In a typical usage scenario systems will go through different modes of operation, which are significant for power dissipation. However, usage scenarios are not limited to usage within the end system only. Before embedding into the final system, HI 3D ICs will go through manufacturing flow, where they will be tested to ensure their functionality. The Design For Test (DFT) flow will put the chiplets into power states which are significantly different from typical end-user environments. In these states chiplets as a whole, or parts of the chiplets, are expected to dissipate significantly higher power.

The power analysis in DFT consists of primarily three components. Scan Shift mode, Scan Capture mode and Memory BIST Power. The power analysis to be carried out at gate level netlists of the constituent chiplets and collect the data. This exercise to be repeated at SiP level by activating the pattern sets for 3 DFT modes. The typical peak power in DFT mode especially in scan shift and Memory BIST mode is higher than the functional number and needs to be accounted for in power planning.

Inputs/Outputs

<table>
<thead>
<tr>
<th>Inputs</th>
<th>Personal[table]</th>
<th>Description/notes/comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>System Power Requirements</td>
<td>SA</td>
<td>Maximum power limits/budgets, Maximum supply current limits.</td>
</tr>
<tr>
<td>Scenarios</td>
<td>SA</td>
<td>Includes usage scenarios, and when applicable, power management states, e.g. idle, sleep, low voltage, thermal throttling. This also includes DFT power modes.</td>
</tr>
<tr>
<td>Power tree</td>
<td>SA, PE</td>
<td>Power tree, including VR and connectivity.</td>
</tr>
<tr>
<td>Component power dissipation models</td>
<td>SA, PE</td>
<td></td>
</tr>
<tr>
<td>Current limits on DFT pads</td>
<td>SA</td>
<td></td>
</tr>
<tr>
<td>Data/file/step</td>
<td>Persona [table]</td>
<td>Description/notes/comments</td>
</tr>
<tr>
<td>---------------------------------</td>
<td>-----------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Power tree</td>
<td>SA, PE</td>
<td>Power tree, including VR and connectivity.</td>
</tr>
<tr>
<td>Scenario Design Power</td>
<td>PE</td>
<td>Scenario Design Power results are used by mechanical design</td>
</tr>
<tr>
<td>Current draw profile</td>
<td>PE</td>
<td>Max currents are used by PI to analyze package IR drop, EM limit compliance, and current crowding. Current ramps are used by PI engineers to design and analyze system power supplies, and amount and placement of decoupling capacitors.</td>
</tr>
<tr>
<td>Detailed transient power density map</td>
<td>PE</td>
<td>Power Density maps are used for Thermal engineers to analyze heat dissipation and maximum temperatures and by Mechanical engineers to analyze stress, warpage and displacement.</td>
</tr>
</tbody>
</table>
Thermal Modeling and Analysis workflow

Introduction

Thermal analysis for ICs serves to find if any regions of the ICs exhibit high temperatures during normal operation.

Typically temperature limits are based on reliability and performance concerns. Difference in power densities causes some regions of the die to exhibit higher temperatures than the rest, causing hot spots.

Integration of multiple ICs in a SiP allows for bigger and more complex ICs, but also increases the overall power density and concerns about maximum temperatures. In addition, heterogeneous components come with their own maximum temperature limits, e.g. memory modules may have lower maximum operating temperatures than a GPU.

In chiplets implementing advanced power management techniques, such as thermal throttling and DVFS, thermal analysis can help answer what, if any, impact the thermal cooling design has on performance.

Purpose of flow

As part of the overall system design Architects need to consider ICs temperature when exploring package technology choices, materials, chiplet integration strategy and a thermal cooling design. All these choices have a strong impact on heat dissipation capability.

Detailed thermal modeling and analysis, which includes consideration of chiplet operation and power densities, package technology choice, placement options and cooling solution options, is required to:

- Determine and validate optimal thermal placement
- Determine and mitigate thermal coupling between components
- Determine max components temperatures during operation, e.g. memory, optical, CPU
- Since many advanced components, such as GPU, CPU, implement advanced power and thermal management techniques, detailed thermal analysis is needed to determine actual operational components’ temperatures

This work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International
To accomplish these goals vendors of chiplets need to provide physical geometry and power dissipation information for their components. This is best accomplished by using ECXML files, which contain the following information:

- Physical geometry information of the die and any additional structures such as copper pillars and bumps
- Material information, including thermal conductivity, density and heat capacity
- Power dissipation, provided as either 2D or 3D heat flux

![Thermal simulation of 2.5D chiplet system. Image courtesy Thrace Systems.](image)

Thermal models for chips can be relatively independent from the functional models of the chips themselves, power consumption tends to an average for any given operating mode. Models can be generated for use in Chiplet design flows that will “piggyback” other models in the functional simulation flow, rather than using more detailed (power aware) models within the functional simulation. Although the detailed models are required to create the piggyback versions, they are not likely required to be delivered with the IP. Thermal simulation can be performed using the same simulators as analog circuit simulation, and using the same HDLs, as with electronics, AI/digital-twinning can be used to create reduced-order mathematical models for the thermal behavior.
A separate file can be provided for each significant power dissipation mode, but at a minimum there should be one for each of Functional and Test modes.

Inputs/Outputs

<table>
<thead>
<tr>
<th>Inputs</th>
<th>Persona</th>
<th>Description/notes/comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scenarios</td>
<td>TE</td>
<td>Includes usage scenarios, and when applicable, power management states, e.g. idle, sleep, low voltage, thermal throttling</td>
</tr>
<tr>
<td>Materials</td>
<td>ME</td>
<td>Material thermal properties, e.g. thermal conductivity, density, specific heat capacity, etc.</td>
</tr>
<tr>
<td>Packaging choices</td>
<td>TE, PE</td>
<td>Options for package design, chiplet placement, and material information</td>
</tr>
<tr>
<td>Power dissipation of components</td>
<td>PE</td>
<td>Includes, when applicable, power at different power management states, e.g. idle, sleep, low voltage, thermal throttling</td>
</tr>
<tr>
<td>System design and cooling options</td>
<td>SA, TE</td>
<td>Includes data for items outside of the SiP, such as board, components, board mounting, heatsink, etc. When applicable may include additional system and environmental information, e.g. enclosures, air flows, etc.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Outputs</th>
<th>Persona</th>
<th>Description/notes/comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max component temperatures</td>
<td>TE</td>
<td></td>
</tr>
<tr>
<td>Hot spot locations</td>
<td>TE</td>
<td></td>
</tr>
<tr>
<td>Temp sensor</td>
<td>TE</td>
<td></td>
</tr>
</tbody>
</table>
placement guideline
Signal Integrity and Power Integrity workflow

Introduction

During the micro-architectural specification workflow described earlier, an SA physically partitions a custom SOC into several dies or chiplets to be interconnected within a package. There are many ways this partitioning can be done (e.g., whether to have separate chiplets for core, memory, and IO; whether to use same or different process nodes for chiplets; whether to go serial or parallel interface; what protocol to choose). There are also several choices for packaging technologies (e.g., whether substrate-based, interposer-based, or RDL fanout-based). Thus, there are several possible design scenarios (or micro-architectures or SiP design scenarios) that need to be analyzed to choose an optimal one. Figure 15 shows a few possible micro-architectures. Die 1 and Die 2 in Figure 15 can be connected through an organic substrate, or an silicon interposer, or RDL interposer.

![Figure 15: A few different packaging technologies for 2.xD integration (Courtesy: Microchip)](image)

As part of the micro-architectural workflow, architectural specifications (e.g., power dissipated, performance, area, overall cost, time-to-market) are refined into high-level specifications (e.g., power dissipated, latency, data rate, bandwidth, packaging technology, cost) for each micro-architecture. The objective is to choose the optimal micro-architecture that best meets the architectural specifications. Functional specifications are part of the broader architectural specifications and describe if the interface will work as intended. Predictive signal integrity analysis is done to check the D2D’s performance against the minimum functional specifications.

This work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International
Purpose of flow

A chiplet (or die) can usually be a core, a memory, and an IO. Each D2D interface can be classified as either serial or parallel. There are several standards or protocols available in each type of interface. For example, USR, XSR are couple of the available protocols for a serial D2D interface, while Intel’s AIB, Intel’s MDIO, TSMC’s LIPINCON, OCP’s Bunch of Wires (BoW), and ODSA’s Open HBI are some of the available protocols for a parallel D2D interface. The choice of the type of interface and the protocol depends on many factors, including but not limited to: data rate, total bandwidth, power dissipated, latency between chiplets, routing constraints, choice of the packaging technology and interposers, and number of pins. Each {type, protocol} pair of a D2D interface has minimum specifications to be complied to at the transmitter and receiver. The purpose of the workflow is to describe how to go about verifying for this compliance for a given D2D interface.

Expected results

a) Determine if D2D interfaces in a given micro-architecture are in compliance with their respective SI functional specifications.

b) Determine the optimal configuration (placement, trace width, etc.) for a given micro-architecture that best meets the SI functional specifications.

c) Determine the optimal micro-architecture that meets the SI functional specifications.

Serial D2D Interface:

A conceptual system-level equivalent circuit of a serial D2D interface is shown in Figure 16.

Figure 16: Conceptual system-level equivalent circuit of a serial D2D interface

In Figure 16, the bits to be transmitted (0100110 in Figure 16) are given as input to an optional transmit equalizer (“Tx Equalization” in Figure 16). This equalizer boosts the amplitude of the high-frequency contents of the input bit waveform relative to its low-frequency contents, through an Finite Impulse
Response (FIR) filter. There may be an additional equalization at the receiver. In present, this equalization is usually a decision feedback equalizer (DFE). Both the Tx and Rx equalizations are optional. If specified, the realization of the filter is kept as a secret, and usually a DLL that implements the equalization is given. Accurate modeling of the components from Tx to Rx is required to enable tools to minimize reflections in the channel, likely a long gradient-descent optimization exercise involving multiple channels at the same time to also minimize crosstalk, the SERDES operate at mm-wave frequencies, and the mechanical design impacts behavior.

The Tx Driver and Rx Driver in Figure 16 are the drivers at the transmitter and receivers, respectively. Each of them is specified through an IBIS model. The model of an IO chiplet should contain this IBIS model and optionally an AMI DLL for the equalizer. If IBIS models are insufficiently accurate, Verilog-AMS is an alternative (standard).

The pins of these drivers are attached to either a bump or a microbump, which is usually represented by a RLC model (series R, series L, and a shunt C). A quasi-static 3-D electromagnetic solver (e.g., Ansys Q3D) can be used to construct an equivalent circuit of the bump or microbump.

The wires connecting Die 1 and Die 2 in Figure 15 are usually copper. For a SERDES configuration in Figure 16, the line is a differential transmission line. An equivalent circuit for this line is needed. Usually, the equivalent circuit is given in the form of a multi-section W-element transmission line model. Such a model can be constructed using 2-D electromagnetic solvers. Alternatively, scattering parameters (S-parameters) of the line can be extracted through a 3-D electromagnetic solver and a Touchstone file containing them can be used as the model for the line. The system/circuit simulation tool can construct an equivalent circuit that best represents the S-parameters.

When each bit in the transmitted or received bit sequences are overlaid in the same time window of an unit interval, an eye diagram is usually seen (circles titled 2 and 3 in Figure 16). A wide open eye suggests a design with good signal integrity, and a small open eye or a completely closed eye represents a design with poor signal integrity. For SERDES protocols, minimum eye width and eye height of the eye diagram at the transmitter (at circle 2) and at the receiver (at circle 3 and circle 4) are specified at a particular bit error rate (BER). Errors in received bits are unavoidable, so the only thing that can be specified is the fraction of the number of bits that can be received erroneously, defined as bit error rate (or just BER). Usually, BERs for D2D interfaces are less than or equal to 1E-10 (which means one in 10 billion bits transmitted can be received erroneously). These quantities and the corresponding BER should be part of the chiplet signal integrity (SI) model. The rectangle (red rectangle inside eyes at circle 2 and 3) with length as the eye width (EW in Figure 16) and breadth as eye height (EH in Figure 16) whose center
is the centered at the center of the eye is known as the eye margin. The eye margin formed by the minimum eye height and eye width at the target BER is referred to as the eye mask.

The whole design in Figure 16 should be created in a circuit simulator (e.g., Siemens HyperLynx, Ansys Electronic Desktop). A SERDES channel simulation (bit-by-bit or statistical) could be launched, and the eye margins at different points in the system found at the target BER (the one present in the chiplet SI model). The eye height and eye width of the eye diagram at transmit pins (circle 2 in Figure 16) at the target BER should be greater than those specified in the protocol. The same expectation applies at the receive side (Circle 3 and Circle 4 in Figure 16) as well.

If any of the eye diagrams at Circle 2 or Circle 3 or Circle 4 violate its corresponding eye mask, then the D2D interface is considered to have failed the SI test. A design space exploration (like an design of experiments (DoE)) should be done to see how eye height and eye height at different points change as a function of spacing between dies, width of the transmission lines, etc. If the DoE study reveals no configuration where the SI test passes, then the D2D interface is considered to have failed the SI test. If, on the other hand, the DoE study reveals that there is, at least, one configuration (trace width, placement of dies, etc.) of the D2D interface that passed the SI test, then the D2D interface is considered to have passed the test; in this case, the configuration that provides the largest difference between the eye margin and the eye mask is chosen as the optimal D2D interface.

**Parallel D2D Interface:**

If there is a parallel D2D interface in the micro-architecture, then the block diagram below (Figure 17) captures how this interface will look like for any parallel protocol. The input bit stream to be transmitted is divided into multiples of bytes, and each divided chunk is transmitted simultaneously. Each bit in this chunk is transmitted in a separate data line, as shown in Figure 17. Each line consists of a transmitter, which is usually just a transmit driver, but can optionally include a feed forward equalization. The transmit driver is used to convert digital bits into analog pulse waveforms. The Tx driver drives the analog waveform into a bump/micro-bump, which return is connected to the passive analog interconnects. Each data line is single-ended (i.e., does not have its dedicated return path). Different data lines usually run parallel to each other and usually are of same length to minimize skew between received waveforms. Data lines are electromagnetically coupled to each other, a fact important in the extraction of transmission-line models for these lines. Unlike in a serial D2D interface, a coupled transmission line model (or coupled scattering parameters models) is needed for coupled data lines. Clock is usually transmitted in a differential line (i.e., has its dedicated return path). Each line gets terminated in a bump/microbump at the receiver die. This bump/microbump is connected to the Receiver, which is usually just the receive buffer, but can optionally include equalization blocks (e.g., continuous-time linear equalizer or decision feedback equalizer). The receive buffer converts the analog waveforms into digital bits at each clock transition.
Each parallel PHY D2D protocol describes minimum signal integrity specifications, like minimum setup and hold times of analog waveforms at transmit and receive sides. Each line’s SI performance should be compared with these minimum SI specifications. A micro-architecture is said to pass the parallel D2D phy test if there exists at least one configuration that meets the minimum SI specifications dictated by the standard. If there is more than one configuration that meets the minimum requirements, then an optimum configuration can be chosen among acceptable configurations. Each parallel PHY D2D protocol describes minimum signal integrity specifications, like minimum setup and hold times of analog waveforms at transmit and receive sides. Each line’s SI performance should be compared with these minimum SI specifications.

For the short link D2D through silicon interposer, the parasitics network can be simplified as RC model instead of RLC model based on study from Jangam et al.[1] as shown in Table 1 below. The silicon interposer routing electrical characterization concluded that silicon interposer behaves like RC-like behavior for short links less than 585um length and 10um width and the insertion loss is less than 2 dB for frequencies up to 30GHz for those short links[1].
Table 1. Si-IF vs Interposer vs PCB Parasitics Comparison[1]

<table>
<thead>
<tr>
<th>Technology</th>
<th>Link length/ Wire pitch</th>
<th>Inductance (nH) (Normalized to interposer)</th>
<th>Capacitance (pF) (Normalized to interposer)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Si-IF</td>
<td>100 μm/ 4 μm&lt;sup&gt;a&lt;/sup&gt;</td>
<td>0.35 (0.11)</td>
<td>0.02 (0.02)</td>
</tr>
<tr>
<td>Si-IF</td>
<td>585 μm/ 10 μm&lt;sup&gt;b&lt;/sup&gt;</td>
<td>0.52 (0.17)</td>
<td>0.06 (0.07)</td>
</tr>
<tr>
<td>Interposer</td>
<td>5 mm/ 6.4 μm&lt;sup&gt;c&lt;/sup&gt; [15]</td>
<td>3.125 (1)</td>
<td>0.855 (1)</td>
</tr>
<tr>
<td>PCB</td>
<td>20 mm/ 320 μm&lt;sup&gt;c&lt;/sup&gt; [15]</td>
<td>~12 (3.84)</td>
<td>~2.0 (2.34)</td>
</tr>
</tbody>
</table>

<sup>a</sup> Estimated from measured parasitics  
<sup>b</sup> Experimentally measured  
<sup>c</sup> Calculated

Input/Output Description

The inputs and outputs to the predictive SI analysis are described in the table below.

<table>
<thead>
<tr>
<th>Inputs</th>
<th>Supplier Persona</th>
<th>Description/notes/comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>D2D Interconnect Geometry, Protocol, and terminations</td>
<td>SI Architect</td>
<td>The length, width, thickness, pitch of D2D interconnects; the type of protocol (serial or parallel), and the actual protocol (UCle, BoW, etc.); the terminations at the end of the interconnect</td>
</tr>
<tr>
<td>SI Stack up</td>
<td>SiP and SI Architect</td>
<td>Stack up of the SiP that matters to SI analysis.</td>
</tr>
<tr>
<td>Constraints and variations</td>
<td>SiP Architect</td>
<td>Constraints to D2D geometry (max/min of width, pitch, thickness, etc); design choices for terminations, package architecture, etc.</td>
</tr>
</tbody>
</table>

This work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International
<table>
<thead>
<tr>
<th>Outputs</th>
<th>User Persona</th>
<th>Description/notes/comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>SI Compliance report</td>
<td>SI Architect</td>
<td>Eye pass/fail decisions for each micro-architecture</td>
</tr>
<tr>
<td>Micro-architectures</td>
<td>SI Architect</td>
<td>Micro-architecture (D2D interconnect + termination + SI stack up)</td>
</tr>
</tbody>
</table>

**Flowchart**

The workflow for an SI simulation at this early stage is different from that for an SI simulation at the post-layout stage in one main respect: the former needs a tight integration with design space exploration (DSE) to ultimately decide on the best micro-architecture from an SI perspective. A workflow for predictive SI simulation for D2D interconnects is shown in Figure 18.

The inputs to this workflow are broadly as follows:

1. The geometry of the D2D interconnects, the stack up, the constraints in geometry, materials, etc;
2. The different choices for terminations (single-ended/ balanced, kind of termination; value of termination) of the D2D interconnects; and
3. The PHY protocol used to communicate between the transmitter and receiver at the two ends of the D2D interconnect; the PHY protocol is dependent whether the D2D interconnect is a serial or a parallel interface; examples of a serial PHY protocol would be XSR and USR, and those of a parallel PHY protocol would be UCIe, BoW, OpenHBI, Intel’s AIB; along with the choice of the PHY protocol, its corresponding electrical specifications (data rate, rise/fall times, eye mask at transmit and receive and/or set up/hold time, target bit error rate, etc) are expected to be available and known.
The DSE is then configured to create different variations of the first two inputs. Even packaging architectures can be varied and given as part of the geometry input. The DSE step then feeds into a parasitic extractor that extracts electrical parasitics of the different passive elements. Finally, an electrical equivalent circuit of the different elements of the SiP architecture is created. This equivalent circuit is excited according to the electrical specifications of the desired PHY protocol. A traditional SI simulation (analysis types: statistical, bit-by-bit, or transient) is now run, and the relevant output parameters (e.g., eye diagram, eye height and width or setup/hold time, etc.) are computed. This process is repeated for different scenarios configured in the DSE. The configuration and the micro-architecture with the largest eye opening are chosen. The eye openings of the chosen scenarios are compared against the minimum electrical specifications of the PHY protocol. If none of the scenarios meet the specifications, then the SI test fails. If not, then the corresponding micro-architectures have passed the SI test.
Power-supply voltages anywhere in an SiP are different from those on the terminals of the voltage source. Furthermore, these voltages also vary with time. These variations (in space and in time) are primarily due to the non-ideal conducting nature (non-zero resistance, inductance) of the power distribution network (PDN) and are triggered when current flows between power and ground nodes of the PDN. The current flow can be either due to leakage in transistors or due to switching of transistors. The former type of current is continuous and present all the time the transistors are powered on. This current, also known as leakage current, when flowing through the PDN, causes DC voltage drops (or just IR drops) due to PDN's non-zero resistance. The other type of current flow occurs only when a CMOS transistor switches (from ON to OFF, or OFF to ON). This current, known as the switching current, when flowing through the PDN, causes switching noise due to PDN's non-zero inductance and resistance (in other words, due to PDN's non-zero impedance). The total noise in the voltage available between a

Figure 19: Predictive Power Integrity Simulation Workflow. Image courtesy of Siemens EDA
power node and its ground node is the sum of the DC IR drop and the switching noise at the node and is referred to as the power-supply noise (PSN for short).

The PSN cannot be eliminated. It can only be controlled and kept small at best. If the PSN is not kept within acceptable limits, it can lead to logic failures. Usually, these limits are given as some small percentage (usually less than +/- 5%) of the ideal power-supply voltage (i.e., the voltage available between the power and ground terminals of the voltage source). The PSN is controlled by keeping the impedance between any power-ground node pair to the power-ground terminals of the voltage source below an acceptable level. Since this impedance varies with frequency, this impedance should be below the acceptable level at all frequencies from zero Hertz to the maximum frequency of interest. This is accomplished in two steps. In the first step, the DC impedance is kept below the target impedance by adjusting the geometry (thickness and width of metal lines, routing, floorplan, etc.) and material. In the second step, the peak impedance is kept under the target impedance by placing decoupling capacitors (between power and ground nodes of the PDN) in carefully chosen places in the chip, the package and the board. The presence of capacitance (especially decoupling capacitance) at different domains (i.e., chip, package, and board) causes the frequency-dependent impedance between the power-ground nodes of switching transistors (and the power-ground terminals of the voltage source) to have a few resonances, too. These resonances are at different frequencies and are different in amplitude. The resonance due to the on-chip decoupling capacitance and the package inductance is at the highest frequency and has the highest amplitude compared to other resonances. These resonances are at different frequencies and are different in amplitude. The resonance due to the on-chip decoupling capacitance and the package inductance is at the highest frequency and has the highest amplitude compared to other resonances. Thus, the PSN budget at a die is higher than at the package or board. The PSN budget at a location comprises a DC IR drop budget and a switching noise budget. The DC IR drop budget is usually smaller than the switching noise budget.

There are two kinds of profiles of engineers working on ensuring power integrity in an SiP. The workflow for predictive power integrity simulation will be different between these two profiles. The first is the chiplet designers who are responsible for ensuring the power integrity of chiplets. The workflow for these designers will be almost the same as that for the monolithic SoC designers. The difference will be that these designers would now have to create a chiplet power model and other information needed for system architects to integrate the chiplet designer’s chiplet into the architect’s SiP.

The second kind of profile is the system architect (SA), who is responsible for designing the SiP comprising multiple chiplets. The SA will have access to CPMs for chiplets from chiplet vendors. They have to use these CPMs to find the right advanced packaging option and the right architecture for the SiP.
Input/Output Description

The inputs and outputs to the predictive PI analysis are described in the tables below.

<table>
<thead>
<tr>
<th>Inputs</th>
<th>Supplier Person<a href="table">a</a></th>
<th>Description/notes/comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>PI stack up without chiplets</td>
<td>PI Architect</td>
<td>Stack up of the SiP that matters to PI analysis (for example, D2D interconnects are not needed in the PI analysis)</td>
</tr>
<tr>
<td>Voltage sources and decoupling capacitors location and value</td>
<td>PI Architect</td>
<td></td>
</tr>
<tr>
<td>Chiplets information (location, connectivity, chiplet power model)</td>
<td>SiP Architect</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Outputs</th>
<th>User Person<a href="table">a</a></th>
<th>Description/notes/comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>PI performance</td>
<td>PI Architect</td>
<td>Whether or not a given micro-architecture met the expected PI performance.</td>
</tr>
</tbody>
</table>

Flowchart:

The flowchart of the predictive power integrity (PI) simulation workflow for an SA is shown in Figure 20.
For this simulation, only the PDN of the SiP is modeled (and other things like D2D interconnects are ignored).

This SiP PDN comprises two parts: 1) SiP without the chiplets, and 2) chiplets. The information regarding the first part includes the geometry, material, stackup, location, the number, and the value of voltage sources, location and values of decoupling capacitances, etc. From this information, an equivalent electrical circuit model of different components in the SiP’s PDN is built. These components include the C4 bumps, microbumps and through-silicon vias (TSVs) if present, silicon interposer if present, package PDN, solder balls, decoupling capacitance, PCB PDN, and voltage sources.

For the chiplets, it is assumed that the chiplet power models (CPMs) are available from the chiplet vendors. It is expected that CPMs contain an equivalent circuit of the chiplet PDN and the different current sources (leakage, switching). It is also expected that CPMs provide the target impedance met by the

---

This work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International
chiplet and up to what maximum frequency, the DC voltage source expected, the DC IR drop and switching noise budgets.

From the models of different parts of the SiP’s PDN, an equivalent circuit of the entire SiP’s PDN is constructed. The current sources causing the voltage drop are captured in the CPMs. The DC voltage source required for each chiplet is modeled as an independent voltage source at the PCB. This equivalent circuit can be created in a circuit simulator capable of running signal and power integrity simulations. The circuit simulator should be capable of doing DC, transient, and AC analysis. The SA should know the target impedance of the SiP’s PDN along with the PSN budgets of different chiplets.

An AC analysis can be used to compute the loop impedance at different locations in the SiP’s PDN. This analysis can cover frequencies in the interval \([0, f_{\text{max}}]\). The quantity “f_{\text{max}}” is the maximum frequency of interest up to which the loop impedance is expected to be less than or equal to the target impedance. This frequency is usually some fraction of the inverse of the smallest rise time in the switching sources.

The DC loop impedance is first checked to see if it is less than or equal to the target impedance. If it is, then we can move to the next step. If it is not, then geometry, material, etc. of the SiP’s PDN without the chiplets can be modified and the AC analysis repeated. Things like modifying metal thickness and width, line pitch, routing, placement, etc. should be explored to meet the target impedance.

Once the DC impedance is within the target impedance, then a DC analysis can be done to make sure that the DC IR drop is within the DC IR drop budget. If DC IR drop is not within the DC IR drop budget, then current drawn by the different chiplets should be checked.

Once the DC loop impedance and DC IR drop are within the target impedance and DC IR drop budget, respectively, then the loop impedance at the rest of the frequencies should be compared against the target impedance. If there is a violation, appropriate decoupling capacitance and its location and value should be changed to remove the violation.

Once the frequency-dependent loop impedance is within the target impedance, a transient simulation can be run to simulate dynamic voltage drop in the PDN due to switching. The transient current sources in the CPMs provide the switching current for this simulation. The dynamic voltage drop should be compared against the total PSN budget between \([0, T_{\text{max}}]\), where “T_{\text{max}}” is the maximum time of interest and is usually chosen to capture the smallest resonant frequency in the loop impedance. The dynamic voltage drop is compared to the PSN budget. This budget will be different at different time intervals. If the
dynamic voltage drop is not within the PSN budget, then the transient current sources in the chiplets need to be examined.
STA and Extraction

Introduction

As outlined in the Architectural Planning and Analysis section of this paper, the timing of the high speed D2D interfaces between chiplets generally utilize traditional Signal Integrity (SI) analysis tools and workflows. In addition to these high-speed, D2D interfaces, there are generally lower speed, D2D connections that also require timing planning, optimization, and analysis. These connections may include test, control, and slow speed data interfaces between the internal chiplets. SI analysis can be used to analyze these slower speed interfaces, but the timing analysis tools and workflows utilized in digital IC designs, namely Static Timing Analysis (STA) may be better suited for this purpose. In the following sections, we will provide a simplified summary of existing STA tools and workflow and discuss new tools and workflows required to adapt this methodology for SiP level analysis.

STA Overview

Timing analysis for small IC designs can be implemented using Spice level simulations and/or back annotated, gate level timings simulations. For larger designs, this approach can be prohibitive. Static Timing Analysis is an alternate method that can exhaustively evaluate all timing paths in large, digital synchronous designs.

STA tools include a delay calculator which computes timing paths based on timing models for the active components and extracted RC models of the wires connecting those active components. Liberty Models (.LIB) models are provided for all the active components within a design. These may include low level building blocks including standard cells, memories, and macro level blocks such as IO buffers, high speed IO macros and analog/mixed signal macros. These .LIB models are created by performing detailed timing simulations (generally at the transistor level using a spice level simulator) to create a table lookup model for the timing paths to/from the input/out pins of the cell/block. These models define the electronic device's internal timing paths considering the input clock rate and associated rise/fall times, the input signals rise/fall times and the input and output capacitive loading. IO checks such as setup and hold time requirements for signals relative to the associated clock signal, and skew requirement between signal inputs or outputs signals can also be defined in the .LIB models.

The Structural Netlist Designer (SND) defines a set of design constraints using the Synopsys Design Constraint (SDC) format to specify the design intent, including timing, power, and area constraints for a digital design. The SDC constraints define the design’s clock domains, frequencies, and all the necessary timing requirements for the design.
The inputs to the STA workflow include the SDC timing constraints, the .LIB models for the active components/block used in the design in conjunction with a set of parasitic RC extractions (PEX) files in Standard Parasitic Extended Format (SPEF) files for all of wiring interconnect between the design's signals and active logic component/blocks. These inputs are used to perform an exhaustive analysis of the design, checking the timing results to the design timing constraints. Reports are generated for paths not meeting the timing constraints. The physical design team uses these reports to refine the design to meet timing or relaxing the timings constraints as required. Due to process variations in the ASIC manufacturing process, different .LIB models are provided to represent these variations that can impact timing. Different RC models for the interconnect are also used to account for the process variations of timing attributed to the routing paths which can be appreciable. Capacitive coupling between adjacent wiring paths is also included in the extracted RC SPEF files to account for crosstalk timing effects. Additional statistical mechanisms are used in the STA timing delay calculator and models to account for process induced timing variations.

STA is the predominant method used to analyze and sign off timing for digital, synchronous ASIC and chiplet devices. For very large ASIC/chiplet designs, a hierarchical design approach breaks the design into physical blocks to manage the complexity of the design and practical limitations of the EDA design and analysis tools. This enables a divide and conquer approach to break the design into multiple blocks that can be concurrently designed, analyzed, and optimized. For smaller designs, a complete set of timing constraints can be created to check the entire design. For larger, hierarchical designs, a complete set of timing constraints are defined for each block. To analyze the entire design, top-level timing constraints are defined to check the block-to-block timing as well as the chip level input/output timing constraints. To constrain the internal blocks to accommodate the top-level timing constraints, an additional set of IO block constraints are defined for each block to be used by the block designers to meet the timing constraints in their respective blocks. This process is often referred to as timing closure. Once all blocks are timing closed, the full chip level timing can be checked using the detailed top and block level models. For very large designs, it may be prohibitive to run a full chip-level STA analysis using detailed block models for all blocks, so the chip-level STA runs can be broken into separate full chip runs that focus on specific block-to-block interfaces using the detailed block models for the blocks of interest and abstract models for the other blocks.

**SiP Level STA Overview**

For large SiP level designs, a full gate-level STA timing approach will most likely be prohibitive. As such, an analogous, hierarchical approach can be deployed treating each of the chiplets as “blocks” and the SiP level analysis will focus on top level, chiplet-to-chiplet, low speed interfaces. This methodology will require a simplified .LIB model to be created for each of the chiplets primarily focused on the chiplet input/output
timing constraints. Designers may reference the relevant C2C/D2D specifications used in the respective chiplets to assist in the development of the SiP level STA timing constraints and analysis. Multiple .LIB and RC extraction SPEF models will be required to account for the process variations of the respective chiplets and D2D interconnect.

In 2.5D designs, there are many different packaging technologies used to implement the D2D interconnect. This interconnect can deploy silicon and/or organic substrate/interposer materials which require different extraction tools and workflows. A hybrid, silicon/organic parasitic extraction (PEX) workflow is required to support 2.5D STA analysis.

True 3D digital IC designs will likely deploy traditional, ASIC level STA methods for both die level and die-to-die level timing analysis, but changes will be required to accommodate the new Hybrid Bonding (HB) D2D interconnect and muti-ASIC technology nodes.

SiP Level STA Workflows

A high-level summary of a 2.5D, SiP level STA workflow is illustrated in the figure below.

---

Figure 21: Flowchart describing the SiP Level STA Workflows. Image courtesy of Siemens EDA

This work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International
## Input/Output Description

<table>
<thead>
<tr>
<th>Inputs</th>
<th>Supplier Persona [table]</th>
<th>Description/notes/comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>CDX (Chiplet .LIB )</td>
<td>Chiplet Vendor</td>
<td>Block level timing file defining the top level I/O timing of each chiplet. This may include setup and hold time requirements with respect to an actual or virtual clock, latency of the signals input load capacitance. The chiplet .LIB timing models also include the chiplet output timing with respect to a clock and/or data inputs. Output capacitance loading and the maximum allowed output loading may also be included</td>
</tr>
<tr>
<td>Timing Constraints (SDC)</td>
<td>Structural Netlist Designer (SND)</td>
<td>The timing constraints define the SiP level, external input IO setup/hold timing, input slew rate, external IO timing and maximum loading requirements. Also included are the chiplet-to-chiplet (C2C) timing requirements for all low speed C2C signals.</td>
</tr>
<tr>
<td>SiP Level Netlist (.V)</td>
<td>Structural Netlist Designer (SND)</td>
<td>SiP level netlist in Verilog format generated from the top-level physical package design database including the chiplet instantiations, top level Input/Output declarations and all external I/O to chiplet and C2C signals.</td>
</tr>
<tr>
<td>SiP level PEX files (SPEF)</td>
<td>ASIC logic designers (ALD) and/or PI/SI engineer/designer (PISI)</td>
<td>RC parasitic extractions (PEX) in SPEF format for all external I/O to chiplet and internal C2C signals traces. Note that different extractions tools and persona may be required for signal or segments of a signal trace.</td>
</tr>
<tr>
<td>Outputs</td>
<td>User Persona</td>
<td>Description/notes/comments</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>--------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>STA Timing Report(s)</td>
<td>ASIC logic designers (ALD)</td>
<td>An ALD will run the SiP level STA tool and generate timing reports from the STA tool which are used to validate the SiP level timing (excluding high speed paths analyzed with SI analysis). The ASIC and Package design teams will modify the chiplet, interposer and substrate as required to correct or relax/waive any timing violations identified in the STA report.</td>
</tr>
<tr>
<td>Timing Summary Report(s)</td>
<td>ASIC logic designers (ALD)</td>
<td>An ALD will often generate Timing summary reports documenting a summary of the timing constraints, STA timing results and any resultant waived or relaxed timing violations. The reports may be used in project design reviews.</td>
</tr>
</tbody>
</table>
Mechanical Analysis workflow

Introduction

As the industry enters the digital transformation and exascale computing era, the increasing amount of data from all sectors is raising a problem of operational and storing cost of the data. Chiplets or die partitioning offers a compelling value proposition compared to traditional monolithic SoC for yield improvement, IP reuse, performance and cost optimization as well time to market reduction. Chiplet integration allows the integration of disparate technologies from multiple vendors to provide more flexible mix-and match systems to accelerate performance and improve power efficiency without requiring deployment of these technologies across an entire SoC simultaneously. Various advanced packaging technologies have been developed and offered a wide portfolio for chiplets from low density to high density including FCMCM (Flip-chip multi-chip-module), High density Fanout such as FOCoS (Fanout chip on substrate) and 2.5D Si TSV.

Mechanical modeling and analysis (M&A) are key enabling technologies for chiplets Integration that will support product development across the chip-package-board-system domains. Analysis using modeling and simulation tools today are mainly applied for a single mechanical domain (Die, Package or Board/System). The future will require multi-physics and scale capabilities as well design collaboration (die-package-board/system). The mechanical modeling and simulation tools will also be required to support development of both process and assembly design kits (PDK and ADK).

Mechanical reliability concerns are more elevated in a chiplet-based design than in a monolithic SoC for the following reasons:

1. Heterogeneous Integration: Chiplet-based designs typically involve integrating multiple chips from different suppliers or with varying technologies. This can lead to differences in material properties, thermal coefficients, and mechanical behaviors, making it more challenging to ensure uniform mechanical reliability across the system.

2. Interconnection Complexity: In chiplet-based designs, there is a need for numerous interconnections between chips, such as through-silicon vias (TSVs), microbumps, and solder joints. These interconnections are susceptible to mechanical stresses, including thermal expansion mismatches and mechanical shock, which can lead to reliability issues like solder joint fatigue, cracking, and delamination.

3. Thermal and Power Challenges: The thermal and power challenges in chiplet-based designs can be more prominent due to non-uniform temperature distribution among the chiplets leading to thermal hotspots and gradients that can exacerbate mechanical stress and reliability concerns.
4. Packaging Complexity: The packaging of chiplet-based designs must account for multiple chips, each with its own mechanical properties and requirements. This complexity can lead to mechanical inconsistencies. These variations can result in uneven mechanical stress distributions and affect long-term reliability.

Addressing these concerns requires a holistic approach that starts at the architecture exploration stage, material selection, design optimization, packaging/manufacturing, and reliability validation in the chiplet-based systems. The workflow for this approach is proposed in the following section.

Purpose of flow

The traditional approach to mechanical analysis was predominantly undertaken by systems designers who would subsequently impart the requirements/constraints to package designers. The primary focus of chip designers has historically centered on electrical analysis. However, in the realm of multiple chiplets integration, chip design can no longer avoid issues related to mechanical and thermal stress, particularly for advanced high density 2.5D/3D heterogeneous packages.

To navigate this evolving landscape effectively, multi-scale modeling becomes crucial. This requires the need to model chip interactions at the Si to package at the micrometer to millimeter scale, covering components such as Through-Silicon Vias (TSVs) and microbumps. Furthermore, it encompasses analyses at the millimeter to meter scale for entire systems, incorporating elements like heat sinks and printed circuit boards (PCBs). The analysis tools must encompass not only electrical analysis but also address the interaction of thermal and mechanical aspects within the chip design workflow.

As a result, the mechanical analysis flow for chiplet integration emerges as a multifaceted process encompassing a diverse array of tasks. These tasks span chip-to-package co-design, stress analysis, thermal analysis, failure analysis, and optimization. By analyzing and optimizing the mechanical behavior of the chiplets and packages, this comprehensive process plays a critical role in ensuring the reliability and performance of the integrated system. It operates across three distinct levels of analysis: the chip-to-package interconnect level, the package level, and the board-system level. Figure 22 shows the schematic mechanical analysis paradigm for chiplets integration system.
Flowchart of Mechanical Analysis

Stress modeling is one of key attributes in mechanical analysis. In traditional packages, there’s usually a safety margin in mechanical design to prevent stress from reaching failure levels. So, stress modeling is often used only after a failure occurs. However, for advanced packages with different chiplets integration systems, various factors such as complexity of package structures, new materials, and form-factor reduction may drive stresses to the limits. Therefore, stress (thermal and mechanical) modeling should be considered at the early design stage. It helps predict reliability and ensures that the package and system can handle the expected stress, preventing potential failures.

Meanwhile, the integration of chiplets and heterogeneous systems combines various functions such as RF, digital, and analog/mixed-signal, into a single package. When the electronic device is powered on, Joule heating can cause hot spots and uneven temperatures in the system. Additionally, different materials used in the chip, package, and board levels will experience varying degrees of thermal expansion. This can generate thermal stress induced by thermal mismatch among the materials and...
ultimately cause many failures in electronic devices. Differential expansion can also change electrical properties through the piezo-resistive effect, which will affect timing.

In addition, moisture absorption and desorption can also introduce additional mechanical stress. For moisture-sensitive materials such as polymeric materials, underfill material, swelling can occur during moisture absorption, while contraction can occur during desorption, leading to the induction of hygroscopic stress. Furthermore, during reflow, high internal vapor pressure can also be generated, in addition to thermal and hygroscopic stress.

Therefore, mechanical analysis must address the need for chip-interactions at the device level, package interactions at package level and system integration (heat sink, PCB board) at system level. The analysis flow can be considered as below:

- **Pre-Assembly/ Pre-Test(warpage/stress) Analysis** - design, materials, structural analyses
  - Goal is to find appropriate BoM (Bill of Material), Package architecture using mechanical modeling and simulation (2D & 3D model).
  - Inputs are material properties (Tg, Young’s modulus (E), CTE), structure (die sizes, package sizes, stackup), design rules (metal densities, keep out zones, d2d spacing), performance and stackup dependency

- **Post-Assembly/Verification (Post warpage/stress)** - include balls, stiffener ring design
  - Goal is to verify chiplets integration package and system can withstand the stresses.
  - Once the package design is complete, a stress analysis is performed to determine the stresses and strains that will be present in the package during operation. This analysis typically involves finite element modeling (FEM) to simulate the mechanical behavior of the package and chiplets under various loading conditions.

- **Package level stress (reliability concerns)**
  - Assembly temperature from hot to room temperature
  - Temperature cycling
  - Unbiased (hot storage temperature, moisture)
  - Biased Power cycling

- **Design considerations**
  - Thermal expansion offset (design rule for guidance + simulation)
  - Layout considerations (keep out zones for chiplets and assembly manufacturing, stiffener ring)
  - Other considerations

This work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International
Input description

1. Geometry and materials information: This includes the dimensions and materials properties of the chiplet, package, and board, as well as the interconnect details.

2. Boundary conditions: This includes the loads and constraints applied to the chiplet package, such as thermal loads, mechanical loads, and constraints due to mounting and support.

3. Operating conditions: This includes the expected operating temperature range and power dissipation of the chiplet package, as well as any other relevant environmental factors.

4. Material properties: This includes the material properties of the chiplet, package, and board, such as thermal conductivity, coefficient of thermal expansion, elastic modulus, and Poisson's ratio.

5. Interconnect details: This includes the design and layout of the interconnects between the chiplet and package, such as the number of interconnects, their dimensions, and the type of material used.

6. Manufacturing process details: This includes details about the manufacturing process used to assemble the chiplet package, such as the temperature profile and duration of the reflow process.

<table>
<thead>
<tr>
<th>Inputs</th>
<th>Persona/table</th>
<th>Description/notes/comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Structural Requirements Doc (pdf)</td>
<td>APLD/SND</td>
<td>Product specification defining high level structures, number of chiplets, chiplets size, thickness, substrate size, stiffener ring and physical footprint requirements</td>
</tr>
<tr>
<td>Material Property Specification (pdf)</td>
<td>MPS</td>
<td>Package material including UF, Molding, PI, Substrate. etc properties such as Modulus, Posson's ratio, CTE, Tg, Dk, Df, etc</td>
</tr>
<tr>
<td>Verification Specification (pdf)</td>
<td>SA/MECH</td>
<td>Detailed documents defining mechanical verification requirements of the system. Includes detailed stress, warpage, system assembly such as heat sink, stiffener ring verification plan and methods</td>
</tr>
</tbody>
</table>
Output description

1. Stress and strain: The mechanical analysis provides detailed information on the distribution of stresses and strains in the chiplet package, including the magnitude and location of peak values, areas of high stress concentration, and areas of low stress concentration. This information is critical for identifying potential failure modes and areas where design changes could be made to improve reliability.

2. Fatigue life: The analysis provides an estimate of the fatigue life of the chiplet package based on the predicted stress cycles and the material properties of the package. This information is critical for ensuring that the package will perform reliably over its expected lifespan.

3. Thermal analysis: The analysis provides information on the temperature distribution in the chiplet package and the thermal stresses induced by thermal expansion. This information is important for understanding how the package will perform under different temperature conditions and for identifying areas where the design could be improved to reduce thermal stress.

<table>
<thead>
<tr>
<th>Outputs</th>
<th>Persona[table]</th>
<th>Description/notes/comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Refined and Formalized Requirements Doc (pdf)</td>
<td>SA/PA</td>
<td>Product specification defining high level functional, thermal, reliability, and physical footprint requirements based on the mechanical/thermal stress analysis.</td>
</tr>
<tr>
<td>Structural Specification (pdf)</td>
<td>SA/PA/PLV/SILV</td>
<td>Mechanical analysis provides how the package will perform under different temperature conditions and to external forces thus can optimize and finalize the package structures to reduce thermal/mechanical stress and increase reliability</td>
</tr>
<tr>
<td>Integration and Reliability Risk Analysis (pdf)</td>
<td>SA/PA/MECH</td>
<td>Results of the chiplets integration process and reliability risk analysis that provides the distribution of stresses and strains in the chiplet package, including the magnitude and location of peak values, areas of high stress concentration to identify potential failure modes and areas where design changes could be made to improve reliability.</td>
</tr>
</tbody>
</table>

Expected Results

1. **Stress and strain analysis**: The mechanical analysis provides information about the distribution of stresses and strains within the chiplet package under different loading conditions. This information can be used to identify potential failure modes and optimize the package and board design to minimize stress and strain.

2. **Thermal analysis**: The mechanical analysis also provides information on the temperature distribution within the chiplet package under different operating conditions. This can be used to optimize the thermal design of the package to ensure proper heat dissipation and reduce the risk of thermal failures.

3. **Deformation analysis**: The mechanical analysis provides information on the deformation of the chiplet package under different loading conditions. This can be used to optimize the package and board design to ensure proper fit and alignment of the components.

4. **Fatigue analysis**: The mechanical analysis can also be used to predict the fatigue life of the chiplet package under cyclic loading conditions. This information can be used to optimize the design and ensure the longevity of the package.

5. **Determination of reliability**: The mechanical analysis allows for the determination of the reliability of the chiplet package under different operating conditions, such as temperature and mechanical loading. This information can be used to optimize the design and improve the reliability of the system integration.

Figure 23 shows the thermomechanical stress analysis flow chart for chiplets integration by using FEA (Finite Element Analysis)

1. Define the chiplets size, package and board layout, including the position of chiplets, interconnects, and other components (Material and Structural inputs and Drawing)
2. Conduct finite element analysis (FEA) to simulate the mechanical behavior of the system under various loading conditions, such as thermal and mechanical stresses.
3. Use thermal simulation tools to model the thermal behavior of the system, including temperature gradients and temperature cycling.
4. Use FEA to model the effects of thermal expansion and contraction on the system, as well as the resulting stresses and strains for thermal and mechanical stresses analyses (Outputs).
5. Validate the simulation results through testing and iterative refinement of the design.
6. Analyze the results of the simulations to identify potential failure modes and evaluate the system's reliability under different operating conditions.
7. Optimize the package and board design to minimize stress and strain, and maximize reliability.

Figure 23: The thermomechanical stress analysis flow chart for chiplets integration by using FEA. Courtesy of ASE US Inc.
Test Workflows for Multi-die System-in-Package

Introduction

In this chapter, we will cover the recommended workflows for testing chiplet-based System-in-Package (SiP) designs. We will discuss a tool-generic 3D workflow for DFT, IP, and wafer, and SiP package test.

We will start with chiplet test planning considerations, then move on to DFT and IP test implementation at the die level. Planning concepts include coordination with other dies in the package, and test access methods.

After that, we will cover known-good die (KGD) wafer probe DFT and IP test requirements and techniques, including microbump test access considerations.

We will move on to System-in-Package (SiP) testing after die/package assembly. Topics for SiP package test include die-to-die (D2D) interconnect test and repeating the KGD DFT and IP tests at the package level, including die-level test vector application from the package pins.

System-level test with the fully-tested and packaged parts at the board level will be next in our discussion. Topics here will include interconnect testing, along with DFT and IP test access at the board and system level.

Along the way, we will highlight test-related package, power, thermal, reliability and test hardware requirements.

2.5D and 3D Overview

System on a Chip (SoC) technology enabled the expansion from simple devices comprised entirely of I/O, logic, and memories to include third-party IP cores such as serializer/deserializer (SerDes), double data rate (DDR) memory, physical layer architecture (PHY), and special I/Os such as low-voltage differential signaling (LVDS), and positive emitter-coupled logic (PECL). This created a new IP core marketplace. These IP cores necessarily came with design-for-test (DFT) libraries, embedded DFT logic such as scan, or both.

Through creative packaging techniques, SoC has evolved into a system in a package (SiP) where we can instantiate multiple dies inside a single chip package. This has created a new Chiplet market where IP
vendors sell chiplets with specific functions to third-party SiP aggregators who build the SiP with chiplets and their own chips.

**Figure 24: Image courtesy of Siemens EDA**

In a SiP, each die connects to other dies using either 2.5D or 3D interconnect technology. SiPs have the following advantages over multi-chip modules (MCMs) and printed circuit boards (PCBs):

- Better performance
- Smaller size
- More functional capability
- Cost savings

Figure 24 shows some packaging strategies for multi-die systems in package (SiP) strategies. For reference, a single-die in a package is called a 2D design. A 2.5D SiP is composed of 2 or more die’s or “chiplets,” usually mounted to a silicon interposer or some other silicon-based interface. In the past, these were called MCM’s, but current 2.5D strategies may have 10,000 or more I/O buffers that are interconnected through a silicon interposer. Another strategy shown in Figure 24 is a 3D or stacked-die strategy as shown in the center image. In this scenario, Through Silicon Vias (TSV’s) are primarily used as the method of die-to-die communication. With 3D, even more interconnects are possible between dies as compared to 2.5D. The rightmost image shows a hybrid of 2.5D and 3D. This is normally still referred to as “3D,” but some specifications such as IEEE-1838 refer to this configuration as 5.5D.

SiP strategies have distinct advantages over a very large SoC design with the same logic, including:

- Better silicon yield as a result of breaking large SoC functions up into smaller dies
- The capability to manufacture packaged ICs much larger than the maximum die size for the target process. This can help avoid the removal of chip functions to stay within the reticle limit

This work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International
Better profitability as the packaging of bad parts is avoided with Known Good die (KGD) test strategies

SiP challenges compared to a very large SoC include:

- Higher power density, IR drop, and noise
- Thermal load and cooling issues
- Lower package yield due to increased packaging steps and complexity
- Dies must be fully tested at wafer probe before packaging
- Wafer probe is more difficult for SiP chiplets and SoCs than for standard ICs

2.5D Design Overview

Before the advent of System in a Package (SiP), the primary means for interconnecting multiple chips was using printed circuit boards (PCBs), which are considered 2D technology. With 2.5D, we are moving the board, or subsystems of the board, into the IC package.

Figure 25 shows an example with a large primary die. The primary die provides the TDO signal, and the TDI connects to a nearby chiplet. The chiplets are considered to be secondary devices.

![In-package boundary scan for a 2.5D device](Image courtesy of Siemens EDA)
You can also incorporate advanced technologies such as High Bandwidth Memory (HBM). Technologies such as HBM3 have their own DFT and IP test requirements. Some of these test requirements are built into the HBM3 3D memory stack, like DRAM memory BIST. Others, such as I/O BIST and lane repair, are in the base die of the memory stack. IP cores such as HBM3 PHYS are incorporated in the ASICs or SoCs and mirror the test behavior of the HBM3 3D stack base die. Note that 3D HBM memory stacks are delivered as a fully-tested/KGD memory stack to the SiP aggregator. These technologies may require thousands of I/Os between dies in the package. A much finer bump pitch than the standard bump pitch is required to accommodate large numbers of lanes.

Standard substrates cannot accommodate such fine bump pitches, so silicon is used as the substrate. The silicon substrate (or “interposer”) is manufactured in an IC fab. The PCB is moved to the interposer and into the package. You can instantiate many chips on the interposer; thus, the term “System-in-Package”. With this methodology, the circuitry of a conventional large PCB can fit into a relatively small package.

Multi-Chip Modules (MCMs) are a variation of SiPs with multiple dies instantiated on a PCB or ceramic substrate. Most of the 2.5D capabilities described also apply to MCMs, and this documentation highlights any differences where they exist.

This document describes use of the standard DFT insertion flows for the individual dies. Boundary scan insertion capabilities and flows have been enhanced for die-to-die interconnect testing and for Unified BSDL.

Multi-die Test Planning

In some sense, this is the most important step in the 2.5D/3D test flow. Some extra time planning up-front will result in far less overall time in implementing the DFT and IP test insertion flow. We should note here that there are 2 basic requirements for multi-die test:

- Dies-to-die interconnect testing. Typically done with scan or boundary scan. For 2.5D Phy IP’s, it may also include IO BIST with lane redundancy and repair.
- Transport and apply chiplet level DFT and IP test vectors to the dies of interest. This is where using IJTAG can really help to apply the tests to the dies or even blocks of interest for package-level silicon test bring-up or diagnosis.

Note that all of the DFT and IP test logic resides in the chiplets as this DFT section assumes that there are no active components other than chiplets on the interposer or in the 3D stack. As such, we will not focus too much on the Chiplet DFT insertion, as it stays mostly the same as 2D DFT insertion. We will focus on the cases where there are new flows or requirements.
Here are the important aspects for DFT and IP test planning. In the Chiplet Design Exchange and Workflows document, we covered the Chiplet Design Kit (CDK) recommended handoffs from the chiplet vendor to the SiP aggregator, so we will not cover those again here.

- **Die-to-die (D2D) interconnect testing:** Planning criteria include:
  - Common interfaces between dies. For 2.5D this is mainly through specific Phy standards like UCIe, BoW, Open HBI, and HBM2e. The interfaces between dies must match. Other considerations include Phy reversal and rotation so that lanes are not crossing on the interposer with long routes between dies.
  - Clocks and resets are transferred between dies. This is really the same problem functionally, but we may need to bring over an extra scan clock or scan and compression controls. This is where packet based scan can help because the scan clocks and controls are not global signals for packet based scan. They are developed at the block level.
  - For my Phy IP, is there an IJTAG or IEEE 1500 control or side channel between dies. If so, we must be sure to allocate enough I/O’s. Please note that this control channel is for D2D Phy interfacing and is separate from the FPP or IJTAG interfaces.
  - DFT or IP test ECO strategy for netlist or RTL level. For example, know how to add a memory to a shared MBIST controller late in the design cycle.
  - Any binning strategies for mixing chiplets using different process technologies.
  - Chiplet: Be sure to have a core wrapper scan chains that only interact with the die’s IO’s or TSV’s, and not core-to-core interconnects within the die.. These wrapper chains will be used for Die-to-Die scan testing.
  - The test IO must match for each chiplet. For example, if we are doing standard compressed scan, we must not exceed the number of bits in the FPP bus. This may require the chiplets to be tested in “chunks” by reusing the available IO for several passes of ATPG. This is an area where using packet-based scan can really shine. All of the chip logic can be tested simultaneously with a fixed set of IO from the FPP bus. The overall effect is lower test time, volume, and reduced cost.
The recommended 2.5D DFT Insertion workflow can be seen in Figure 26. Again this is all done at the Chiplet or die level.

- In the first step, we insert as much DFT logic as possible at the RTL-level. This allows us to “shift left” for DFT insertion, which allows us to see the impact of DFT very early in the design cycle. Note that most DFT tools can operate at both RTL and netlist levels. Netlist is required by ASIC suppliers as their functional design customers may not want to share their proprietary RTL. The EDA tools also generate SDC timing constraints for all the DFT logic generated here. This should include “disabling” constraints so that we do not try to close timing between non-interacting DFT and functional logic.
- In the next step we insert the supporting structures for scan testing into the design. SDC timing constraints for the supporting scan structures are also generated by the EDA tools here.
- During the synthesis step, we take the RTL and SDC timing constraints from the previous steps and synthesize. Initial min/max SDF may also be generated at this step for simulation and constraint checking.
- In the next step, we take the synthesized netlist with supporting scan structures and other DFT, and we generate the scan chains and stitch them to scan decompression/compression engines.

---

This work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International
Note that we run complete verification at each DFT insertion step above. These include:

- Clock-domain Crossing checks (CDC) and Reset Domain Crossing checks (RDC)
- RTL-to-RTL Equivalence check on the functional path for RTL-level DFT insertion.
- RTL-to-netlist functional equivalence check for synthesis
- Netlist-to-netlist functional equivalence check for post scan insertion and after physical design.
- Unit-delay simulations for RTL simulations.
- Min/max SDF simulations for post synthesis and post-layout timing. This can be useful for constraint checking for IP test, as STA will sometimes pass with plenty of margin, but fail during simulation due to an unexpected path.
- STA with On-chip variation (OCV) and noise for post-synthesis/scan insertion and post layout DFT modes.
- Optional SAIF simulations for power/activity factor
- Static and dynamic IR-drop analysis which is important because of the very high activity factors during DFT and IP testing.

We also need to redo the simulation steps at the package level with all the dies active. This is actually a fast simulation because we “graybox” all of the die-level block contents, and only leave the boundary scan, outward-facing (EXTEST) wrapper chains, IJTAG, and packet-based scan bus/logic in place. These are necessary for Die-to-die interconnect testing.

DFT in 3D Designs

The IEEE 1838 standard defines the test access architecture for 3D stacked integrated circuits. In addition to the test access architecture, the IEEE 1838 standard also sets requirements for test access. You should create the test access with a mandatory serial test access interface and a parallel interface (optional for the standard but practically required for real designs). The serial access uses the Primary Test Access Port (PTAP) and the Secondary Test Access Port (STAP). The PTAP and STAP control the test and debug features in each die. The Multi-die flow uses the PTAP/STAP interface as the die-to-die IJTAG interface. The parallel access is called the “Flexible Parallel Port” (FPP). The FPP covers the multi-bit test access between the primary and secondary interfaces and the core functional logic in the die. The FPP is also the interface test access for the die stack under test.

The individual dies use the standard DFT insertion flows. IJTAG insertion can insert the PTAP and STAP for serial IJTAG support, and the FPP is implemented with the packetized scan bus. Wrapper analysis reuses physical block wrapper cells and incorporates them as the Die Wrapper Registers (DWR). The DWR enables you to perform die-to-die scan captures to test the through-silicon vias (TSV) connections between dies.

Upper dies connect only to other dies in the stack with TSVs. The upper die ports are typically not equipped with full-sized bumps and Boundary Scan and use fine-pitch micro-bumps with TSVs for die-to-die interconnect. The wrapper cells provide die isolation. You must add probe pads for at least the TAP interface IOs and packetized scan bus for Known Good Die tests. These sacrificial probe pads are not used after stacking.
Figure 27 shows a design with a base die and two upper dies. The base die provides the test interfaces to the upper dies in the stack. The base die also contains the IO buffers. Note that the base die does not have to be the bottom die, but it usually is. All dies above the base die in the diagram are considered the upper dies. The DWR is the die’s external mode (EXTEST) wrapper chains that interface to the die’s ports. Note that the top die does not have 2 DWR's it is missing the top DWR. If these are identical dies, we will need a bypass for the upper DWR.

**IEEE 1838 compliant 3D DFT Architecture**

Figure 27. 3D Design Stack with DFT. Image courtesy of Siemens EDA
Note that you can use 3D and 2.5D methodologies together in the same package. For example, creating a 3D stack on top of the die used in 2.5D creates a 5.5D design (per the IEEE 1838 standard), which benefits from both methods.

3D DFT Workflow

Figure 28: The 3D test insertion workflow. Image courtesy of Siemens EDA

The 3D DFT and IP test insertion workflow is very similar to the 2.5D workflow shown in Figure 28 so we will highlight the differences as described below:
• The IEEE 1149.1/1149.6 boundary scan insertion step is only done on the base die. The rest of the die-to-die interconnect testing is done through the directly-bonded TSV's that connect to the upper die.
• We insert the PTAP/STAP shown in Figure 28 during IJTAG insertion for the first DFT insertion step.
• The FPP is inserted by using the packet-based scan bus from the second DFT insertion step.

Purpose of the multi-die flow

The goal of SiP-level hierarchical DFT Architecture is to enable cost-effective manufacturing of chiplet based designs in the ATE test environment. This includes ICs and systems built by stacking dies in vertical dimensions (3D) or 2.5D ICs using an interposer, or a combination of these two advanced packaging techniques(5.5D). A new DFT methodology approach for heterogeneous integration of chiplets in one package is needed to achieve quality and test cost reduction in a high-volume manufacturing environment.

Expected results

Example chiplets could be a memory or a logic die. Each chiplet needs different DFT and IP test techniques. For example MBISTand repair is needed for Memory die and scan or LBIST (Logic BIST) for Logic die. A chiplet may also need IP tests like SerDes loopback both at the die level and at the SiP level. A Hierarchical DFT architecture and flow must be followed for multi-Die testing to be practical.

IEEE 1149.1, IEEE 1687, and the subordinate IEEE 1500 standards have been successfully used in PCB’s, multi-core processors and complex SoCs that integrate custom logic and 3rd party IP such as PCIE, USB, HBM, DDR-PHY, PAM4 SerDes and embedded microprocessors to enable structural Scan Tests and BIST. IEEE 1687 standard is comprehensively used to trigger internal DFT resources/instruments added per chiplet to perform the designed operation such as comprehensive memory Built-In Self-Test and scan out results using Test Access Port (TAP). IEEE 1838 is the latest standard for Test Access Architecture for Three-Dimensional stacked integrated circuits. It mandates a die wrapper register with which one can perform interconnect tests for static as well as performance related tests.

Quality Metrics are important and test coverage is the direct measurable statistic to prove the quality targets are met. This coverage is to be determined over the number of test types and test type covers different defects that can be prevalent during SiP assembly with constituent chiplets. At Chiplet level, stuck-at test coverage (>99%) and at-speed test coverage (>95%) numbers are very important to guarantee the package level test coverage goal is met. In addition to these, due to aggressive scaling of...
process nodes and increasing number of chiplet integrations, the chiplet level bridging fault coverage, cell aware static and dynamic test coverage numbers play a very critical role to achieve package level quality. Even higher quality metrics are important for safety-critical applications like ISO 26262 Automotive test.

Security is an important criterion in today's chips so DFT needs to provide custom designs to ensure the testability and security requirements are both met satisfactorily. This requires special attention especially since the internal state of the chiplet can be illegally accessed by the hacker using scan or boundary scan access.

Flowchart

The SiP Test Flow includes IEEE 1149.1/1149.6 compatible chiplets with IEEE 1687 or IEEE 1500 compatible IP cores using Memory BIST and internal instruments to enable Process/Voltage/Temperature (PVT) Sensors, PLL BIST Engine, IO BIST. IEEE 1687 is the preferred interface for IP cores because of ICL (Instrument Connectivity Language) and PDL (Procedural Description Language) that can be provided in the IP core design kit. With PDL, IP-level tests like SerDes loopback can be delivered with the IP, and read by IJTAG tools and then automatically generate chip top-level IEEE 1450 STIL test vectors and simulation testbenches. With IP-level PDL and ICL, this becomes a well automated process and not an end-user exercise to map an IP-level testbench up through the hierarchy to the chip top level, while performing an eVCD dump to get patterns that may be translatable into tester format. These IP Cores and/or Chiplets are to be designed with embedded On Chip Clock Control (OCC), and Hardware-based Scan Compression/Decompression for comprehensive ATPG and test/fault coverage. Packetized scan is also very useful for providing an inherent FPP, along with the capability to test any block, die, or full stack simultaneously. Some packet-based scan can also make a huge improvement in scan-related power, with significant reduction in false failures/yield loss due to IR-drop.

Top Level (SiP Level) test controller need to enable all the built-in DFT resources described above in addition to activating 3rd party IPs such as PCIe, USB, DDR-PHY, Serdes and ARM to enable structural Scan Tests and BIST. The test fabric based on IEEE 1687 standard is comprehensively used in the flow and makes use of Test Access Port (TAP defined in 1149.X standard). IEEE 1838 is the latest standard for Test Access Architecture for Three-Dimensional stacked integrated circuits. It mandates a die wrapper register with which one can perform an interconnect test. It can be flexible to be used as a resource of Interconnect BIST

Manufacturing Test
A significant portion of overall device cost is consumed by test. These test costs include silicon overhead of test, and DFT structures, test time, and labor/NRE costs. The majority of device cost is consumed by other dies, mask costs, and advanced packaging.

The main purpose of SiP test is to screen-out defective dies so they are not assembled with other good dies, good interposer, and expensive multi-die packages. The effect of assembling a SiP with a defective die is that we wipe out the whole SiP.

A major aspect of manufacturing test is Yield. A major driver of yield is area. We can disaggregate large dies with low yield into smaller dies or chiplets with high yield, then assemble fully tested dies into advanced SiP packages with other “good” chiplets.

Types of manufacturing test

Wafer probe

The purpose of wafer probe is to provide high confidence to assemble and package the part with minimal risk due to undetected defects. At wafer probe, we take full wafers and touch down on each die with a wafer probe card. Test is normally run at room temperature for wafer probe. The probe card provides the test data to the chiplet die, as well as power and ground for each power domain. The location of each passing and failing die is stored, and the bad dies are disposed of after sawing the wafers into individual dies.

The probe card has some limitations:

- High RLC compared to the SiP package. This results in more loading on the device or chiplet under test, or DUT.
- Higher RF noise environment compared to the SiP.
- Reliability. The probe card reliability is a function of number of pins and current density per pin
- High current can “burn” the tips of probe pins, so careful power analysis for each test is important.

The probe card includes “probe pins” that contact the wafer “probe pads”. These probe pins have minimum physical spacing limitations or “probe pitch” requirements. However, chiplets use fine-pitch “micro-bumps” for die-to-die interconnect. Probe card technology that will reliably support microbump spacing is currently in development. Special “sacrificial pads” are used to work around the probe card pin spacing limitation until reliable probe cards with microbump spacing are widely available in the SiP industry. These sacrificial probe pads have standard probe pin pitch or spacing which has proven to be reliable. These sacrificial probe pads are not used for functional die to die interconnect, hence the term,
sacrificial pads. Sacrificial pads are only used only for test. For SiP package interconnect, we only use the microbumps.

As shown in Figure 29, sacrificial pads are muxed for die test inputs, and shorted for test outputs. A sample of shorted power and ground is required. Burnt probe tips and damaged probe cards are a real concern because we are only probing a sampling of power and ground for each supply. This means that the current per probe tip will be higher than normal. Again, careful power analysis like Static/dynamic IR-drop requirement is needed to prevent false failures due to IR-drop which can severely impact chip yield and overall project profitability.

Figure 29: Sacrificial pads for chiplet KGD wafer probe. Image courtesy of Siemens EDA

High defect coverage is required from wafer probe because we are assembling the chiplet dies into the SiP package with other “good” dies. This means we have to be more thorough with wafer probe testing than we normally need to be for standard 2D parts. This extra thorough wafer probe testing is referred to as “Known Good Die” or KGD testing. KGD test includes all DFT and IP tests. For Phy testing, a test mode called Internal Loopback is recommended. The goal of KGD testing is to run the full package test program
at the die level. Burnt probe tips are even more of a concern as we have only a sampling of power and ground for our KGD probcard.

Package test

After KGD wafer probe, the known-good dies or chiplets are assembled into the SiP package. This can introduce a new level of defects from handling and assembling the parts. This requires a new set of tests for die-to-die (D2D) interconnect testing. Along with D2D test, there may be extra or spare"lanes". A lane is composed of Die-level IO’s and D2D interconnect though TSV’s or silicon interposer. The spare lanes allow "lane repair" by mapping-in the extra or spare lanes. There is typically a ratio of spare lanes for each group of lanes. Typically, an at-speed D2D test is run and compared to an expected response. If there is a mismatch, a spare D2D lane is mapped-in to replace the defective lanes. The SiP is marked as defective if there are not enough spare lanes in the current lane group to repair the defective lanes. Other D2D tests may include slow speed tests like IEEE 1149.1/1149.6 Boundary Scan test or D2D stuck-at scan testing.

IP Test

3rd party IP cores may be present in chiplets, just as they are in 2D devices. These include high-speed serial phy’s like PAM4 Serdes, PCIe, and USB, and USR/XSR SerDes for die-to-die communications. High-speed serial interfaces typically use less than 512 differential pairs per chip or chiplet.

The Chiplet may also include parallel phy’s like HBM2e/HBM3, BoW, UCl-e, and Open HBI. These may have thousands of I/O’s per chiplet. The I/O’s for the parallel Phy’s are for die-to-die communications, so they are not accessible at the package bumps of the SiP.

In general, parallel phy’s have unidirectional inputs or outputs. This poses a special problem for KGD testing. Also, because there are so many I/O, there really is not enough room for sacrificial pads for parallel Phy I/O’s, so [parallel phy I/O's can likely not be directly probed during KGD wafer test. For this reason, it is recommended to make every I/O a bidirectional I/O buffer. In this way, we can use a PRBS to generate patterns through a means called to-pad internal loopback during KGD wafer test . The I/O buffer would ideally have Input Enable and Output Enable control pins that can default to their functional I/O buffer direction. During KGD test, we have the ability to drive pseudo random through the output of the I/O buffer, to the pad, and back-in through the input side of the I/O buffer. We can then compare the data that was transmitted, with the data received. That should be a good KGD to the thousands of I/O buffers that would otherwise go untested for KGD test. Note that Serdes, PCIE, and serial USB normally would have the ability to do something similar using a mode called Near-end Internal loopback. Be careful not to configure the Serdes to require external wire connections for loopback. We call such a test “I/O BIST.”Most serial phy’s will have the necessary PRBS and comparison logic (like a MISR) already built into the IP core. For both Serial and parallel Phy’s, some level of calibration will be required if you want to run KGD test at high data rates. Note that it may be difficult to get a true at-speed loopback test due to
high jitter in the wafer probe environment. This may preclude getting a high quality eye diagram due to the wafer probe jitter/noise.

As mentioned earlier, parallel phy’s may have many thousands of I/O’s. These phy’s are the die-to-die communication channel between chiplets. An example is an open connection between an I/O in chiplet A that is connecting to an I/O in chiplet B. We call that connection a “lane”. The die-to die lane connections can be made across silicon interposer (2.5D) or by Through-Silicon Vias (TSV’s). Assembly issues that break connections in lanes may occur frequently enough to impose a significant yield loss. Lane redundancy and repair is used to help mitigate assembly yield loss for SiP devices.

The parallel phy developer may provide “spare lanes” in a small ratio to the functional lanes. These are defined by the Phy design, but may also be defined by the relevant standard for the parallel phy.

At the package test, we can run IO BIST in a “far-end external loopback” configuration, across the functional lanes to the other die and back to the transmitting chiplet where the responses from the functional lanes are compared. If the loopback test fails, something in the lane is bad. We then map-in a spare or “redundant” lane to bypass and replace the “bad” lane that is failing far-end IO BIST lane loopback. The SiP is marked as bad if there are so many failures that we do not have enough spare lanes to repair the SiP.

As with memory repair schemes, an efuse or OTP may be used to store the lane repair signature. That efuse is then unloaded during the SiP boot sequence and the lane repairs are made. This process is called “Hard lane repair”and is defined during the manufacturing test as it is intended to improve part yield and profitability.

If we still have remaining redundant lanes, we can use them to improve art reliability over the product lifecycle. We do this by running IO BIST and lane repair as we functionally boot-up the chip in the system, but we store any new lane failures and remapping signature in registers, in addition to the efuse from manufacturing test. This way, we can repair any new lane failures that occur over the SiP lifetime.

Other tests

Other tests include HTOL reliability testing. This test is commonly known as burin-in. During HTOL, we take many packaged SiPs and put them in a “burn-in oven” for a period of time like 2000 hours. The temperature and/or voltage is raised in the oven to effectively accelerate the SiP lifetime. For example, 2000 hours in the oven may be equivalent to 10 years of part aging. Slow speed tests are continuously run in the oven. At-speed tests are not recommended as they may cause unpredictable part heating and accelerate the part lifetime or cause failures. During burn-in an output is typically monitored for “life testing” to ensure that the part is still toggling and alive. If the monitored output stops toggling, the affected part is removed from the oven and tested with the full package test to determine what failed. If needed, the part is sent to the Failure Analysis lab to determine the failure mechanism. Also, parts are periodically removed from the burn-in oven, and the full SiP package test is run.
Another important step is run before releasing the SiP for high volume production. This step is known as test characterization. The parts are tested across process, voltage, and temperature to provide a stable test program. Each test is individually tested and characterized across many parts. Special test dies that force the pmos/nmos process corners are used. Typically the following corners are used for characterization:

- Fast/Fast
- Fast/slow
- Slow/Fast
- Typical/Typical
- Slow/Slow

The corner parts are shmooed across process, voltage, and temperature (PVT).

Inputs/Outputs

<table>
<thead>
<tr>
<th>Inputs</th>
<th>Persona</th>
<th>Description/notes/comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operations Requirements</td>
<td>DA</td>
<td>Closely works with SA and the marketing requirements to study the use case and define SiP level DFT architecture</td>
</tr>
<tr>
<td>Test Coverage and Test Cost Requirements</td>
<td>DA</td>
<td>Includes usage scenarios, and Stuck-At , Transition as well as advanced fault models. This also includes DFT power modes.</td>
</tr>
<tr>
<td>System DFT Scenario including DFD and SDFT</td>
<td>DA, SDFT</td>
<td>Includes clock freeze, RAM dump requirements</td>
</tr>
<tr>
<td>Outputs</td>
<td>Personal (table)</td>
<td>Description/notes/comments</td>
</tr>
<tr>
<td>---------------------------------</td>
<td>------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Scan Architecture Design</td>
<td>DA, SDFT</td>
<td>Max currents are used by PI to analyze package IR drop, EM limit compliance, and current crowding. Current ramps are used by PI engineers to design and analyze system power supplies, and amount and placement of decoupling capacitors.</td>
</tr>
<tr>
<td>Memory BIST Architecture</td>
<td>DA</td>
<td>Includes advance Memory fault algorithms for comprehensive coverage as well Repair requirements for yield improvement</td>
</tr>
<tr>
<td>IO Test Architecture</td>
<td>DA</td>
<td>Includes IO BIST, Loopback Tests for Serdes</td>
</tr>
<tr>
<td>Analog Modules Testability Plan</td>
<td>DA, SDFT</td>
<td></td>
</tr>
<tr>
<td>D2D Interface Test Plan</td>
<td>DA</td>
<td>Includes Interconnect Test Algorithms and TSV tests at SiP level</td>
</tr>
</tbody>
</table>
Conclusion

This whitepaper has provided a detailed exploration of the evolving landscape in chiplet-based system design, marking a significant shift from traditional SoC methodologies. By examining the entire lifecycle, from system planning and co-design to implementation and verification, the paper has underscored the importance of adapting to new design paradigms brought forth by chiplet technology.

The advent of chiplet design and packaging manufacturing heralds a new era in semiconductor development. It offers a promising avenue to sidestep the deceleration of Moore's Law and the constraints associated with large, monolithic SoCs. However, this advancement is not without its challenges, requiring innovative approaches in areas like STCO, thermal and mechanical analysis, and expanding the scope of traditional chip-level subflows.

Looking ahead, the chiplet-based system design is poised to revolutionize the semiconductor industry, offering more optimized, scalable, and domain-specific solutions. The methodologies and insights presented in this whitepaper are foundational for navigating these changes. Continued research and development in this area will be vital for maintaining momentum and exploiting the full potential of chiplet technologies.

In conclusion, this whitepaper serves as an essential guide and reference for professionals and enthusiasts in the field. As we embrace the new chiplet system era, the strategies and methodologies outlined here will be crucial in driving innovation, efficiency, and customization in system design. The future of chiplet integration workflows is bright, with vast opportunities for technological breakthroughs and applications.
References


2. “Proposed Standardization of Chiplet Models for Heterogeneous Integration”, Open Compute Project, 2022, white paper

3. OCP Chiplet Data Exchange Markup Language (CDXML)

4. JEDEC JEP 30 Part Mode and OCP CDXML Integration

5. Field Solver

6. ADAS 1

7. ADAS 2

8. ADAS 3
License

OCP encourages participants to share their proposals, specifications and designs with the community. This is to promote openness and encourage continuous and open feedback. It is important to remember that by providing feedback for any such documents, whether in written or verbal form, that the contributor or the contributor's organization grants OCP and its members irrevocable right to use this feedback for any purpose without any further obligation.

It is acknowledged that any such documentation and any ancillary materials that are provided to OCP in connection with this document, including without limitation any white papers, articles, photographs, studies, diagrams, contact information (together, "Materials") are made available under the Creative Commons Attribution-ShareAlike 4.0 International License found here: https://creativecommons.org/licenses/by-sa/4.0/, or any later version, and without limiting the foregoing, OCP may make the Materials available under such terms.

As a contributor to this document, all members represent that they have the authority to grant the rights and licenses herein. They further represent and warrant that the Materials do not and will not violate the copyrights or misappropriate the trade secret rights of any third party, including without limitation rights in intellectual property. The contributor(s) also represent that, to the extent the Materials include materials protected by copyright or trade secret rights that are owned or created by any third-party, they have obtained permission for its use consistent with the foregoing. They will provide OCP evidence of such permission upon OCP's request. This document and any "Materials" are published on the respective project's wiki page and are open to the public in accordance with OCP's Bylaws and IP Policy. This can be found at http://www.opencompute.org/participate/legal-documents/. If you have any questions please contact OCP.

About Open Compute Foundation

The Open Compute Project Foundation is a 501(c)(6) organization which was founded in 2011 by Facebook, Intel, and Rackspace. Our mission is to apply the benefits of open source to hardware and rapidly increase the pace of innovation in, near and around the data center and beyond. The Open Compute Project (OCP) is a collaborative community focused on redesigning hardware technology to efficiently support the growing demands on compute infrastructure. For more information about OCP, please visit us at http://www.opencompute.org

This work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International