DESIGNING AND INTEGRATING MCM/SIP PACKAGES INTO SYSTEMS PCBS

KEVIN RINEBOLD AND KEITH FELTON, MENTOR, A SIEMENS BUSINESS

WHITEPAPER

Mentor®
A Siemens Business

DESIGN TO SILICON

www.mentor.com

HETEROGENEOUS SYSTEM-IN-PACKAGE INTEGRATION

Traditionally, MCMs (Multi-Chip-Modules) were a way of integrating several ASICs, or ASICs and memory, into a lower-cost, smaller form-factor, robust module that was an alternative to a single large SoC. Instead of integrating all or most of the system needs onto a large and complex single SOC, you could design and fabricate smaller, high-yielding ASICs (Figure 1) and make them behave like a single large SOC. This proved attractive in the defense and aerospace markets where volumes are lower than in the consumer commercial market and where trying to create a mixed technology SoC would be either impossible or cost prohibitive.

Figure 1: Intel® Clarkdale module circa 2009

As the market for mobile wireless-enabled products exploded, along came a new demand for signal processing, flash memory, and wireless communications in a system that could fit inside a product smaller than a human hand and be cost effective. This challenge drove a different approach to packaging, and led the industry to 3D chip stacking of bare die.

This approach, referred to as systems-in-package (SiP), required interconnections from die-to-die so that the package became a critical technology element. Some view SiP as a vertical MCM, in contrast to horizontal MCMs made popular in high-performance computing applications (Figure 2). Benefits to SiP include user IP integration, IP reuse, mixed analog/digital design, low design risk, integration of large memories, reduced process complexity, low developmental cost, and shorter time to market.

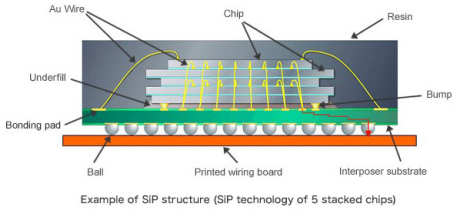

Figure 2: Example of SiP structure (SiP technology of five stacked chips)

Perfect for heterogeneous applications, this building-block approach can be attractive to anyone looking at integrating ASICs into a high-performance sub-system with performance that is close to that of a custom SoC for a fraction of the cost and time to market. In short, SiP brings together ICs, including SoCs and discrete components, using lateral or vertical integration technologies (Figure 3). In this paper we will focus on the design of a SiP that is part of a known PCB system, which is a common occurrence in systems companies. In such a scenario, the SiP not only needs to meet the integration and performance goals as an integrated set of heterogeneous ICs but also must be optimized for its target system's PCB, or even PCBs.

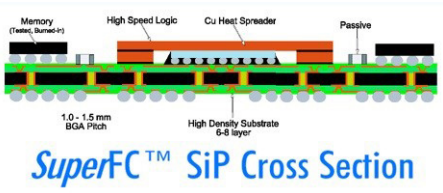

Figure 3: Example of a typical heterogeneous SiP package

UNDERSTANDING THE TOTAL SYSTEM PICTURE

Visualizing and comprehending the entire multi-substrate design (multi-die SiP and system PCB) while keeping essential and critical design requirements in mind can easily overwhelm the designer and design team. Typically, the task of creating a new, multi-die heterogeneous SiP and its corresponding system board(s) involves two to three different engineering teams with three different perspectives. Although it can be possible to get the different design disciplines together in a coordination meeting, a successful joint plan really requires a methodology, process, and usually some technology and automation for tying these worlds together.

SiP designers, often a multidisciplinary team, have to cooperate to solve several optimization problems, including system I/O requirements, thermal and signal integrity constraints, die placement and orientation, stacking configurations, package substrate and interposer design, interconnect design at the IC and package level, while also taking into account the constraints on the system's PCB.

The first requirement for optimum results is the availability of an SiP connectivity environment. It must be able to read and write a top-level hierarchical netlist of all the heterogeneous pieces (ICs -digital and analog-, package, board) and merge them all, even if they're coming from different design environments.

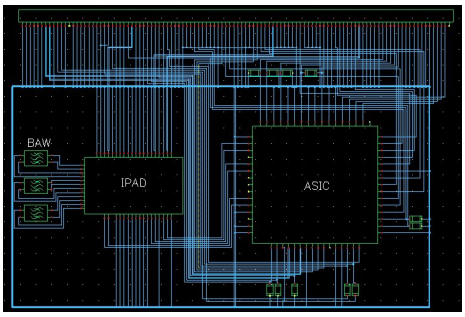

Figure 4. Schematic view of a simple heterogeneous SiP design

It must also be able to compose schematics and generate the complete models (Figure 4).

Then, the complete floor-planning of all the heterogeneous SiP elements must be realized.

A new class of algorithms, with the ability to work in a 3D world, must be available to properly stack the required components, place them along with the rest of the SiP components, and properly plan the interface between all the elements including package bond pads.

A smart I/O planning strategy must be applied to all the ASICs involved in order to concurrently validate and/or optimize initial I/O periphery configuration, respecting different sets of rules in order to satisfy silicon process and electrical and assembly rules.

MULTI-SUBSTRATE INTEGRATION

The integration of multiple levels of substrates common to most complex SiP devices, including the SiP into its host system PCB, is a challenging task, especially if you try to do it during detailed implementation with layout-focused design tools. A more flexible, productive approach is to start with prototyping, planning, and optimization. Here the focus is on modelling the floor plan and I/O requirements of the IC, integrating the package substrate constraints and variables, and taking into account the multiple PCB platforms (form factors) that the SiP is intended to work with.

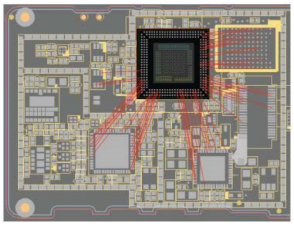

Figure 5: In this example, the SiP package must have its signal/ball assignments aligned with three major PCB-level devices in order to simplify routing and avoid possible signal integrity or quality problems.

Industry standards should be leveraged by individual domain: e.g. LEF/DEF for the IC data and csv-formatted ASCII or AIF for the package substrate data. This provides the engineers with the ability to visualize the complete "system." After the system has been captured, you need to compensate for cross-domain signal naming conflicts and power/ground shorting. The flow should allow designers in each domain to use the design environment with which they are most comfortable, such as Verilog® for the chip designers and graphical schematics for the board designers.

Most often, the SiP substrate pinout (ball and signal locations/assignments) has the most flexibility with respect to optimizing interconnect paths from the multiple IC to the board.

Prerequisites to the prototyping integration process are the availability of all the different device models involved. At a minimum, you need the physical die footprint model, which is often available in csv or AIF format from the IC design team(s). You also need the netlist for the devices which, again, can be csv or Verilog. If this is not available, then you need to generate one manually or construct one during the prototyping phase.

Once the prototype is constructed, it provides a vehicle for performing multiple evaluations before going forward to detailed layout. Such evaluations could include LVS verification of the complete SiP level system netlist, system-level static timing analysis (STA), and interconnect optimization between SiP objects. The goal is to evaluate and optimize a prototype so that it is predictably achievable without (or with minimal need for) performing ECO's to signal to pin/ball/bump assignments during physical design.

DETAILED SIP IMPLEMENTATION

The physical design of the SiP substrate also presents many challenges to the designer and design process and is highly likely to require the support of advanced technologies, such as:

- High density interconnect (HDI)
- Microvias
- Embedded passives (EP)
- RF shapes
- 3D wire bonding
- Bond pattern generation
- Substrate routing
- Complex power plane shapes
- Adherence to extensive fabrication and manufacturing requirements and rules

3D wire-bonding poses a number of challenges for IC packaging design, especially if you stack several dies and if pin count I/Os range between several hundred and the low 1000's.

To successfully synthesize die patterns, design tools must support automatic bond pattern generation of entire die stacks while respecting all the 3D aspects of IC packaging and obeying numerous 3D related rules and constraints. Once a bond pattern has been established, the challenge of connecting the bond pads to the package becomes obvious. Often, the only means for obtaining a routable solution is to let the router assign package pins while routing which may require updates to the IC's I/O planning.

Looking at the domain data from the PCB, SiP, and bumps without unnecessary clutter helps problems pop out more clearly.

Combining data from the three design domains simplifies the process of solving the real problems of designing across domains.

Poor routing, for example, is usually not caused by implementing rats' nests, but by the crossed net-lines to begin with. Untangling (connectivity optimization) takes into account the net's destination on the target PCB, its path through the package, and its connection on the die. Each segment of the system path is comprised of thousands of signals, along which major rats' nests form. These crosses can only be unraveled successfully with visibility across the whole path.

Figure 6 shows the optimal way to address the issue. Identify the application signal groups, add the IC and package, then optimize the connection paths. At first glance, this looks like just a PCB but here you actually see the package internals and the die.

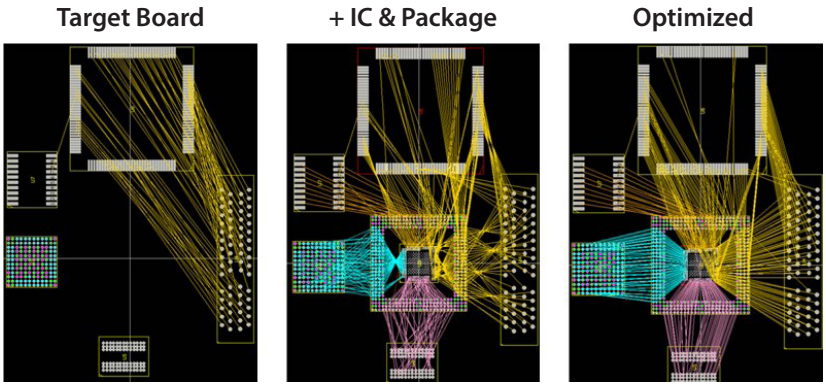

Figure 6 – System path uncrossing and route optimization before layout

CO-DESIGN AS PART OF A DESIGN METHODOLOGY

When designing a new die such as a processor, the system target is an industry-standard memory chip with a fixed pin-out. Unraveling starts from the system PCB and drives toward the die, as the PCB design domain drives the optimization.

If you co-design the PCB, package, and die, pre-optimization enables ideal, easy-to-implement routing. Co-design provides design flexibility in all three domains, as the design of the target PCB, package, and die pads all begin at the same time.

When starting a new application with a known, already-completed die, you still have design flexibility in the BGA ball-out and board-level connections. In this case, unraveling must start from the die and work out toward the system PCB (Figure 7).

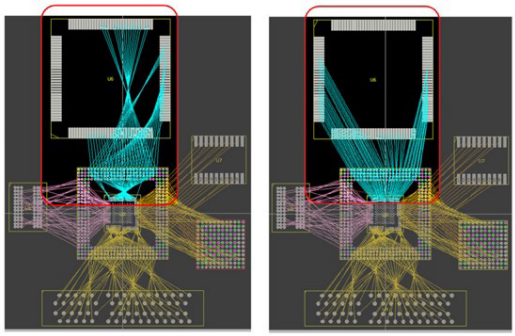

Figure 7– Chip-driven connectivity uncrossing

Looking at these two situations of uncrossing, it is obvious that design software must integrate all three types of design data to provide a system-design environment. Uncrossing must be flexible enough to switch modes, optimizing from the PCB toward the die, or from the die toward the PCB. The blue signal group in Figure 7 shows what an uncrossed system design looks like. Routing becomes easier, faster, and without interconnections. It also has fewer vias, which leads to better signal and power integrity. Fewer vias and crossings mean fewer layers which, in turn, drives cheaper and
smaller designs.

SIP CO-VERIFICATION REQUIREMENTS

In addition to physical implementation, system verification is a critical part of SiP design. Accurately simulating ultra-fast signaling technologies requires system-level analysis, involving a range of physical structures and modeling technologies. Simulation of a complete signal path (or "channel") requires accurate modeling of I/O buffers, all chip-to-chip interconnect (including IC packages, PCB traces, connectors, and vias with associated bypass capacitors), and associated power-distribution structures (planes in the package and PCB, decoupling capacitors, and stitching vias). See Figure 8.

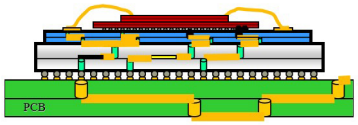

Figure 8. Example of a complete SiP model for system verification

Power integrity concerns itself with proper delivery of power to IC buffers and proper quality of the launched signal as it interacts with plane layers in the PCB or package. To fully understand how to distribute power to an IC requires simulation in the context of the package; and possibly even in the context of the PCB.

"Classic" signal-integrity concerns (like delay) are also becoming an issue in package design, as timing margins shrink and packages become larger. For SERDES signals, very high frequencies that require careful simulation are involved. In addition to simulating, package designers typically follow very tight physical rules, as they attempt to maintain differential impedances, eliminate reflections and discontinuities, etc.

Some structures in typical packages can be extracted/modeled using well-proven, mature techniques from the signal-integrity world (e.g., trace routes, vias, and sometimes, bond wires). Others require more-tedious 3D extraction (less controlled bond wires and possibly balls). Accurate wideband models of such structures are usually captured in the form of S parameters. Although many package designers are tempted to rely heavily on 3D simulation, such electromagnetic tools tend to be difficult to use and very slow. Large-scale 3D extraction involving thousands (or even just hundreds) of wire bonds and other interconnect elements is completely impractical. Thus, whenever possible, the problem should be broken into sub-elements, each one of which is modeled with the best available engine; one that provides sufficient accuracy but also reasonable speed and ease-of-use.

SPEND TIME ON DESIGN, NOT FIGHTING DESIGN TOOLS

Practically speaking, trying to work across design domains can create issues. When combining designs from die, package, and PCB, the three design types should, ideally, share data directly. Package substrates, multi-chip modules, and simple BGAs are all physical entities of electrical systems, so design tasks are similar.

The logical thing is to make package design an extension of physical layout so that all design groups use a single toolset across the system-design environment. The benefit of a single design flow is that there are fewer tools to learn, so designers can be layout masters rather than tool masters.

Often, companies will use PCB layout software to design a package substrate if design features are similar in shape, size, and use, as very small – HD – PCB features, like vias, traces, and the die, are mounted as flip chip. On the other hand, if you are designing a multi-die package, with stacked die, wire bond attach, and are employing an interposer (silicon or organic) with High Density Advanced Packaging type features, such as a 200-via array as a through-via, then focused IC package design software is the better choice.

ALWAYS USE CONSTRAINTS

When designing a package, rules-based I/O assignment and optimization is the equivalent of constraint-based routing. Package design constraints would include signal grouping, locked signals, and a critical signal's proximity to ground. Having a good ball-out implies that the ball is optimal for the die, the package-substrate routing, and PCB routing. It also implies that the ball-out meets all the rules required by die, package, and PCB designers, and that it will always meet those rules, even after possible ECOs.

The best way to remember a rule is to convert it from something that's just in your head, to a constraint within the design. That way, when someone picks up your design in the future, your intent and reasoning are still embedded in the design. Constraints are essential for a correct-by-design methodology.

Breakout routing can affect optimization so dramatically that overall routing must be taken into account. Breakout routes rearrange connections, changing the nature of the rats' nests and providing yet another example of why the entire system must be taken into account at every stage of the design.

THE PROBLEM WITH ROUTING

In real-life applications (Figure 9), the effects of routing are too complex to imagine beforehand. Specific standards or interfaces assume a specific routing plan which adds a layer of complexity that must be combined with break-out routing.

As the interfaces are sketched in the design, you can see how the system could be implemented and how the design could be set up for best results. Once you get to this stage, it becomes clear what uncrossing needs to happen. A few hours of manually changing individual signal assignments for each circuit is usually enough but, if you have a rules-driven unravel engine, a few seconds is a better option than a few hours!

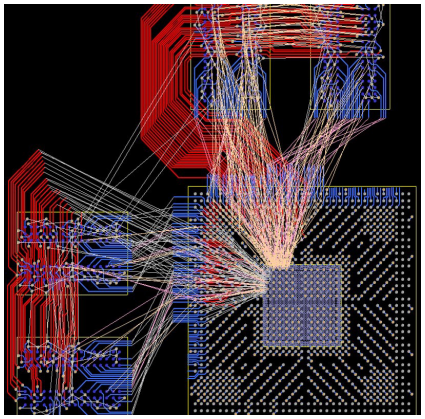

Figure 9: By using the host PCB, the package designer can assign and optimize signal/ball and signal/bump assignments providing an overall system-level optimization

While you can understand and imagine each piece of this complex system one at a time, it is too much for one person to envision and optimize them all at the same time. Automation and abstraction are necessary to productively optimize, and then implement, this level of complexity.

COMMUNICATION IS KEY TO SUCCESS

Instead of passing around Excel® spreadsheets crammed with signals and pins, good design software will export a ball map (Figure 10) without you manually having to type in names, rely on version control, or check that signal lists match. Documentation exported from the system ensures that document revisions are correctly matched to the design which was revised: BOM, ball-out, connection netlist, placement, etc.

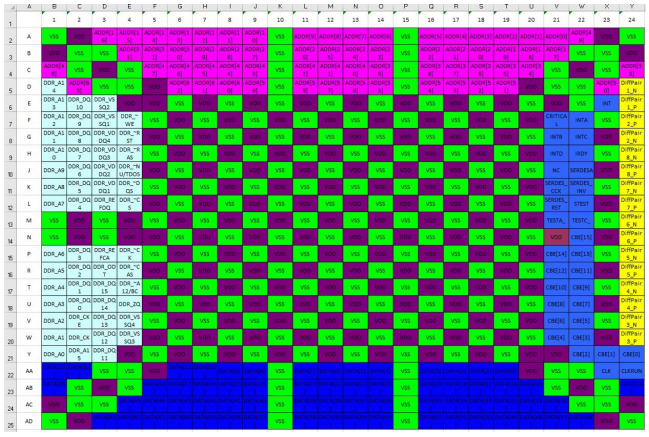

Figure 10: Exporting a ball map is quick and easy when using design software

DRIVING FABRICATION AND MANUFACTURING

Once the design is complete and free of connectivity, high speed, and power delivery errors, it is time to validate the design for fabrication acceptance.

Electronics design companies continuously strive to accelerate new product introductions (NPI) at the lowest total cost while still attaining the highest level of quality. In most cases, their SiP substrates are produced by a leading laminate fabricator, such as Kinsus, Kyocera, Fujitsu, SEMCO, Ibiden, or many others.

It's a proven fact that best-in-class organizations are 68% more likely than their peers to use Design for Manufacturing (DFM) validation, to help eliminate manufacturing defects, reduce revision spins and improve design to market time. And this becomes more important as SiP complexity increases. The goal is simple: reduce design revision spins due to manufacturing feedback, get immediate feedback on potential manufacturing issues during design, make the time-to-market path more predictable, and manage manufacturing costs based on supplier work and activity. The best way to address this is to include DFx during the design cycle, using focused DFx checking technologies that can be accessed during design/layout and that can apply the fabrication rules that will be used by the manufacturer/fabricator.

IN SUMMARY, PROTOTYPE FIRST AND DESIGN WITH A SYSTEM PERSPECTIVE

Completing a package layout with these techniques and steps ensures the best end-design possible. But before you build and test in the lab, try simulation for a quick and less-expensive way of finding unexpected signal integrity issues and ensuring that your design will operate as planned.

Xpedition® Substrate Integrator and Package Designer, in conjunction with Xpedition PCB Designer, HyperLynx® SI/PI and Valor® NPI/Calibre®, provide an integrated co-design methodology, from prototyping to manufacturing-ready tooling (Gerber or GDS) in a productive and predictable design flow.

For the latest product information, call us or visit: www.mentor.com

Corporate Headquarters
Mentor Graphics Corporation
8005 SW Boeckman Road
Wilsonville, OR 97070-7777
Phone: 503.685.7000
Fax: 503.685.1204

Sales and Product Information
Phone: 800.547.3000
sales_info@mentor.com

Silicon Valley
Mentor Graphics Corporation
46871 Bayside Parkway
Fremont, CA 94538 USA
Phone: 510.354.7400
Fax: 510.354.7467

North American Support Center
Phone: 800.547.4303

Europe
Mentor Graphics
Deutschland GmbH
Arnulfstrasse 201
80634 Munich
Germany
Phone: +49.89.57096.0
Fax: +49.89.57096.400

Pacific Rim
Mentor Graphics (Taiwan)
11F, No. 120, Section 2,
Gongdao 5th Road
HsinChu City 300,
Taiwan, ROC
Phone: 886.3.513.1000
Fax: 886.3.573.4734

Japan
Mentor Graphics Japan Co., Ltd.
Gotenyama Trust Tower
7-35, Kita-Shinagawa 4-chome
Shinagawa-Ku, Tokyo 140-0001
Japan
Phone: +81.3.5488.3033
Fax: +81.3.5488.3004

Mentor® A Siemens Business

MGC 05-17 TECH15450-w