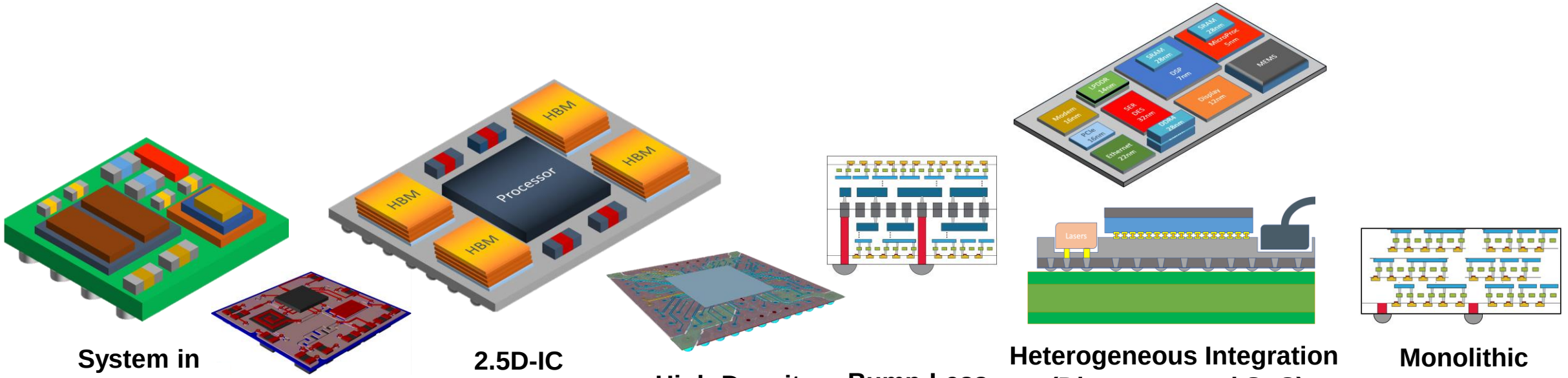
System-Level Design Assembly and Management for 3D-IC Architectures

Brian Jackson

Product Management Director - Cross Platform Solutions

cadence®

Evolution of Advanced Multi-Chip(let) Packaging Technologies



System in Package (SiP) — 1998

RF Module — 2004

2.5D-IC (Silicon Interposer) — 2010

High-Density RDL/FOWLP — 2016

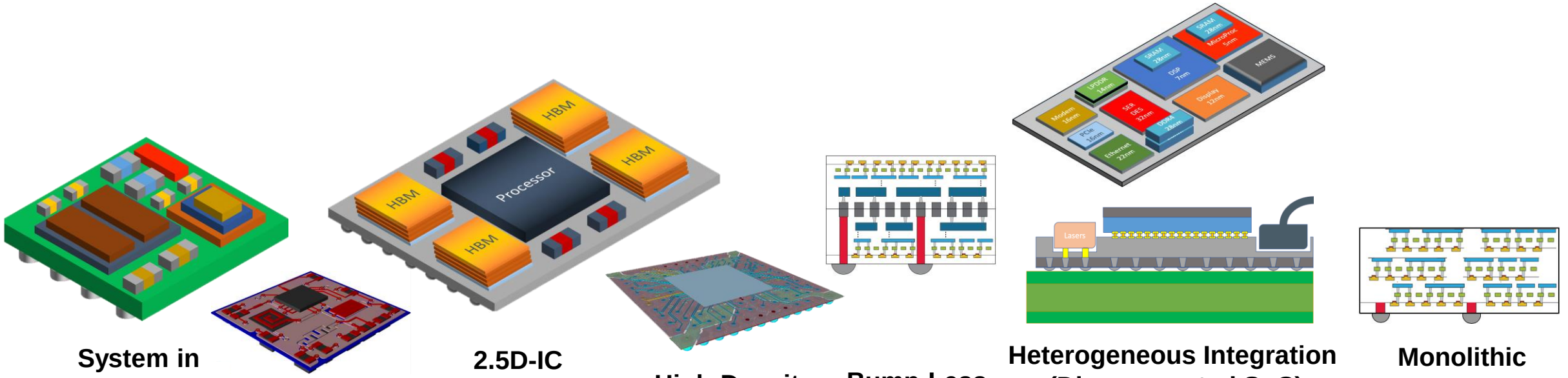
Bump-Less 3D-IC — 2018

Heterogeneous Integration (Disaggregated SoC) Photonics — Now

Monolithic 3D-IC — Future

Evolution of Advanced Multi-Chip(let) Packaging Technologies

New Technologies Require System Design Planning and Management



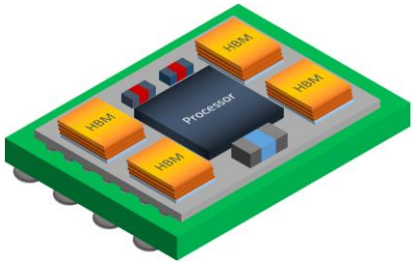
System in Package (SiP)	RF Module	2.5D-IC (Silicon Interposer)	High-Density RDL/FOWLP	Bump-Less 3D-IC	Heterogeneous Integration (Disaggregated SoC) Photonics	Monolithic 3D-IC
1998	2004	2010	2016	2018	Now	Future

Outline - 3D-IC System Design Planning and Management

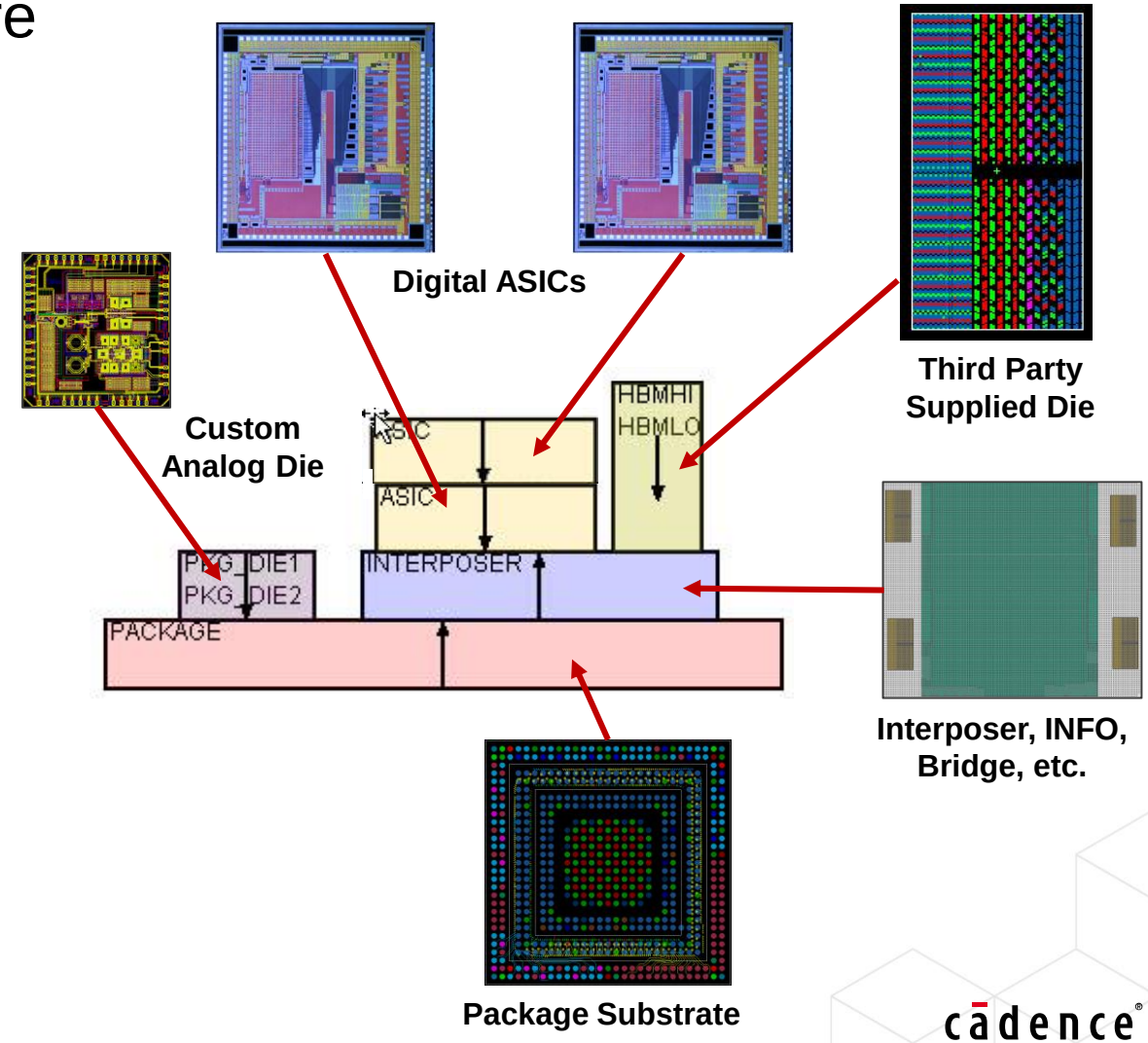
Outline - 3D-IC System Design Planning and Management

- System Design Aggregation and Management
- Manage Logical and Physical Relationships between Die / Package
- TSV Creation and Power Distribution Planning
- Integrated Implementation and System-Level Analysis

Assemble the Complete System-Level Design - Chip(lets), Package and PCB

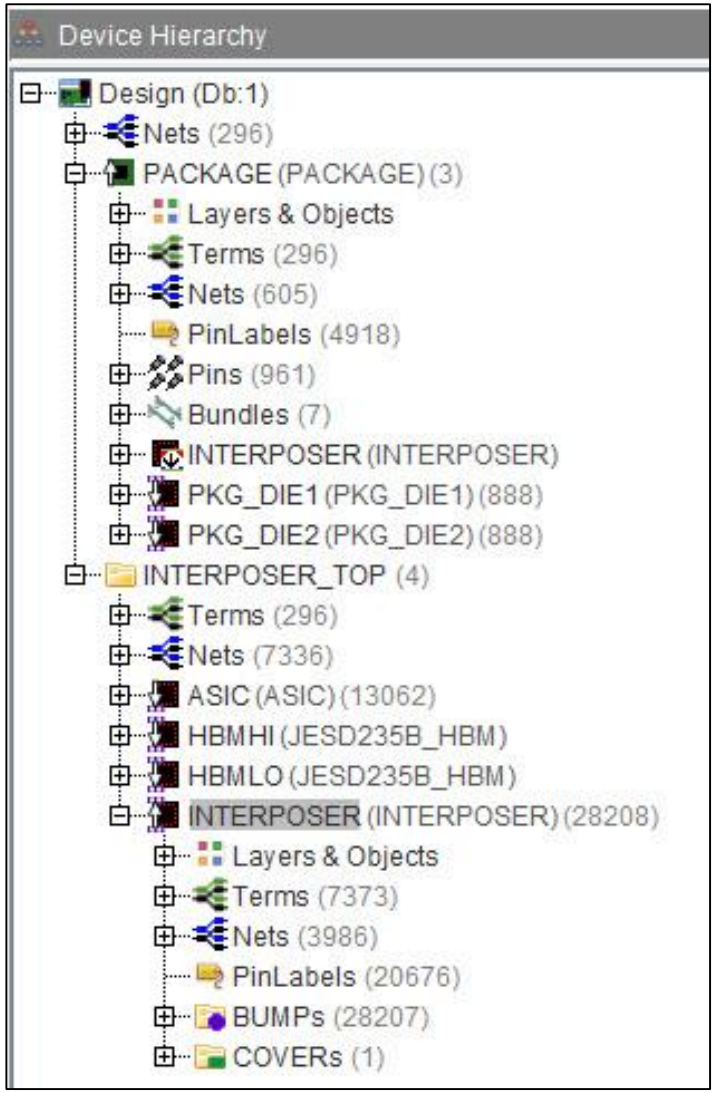


- Define design 3D stack and logic structure
  - Define hierarchy to group die for each stack
- Define layer stackup for each substrate
  - Import tech LEF, CSV
  - Create interactively
- Import or create die and package
  - Utilize integration with layout tools
  - LEF/DEF, CSV, TXT, vendor defined format
  - Ability to create an empty Interposer or Package
- Define or import connectivity
  - Import die or top-level Verilog netlist(s)
  - Import net mapping CSV files
  - Interactively propagate and map nets



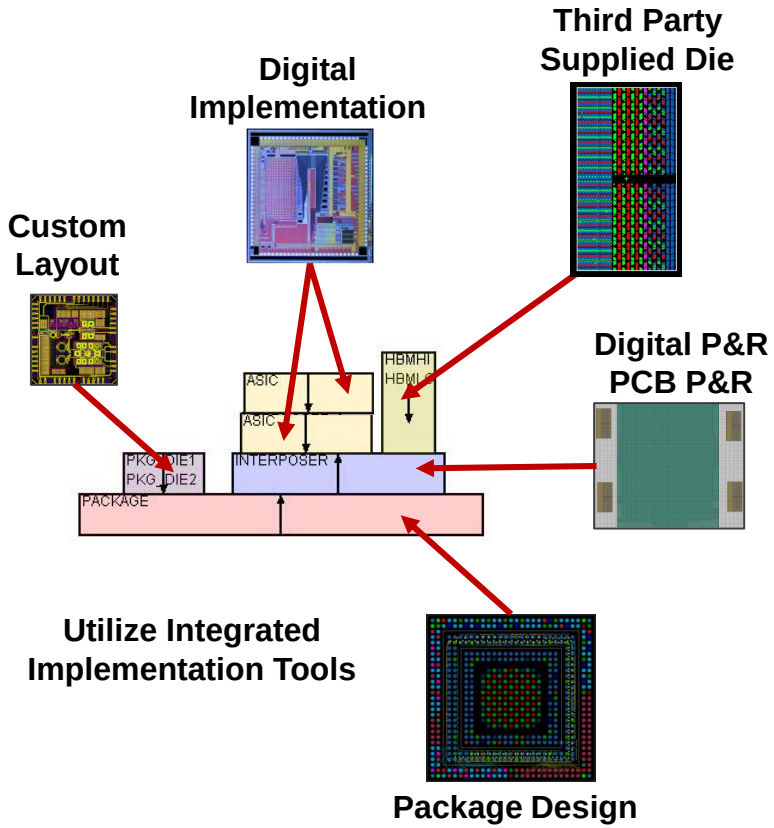
Co-Design Chip(let)s, Package and PCB

- Define and manage the system-level design
  - Define design logical structure
- Manage each design independently
  - Define netlist, bumps, TSVs, etc.
  - Import/Export each die and system data
  - Support specific layout tool data requirements



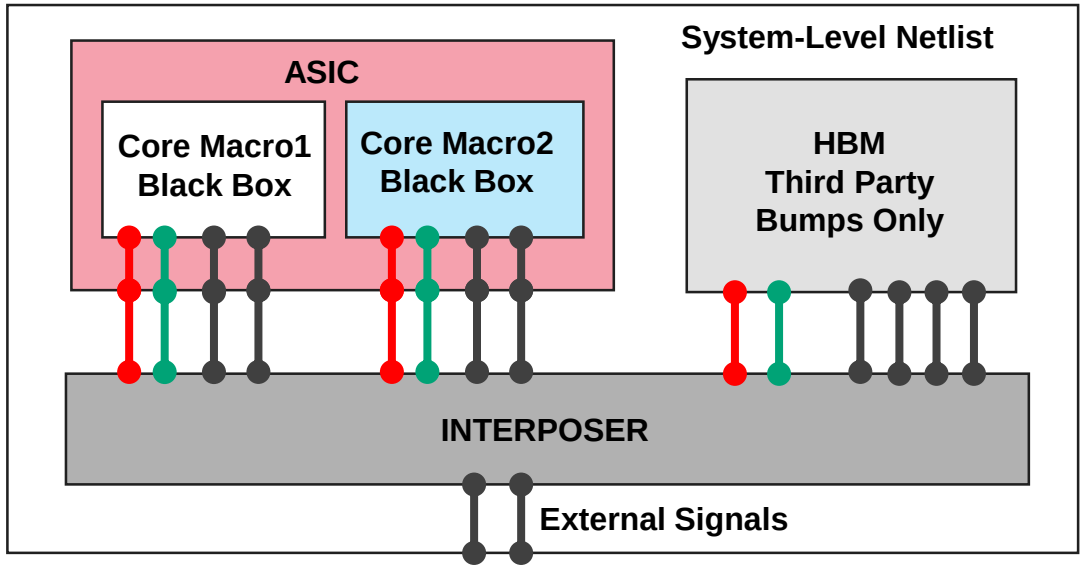
Co-Design Chip(let)s, Package and PCB

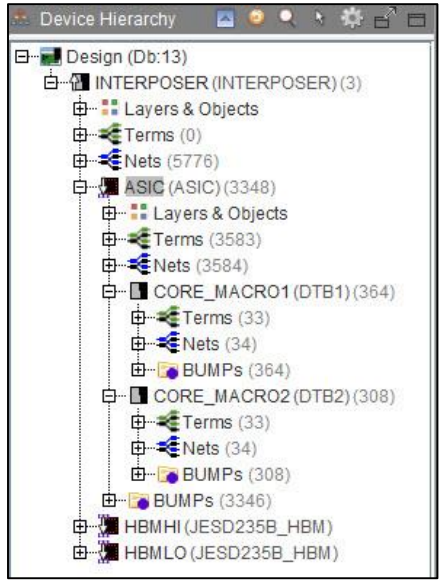
- Define and manage the system-level design
  - Define design logical structure
- Manage each design independently
  - Define netlist, bumps, TSVs, etc.
  - Import/Export each die and system data
  - Support specific layout tool data requirements
- Utilize integrated layout and analysis tools
  - Digital P&R, Custom Layout, Package Design
  - System Level Analysis
- Perform and manage die to die interconnect changes
  - Drive layout tools with ECO changes
  - Store and manage design versions



Manage Chip(let) Interfaces at an Abstract Level

- Netlist contains interface logic only
  - Die contains black box core macros
- Simplify the design
  - Improve tool performance
- Allows IC layout designers to own core logic and layout
  - Separate design responsibilities and data
  - Design in parallel and simply manage interface changes



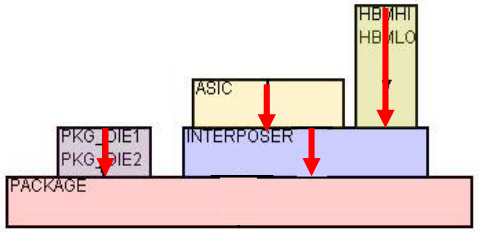


Work with Structural Netlist Only

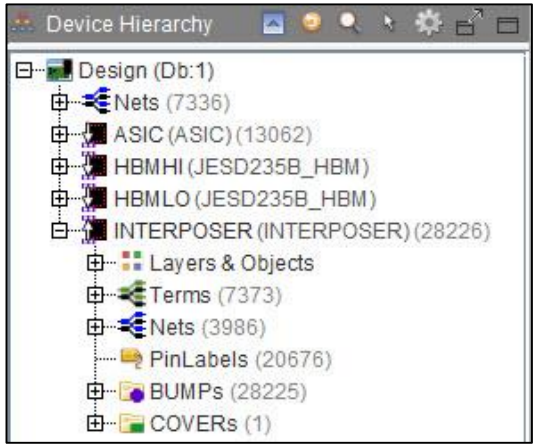
Design and Optimize Passive Interposer and Package

- Derive Netlist(s) from the Die
  - Propagate and map nets to Interposer and Package design
  - Map chip(let)-to-chip(let) connections
- Create contact pads on substrate
- Create and optimize bottom bumps/balls
  - Create larger pad and spacing bump patterns
    - Create power and ground cover patterns
  - Optimize assignment based on die bumps
    - Improve routing and performance
- Export designs ready for implementation
  - Drive integrated layout tools

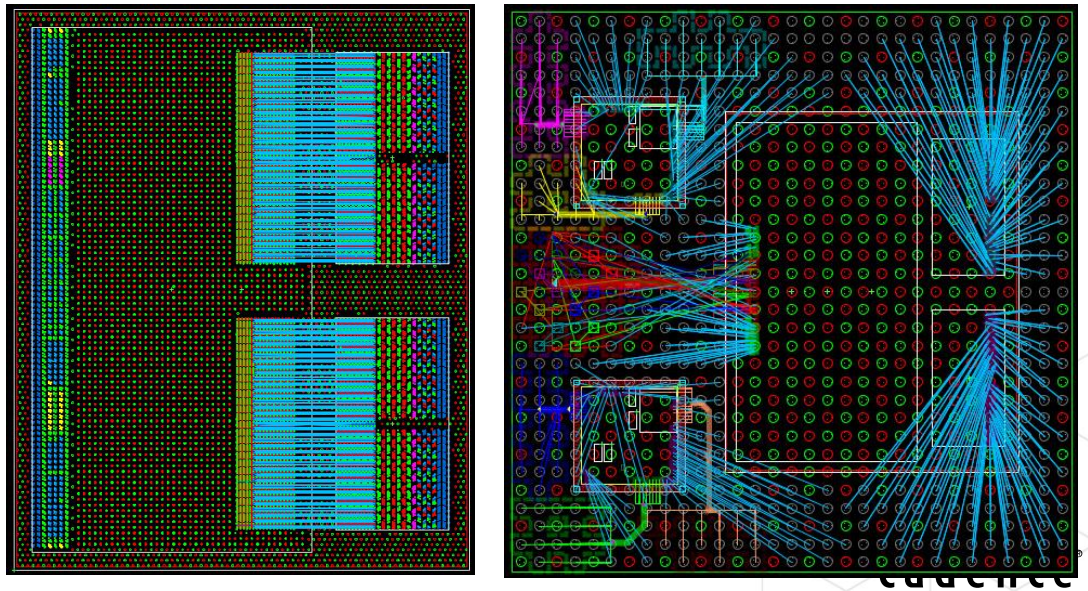
Drive Netlists from Die



Propagate and Map Nets



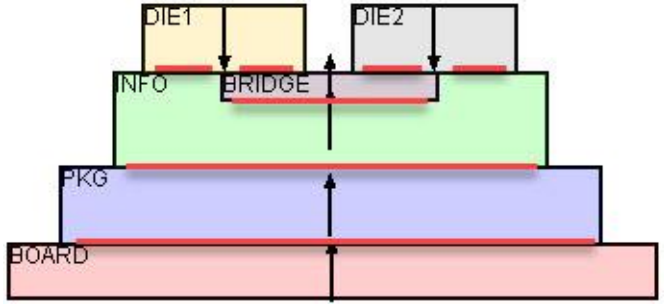
Create and Optimize Bump/Ball Assignments



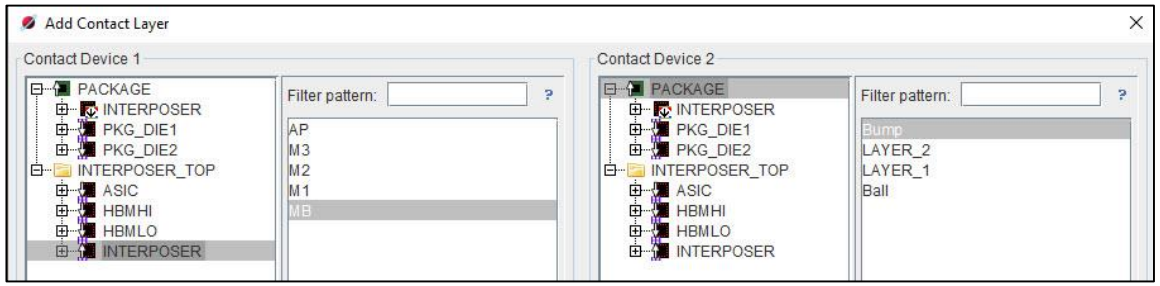
Outline - 3D-IC System Design Planning and Management

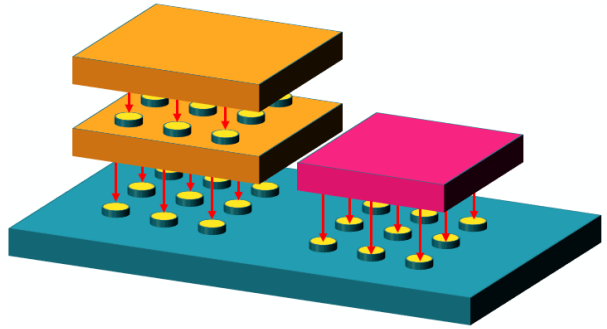
Create and Manage Physical Contacts between Multiple Chip(let)s

- Define design stack and contact layers
  - Define chip(let) orientation and direction
- Support complex die arrangements
  - Bridges
  - Embedded substrates
- Create and manage contact pads
  - Support multiple pad sizes
  - Support various pad shapes
  - Ability to utilize multi-pad LEF macros
  - Move contact pads with die



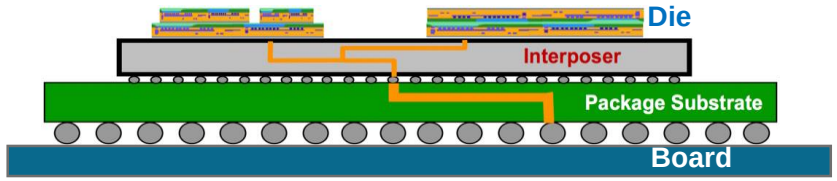
Define Stack and Contact Layers





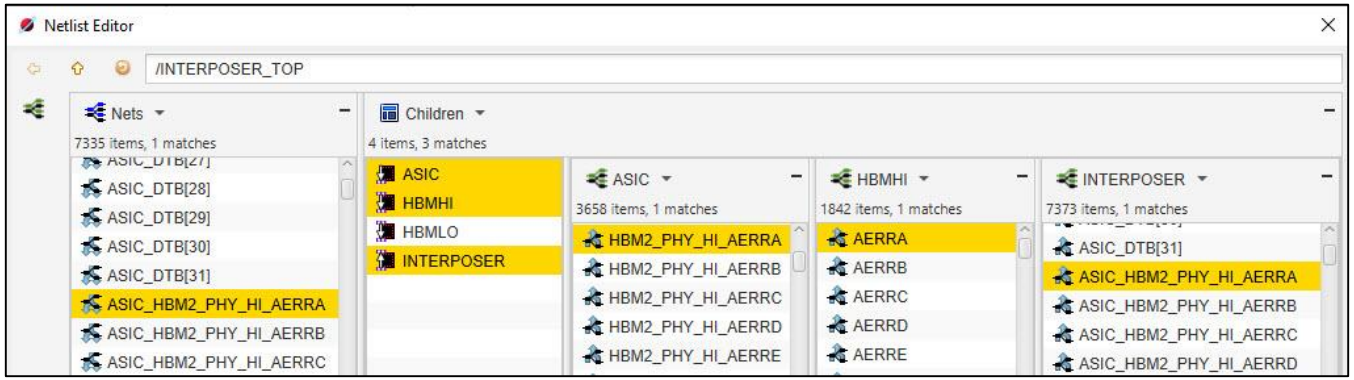
Automatically Create Contact Pads

Establish Logical Relationships between Chip(let)s

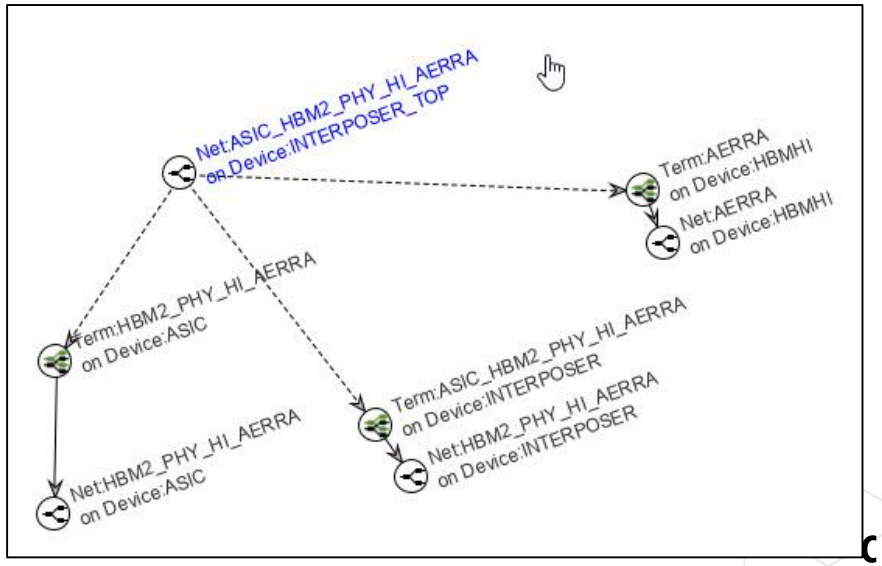


- Define netlist structure and design connectivity
  - Robust interactive net propagation and mapping
  - Import Verilog, CSV, …
- Manage die and system-level netlist
  - Visualize system connectivity
  - Maintain original die netlists
  - Can include die macros

Map System Connectivity



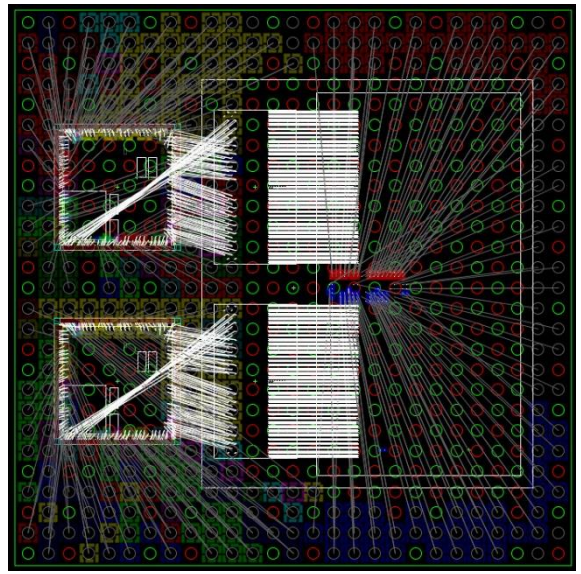
Visualize and Trace System Connectivity



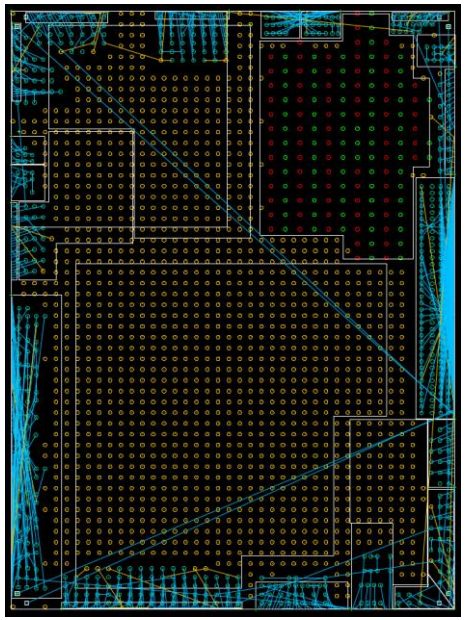
View and Optimize System Connectivity

- System-Level Floorplanning
  - Robust placement features
    - Move, Copy, Flip, Align, …
  - Optimize system connectivity
    - Adjust orientation based on system connectivity
  - Interposer C4 bump and Package ball creation and assignment
- Macro Level Floorplanning
  - Automatic bump creation
  - Optimize die to die connections
  - Create power and ground bump patterns

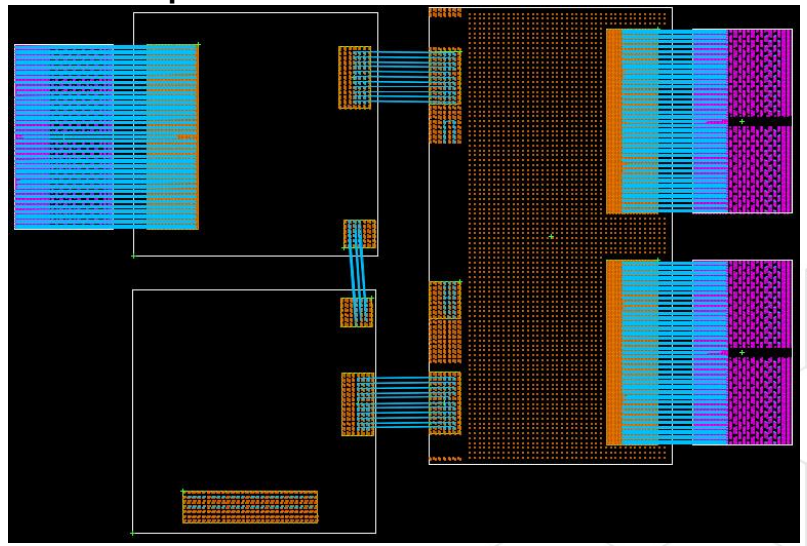
System-Level Floorplanning



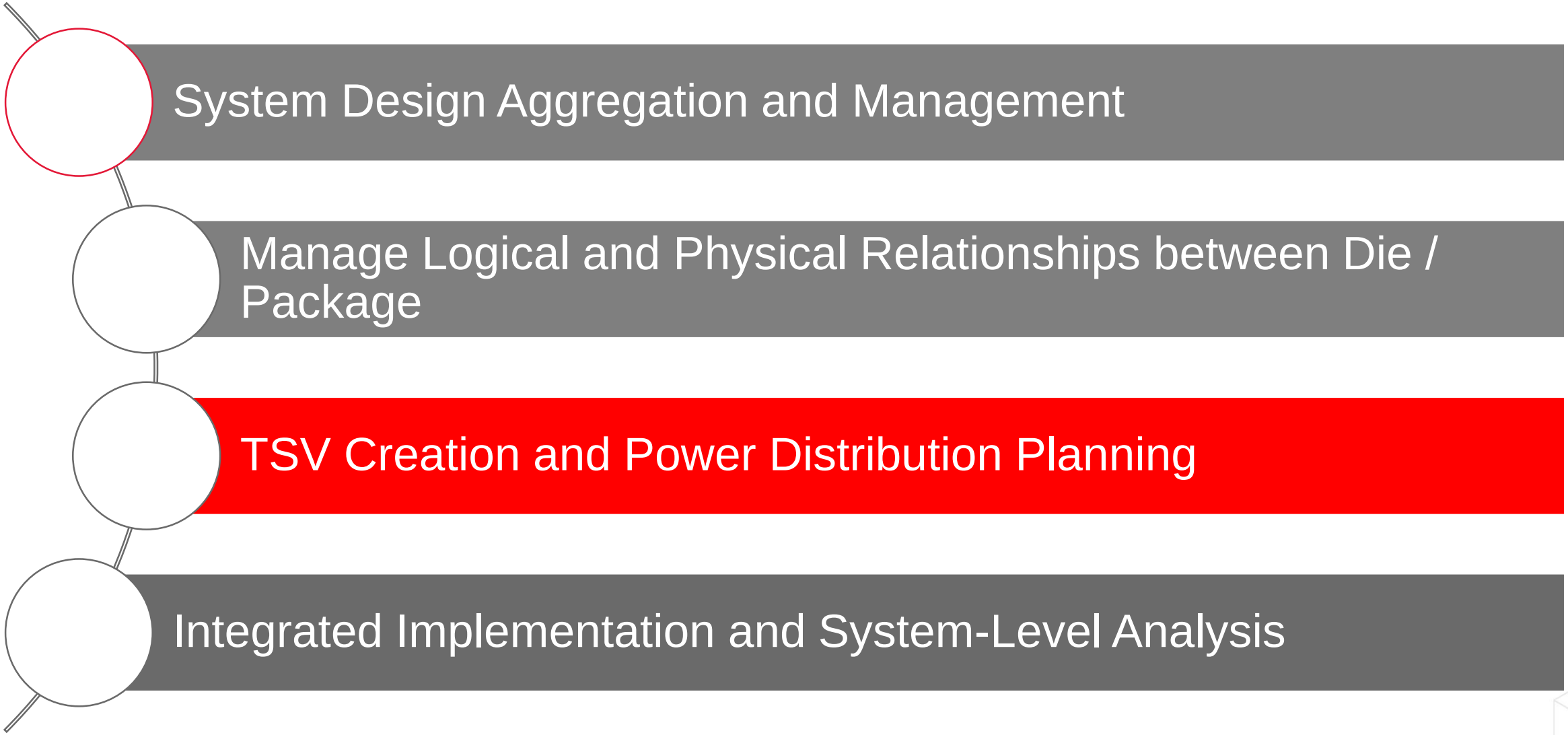
Robust Bump Creation



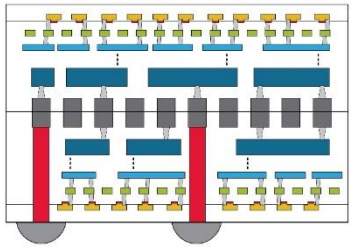
Macro Level Floorplanning
Optimize Die to Die Connections



Outline - 3D-IC System Design Planning and Management

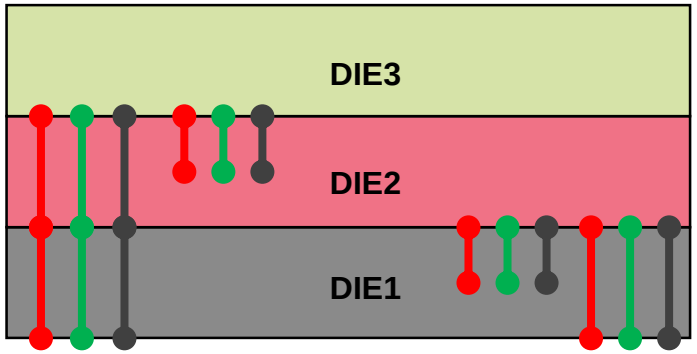


Plan and Create Through Silicon Vias (TSV)



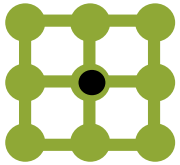
- Proper TSV planning is required upfront
  - Define TSV layers and tech file
  - Create TSVs for external and die to die connections
  - Pattern creation and connectivity optimization
  - Top-level and die netlists need to include TSVs
  - Obstructions needed on die layers for P&R

TSV Layer Planning

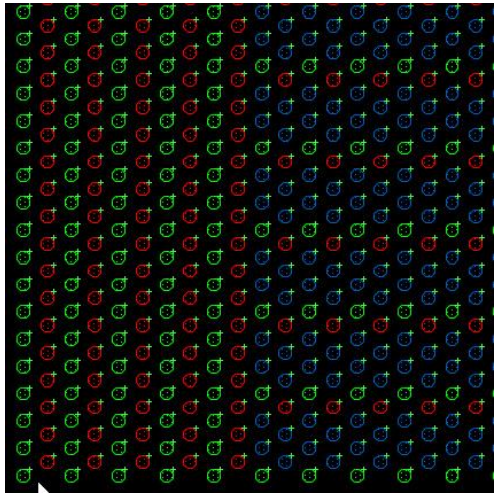


- Proper Power and Ground Delivery
  - Current capacity requires multiple TSVs
  - Generate PG TSV patterns
  - Utilize multi TSV LEF macros

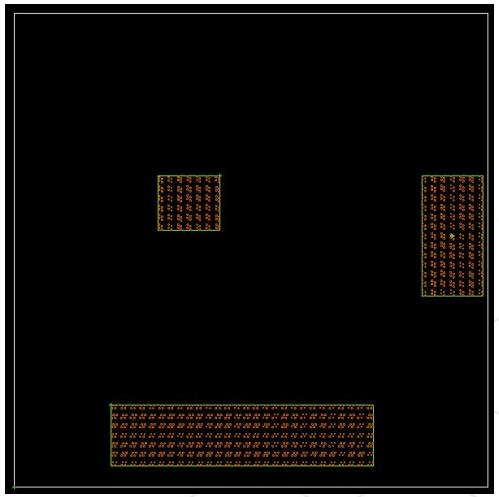
Multi TSV LEF Macros



Robust TSV Patterns



TSV Location and Layer Obstruction Planning

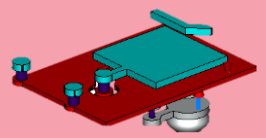


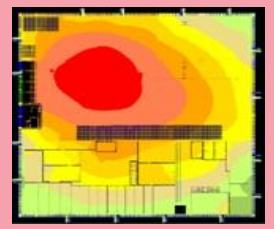
Outline - 3D-IC System Design Planning and Management

Implementation, System-Level Analysis and Sign-Off

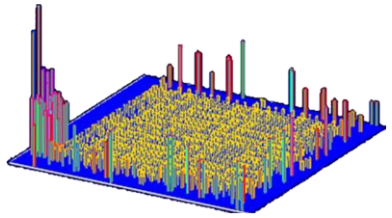
- Integrated Implementation Tools
  - Digital P&R, Custom Layout, Package
- Configure system-level analysis
  - Define tool specific design configuration
- Launch integrated analysis tools
  - Monitor tools and present results
  - All tools work off common design data
- Ability to run early analysis
  - Model design prior to completion

On/Off-Chip 3D EM Extraction

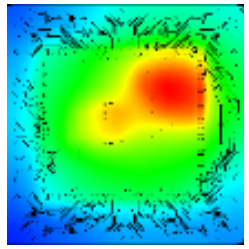




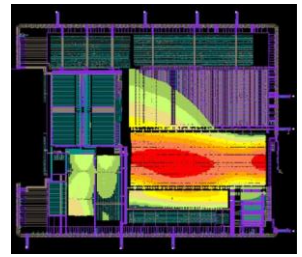
System Level Analysis



Signal / Power Integrity

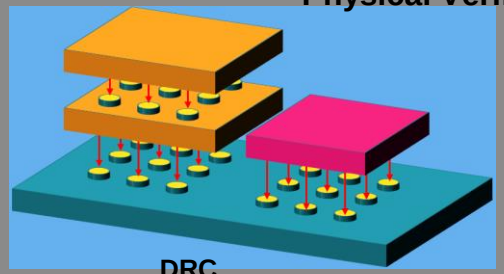


Thermal

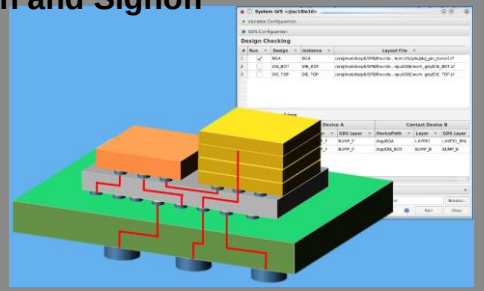


IR Drop

Physical Verification and Signoff



DRC



System LVS Checking

Conclusions – Designing the Next Generation of 3D-IC Designs

- An Open Platform Supporting 3DIC Designs at the System-Level is critical
- Flexible System Assembly, Co-Design and Management is Key to Building Optimized System-Level Designs
- Rule-Driven Advanced Instance Array Generation is Required to Manage and Optimize System-Level Interfaces Including TSVs
- Tight Integration with Implementation, Electrical, and Thermal Analysis Tools is Needed to Provide the Complete Co-Design and Co-Analysis Solution

Thank you for for your attention!

brianj@cadence.com

cadence®

Overview

Today's 3D-IC system designs face enormous challenges to assemble and manage an aggregated top-level system design consisting of multiple die from various process technologies and design tools. This capability is essential to model, analyze, and validate the entire system design both logically and physically.

A system-level planning and co-design environment is needed to effectively create and manage the assembled top-level design, while also managing design data for each individual die and the package design.

The logical and physical relationships between the die needs to be defined and managed throughout the system design project cycle.

Robust net propagation and name mapping capabilities are essential to create and manage the top-level netlist with each device level netlist.

Automated contact pad generation and management is needed to ensure proper pad alignment and logic connections.

These relationships are best managed using abstract data for the external interface nets and contact pads.

Solution requirements for these issues are presented in this session.