USING CO-DESIGN TO ENSURE MULTI-FABRIC SYSTEM-DESIGN SUCCESS

MENTOR GRAPHICS

WHITEPAPER

IC PACKAGE DESIGN

www.mentor.com

ABSTRACT

Today's SoCs, multi-core CPUs and GPUs with their high performance, high bandwidth interconnect interfaces put demanding challenges across the entire system signal path, requiring system-wide optimization for product success. However, although typical system substrate design discipline (chip, packaging, and PCB) have well-developed design approaches, the interaction between them largely remains isolated and undeveloped.

Often the bridge between chip and system PCB, the IC Packaging offers the best opportunity for optimizing signal performance, power delivery/integrity and interconnect optimization. Using a multi-substrate design approach, simulation becomes an easily reachable reality, from the chip's buffer block through package routing and on through system PCB implementation. By optimizing critical and key signal down interface paths throughout the system will result in far fewer downstream issues that will require resolution.

The traditional path to system design is to know what you need to build, to identify what must be done, and then to design to those requirements. The defense and aerospace industries have had great success enforcing this method of constrained design, especially for complex system designs where the correct-by-construction approach speeds design time, improves quality, and lowers costs by reducing design iterations.

UNDERSTANDING THE BIG PICTURE

The correct-by-construction approach forces the designer to make assumptions regarding the limiting issues in the design, and then to simulate these assumptions in software before building a physical prototype. Before translating a conceptual design limitation into a physical design constraint, designers must understand the source of the challenge and where to go in the design to resolve it. This requires designers and engineers to see the big picture in order to make good decisions.

Today's IC Packaging challenges are varied, ranging from octo-core CPUs with Gigabit interfaces to multi-die stacking, Fan-Out Wafer-Level-Packaging (FOWLP), 2.5D silicon interposer based integration, and more. Yet, with each micro-electronics innovation, system architects, electrical engineers and PCB designers are asked to develop designs that are often twice or more as complex and fit them into a smaller space/form factor in less overall time.

Visualizing and comprehending the entire multi-substrate design while keeping essential and critical design requirements in mind often overwhelms the designer and design team. Typically the task of creating a new SoC/chip, its corresponding package, and a new system board(s) involves 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.

There are three concepts that, when added to traditionally deployed design methods and can enable the creation of an optimal system product: multi-substrate visualization, system-level cross substrate interconnect untangling, and cross-substrate co-design. Multi-substrate visualization means not solving micro, or single substrate design issues until you can see the whole picture. Once you step back and look at the design as a whole, the pieces of the solution start to become clear.

To look at and manage the three different substrate design domains, engineers need to be aware of all the various domain design data: physical layout, schematics, Verilog® netlists, ball map spreadsheets, and die design files. An early challenge companies run into is signal naming, each design group usually has

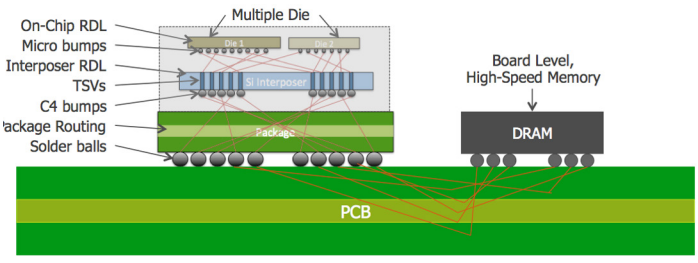

Figure 1 – System constraints and optimization prior to implementation.

its own set of signal names and yet needs to see the entire connection path, including where net names might combine. To make correct design decisions, designers must have all the required information upfront, especially in package centric design where every piece of the system, die-to-package-to-PCB, must be seen (Figure 1).

SEE THROUGH THE NOISE TO THE CHALLENGE

REMOVE THE CLUTTER

If Figure 1 looks too simplistic, take a look at a real designs raw system view in Figure 2. It's a typical design without any abstraction. Study it long enough and you can begin to understand what is going on, yet the overall intent is difficult to grasp.

When looking at a raw system view, it's hard not to be intimidated. The details are endless (Figure 2).

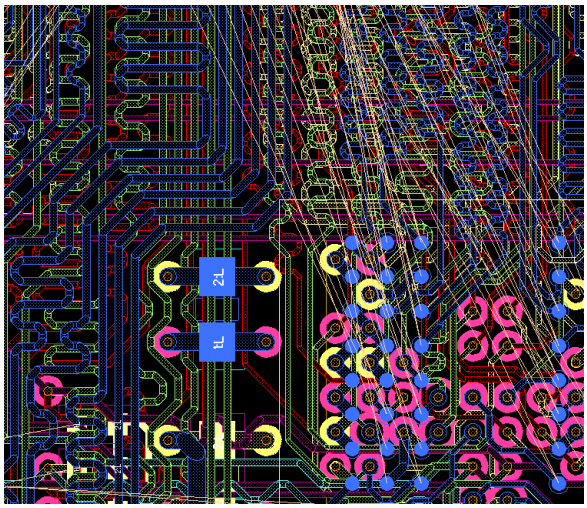

Figure 2 – Raw system view of a multi-substrate design.

The level of detail shown in Figure 2 provides too much information to be useful. Even the addition of coloring and shading provides little help. To understand the key challenge areas you need to isolate the relevant and pertinent information and remove the clutter. Without the right tool for viewing and designing HDAP packages it's extremely difficult to visualize the situation clearly and accurately, much less to identify problems and find solutions.

Start with the assumption that you want to see everything relevant, but only those things that are relevant to a signal going Die-to-Package-to-PCB-part endpoint. With that information you'll be able to see the whole system view without getting lost in too much information. Removing everything that's not applicable to this context presents the view below (Figure 3).

This view shows everything you need to design a signal path through the whole system, and nothing that you don't, thus setting up an opportunity for design optimization.

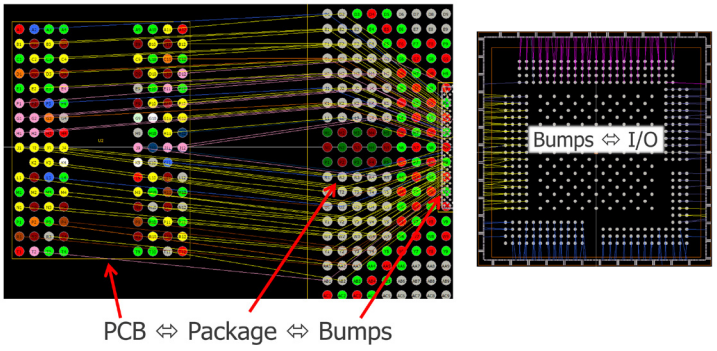

Figure 3 – Multi-substrate visualization leads to system optimization.

Looking at the domain data from the PCB, package, and bumps without unnecessary clutter helps problems pop out more clearly. This type of system-level view is like asking a good question, one that is half-answered by clarifying the problem. Combining the data from the three design domains simplifies the process of designing across domains to solve the real problems.

Poor routing, for example, is not caused by implementing rat's nests but by the crossed net-lines to begin with. Untangling, sometimes known as path finding, 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 rat's nests form. These crosses can only be unraveled successfully with visibility into the whole path.

Figure 4 shows the right 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.

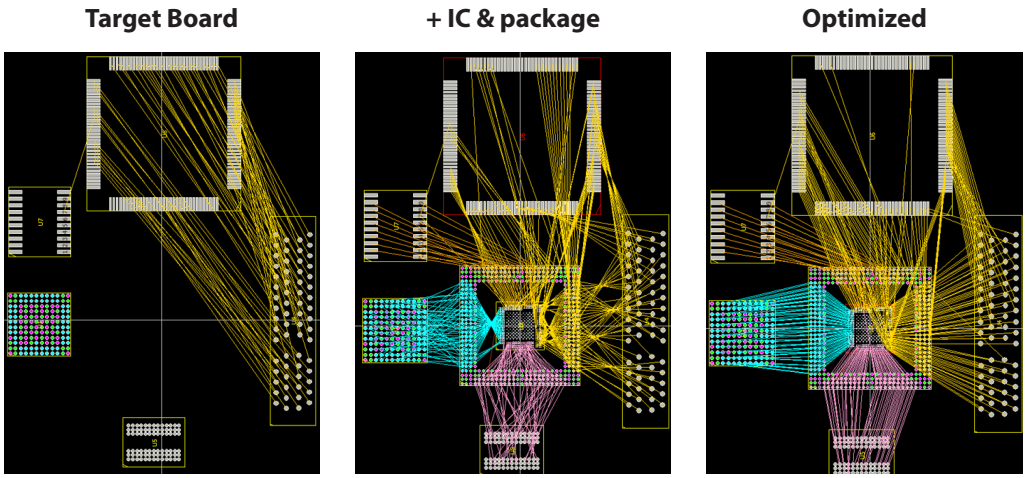

Figure 4 – System path uncrossing and route optimization before layout.

CO-DESIGN AS A 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 5).

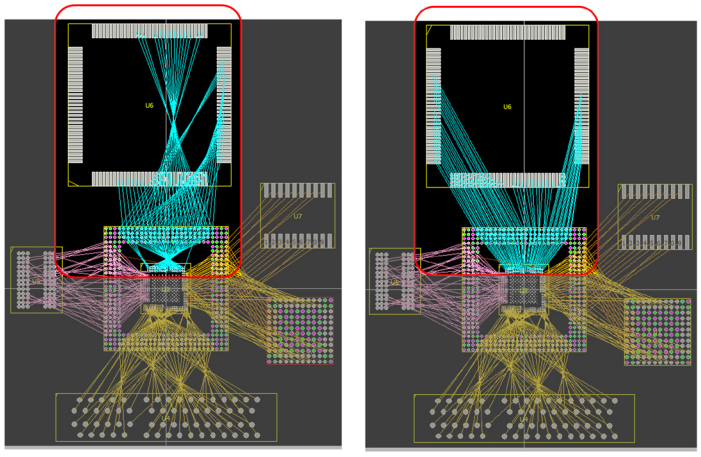

Figure 5 – Chip-driven connectivity uncrossing.

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 5 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 means fewer layers, which in turn drives cheaper and smaller designs.

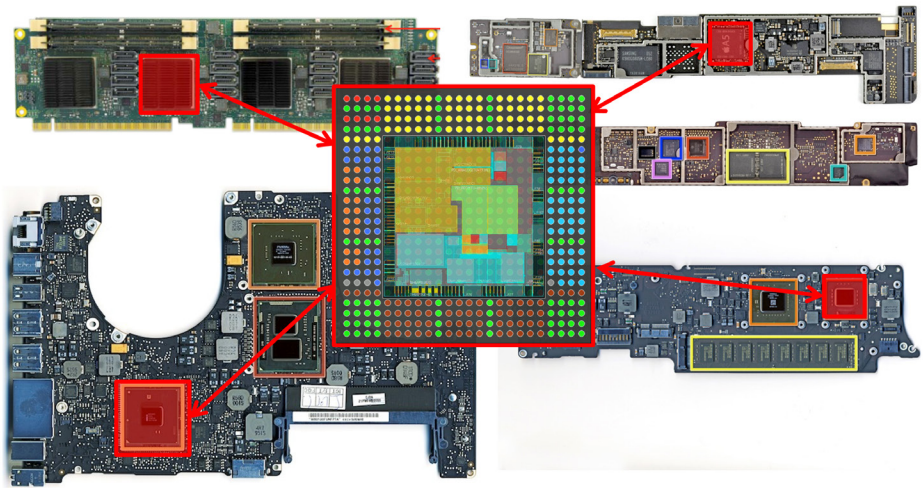

Figure 6 – Connectivity uncrossing and optimizing a common package for variant designs.

The ideal package solution also has to stay ideal for multiple applications. Not only must it allow you to step back and see the 'big picture,' it must also allow you to see multiple big pictures as in the difficult design situation where a single chip must be optimized for several application boards (Figure 6).

SPEND TIME ON LAYOUT DESIGN, NOT ON TOOLS

Practically speaking, trying to work across design domains can create tool 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.

It makes a great deal of sense to use PCB layout software to design a package substrate if your design features are similar in shape, size, and use as very small – HD – PCB features like vias and traces. On the other hand, if you are designing a silicon interposer with die-like features, such as a 200-via array as a through-via, then silicon design software may be 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 rat's 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 7), 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.

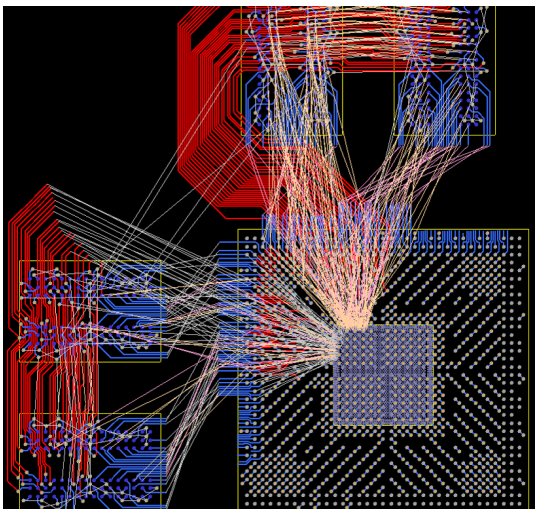

Figure 7 – The complexities of interface and module routing.

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!

If you were thinking maybe you could handle a little more complexity without system-level uncrossing, what happens when a design like Figure 8 shows up? The challenge is to design a board that is very complex, very high-speed, and therefore highly constrained. The complexity of a huge FPGA gets multiplied when you use a whole array of them.

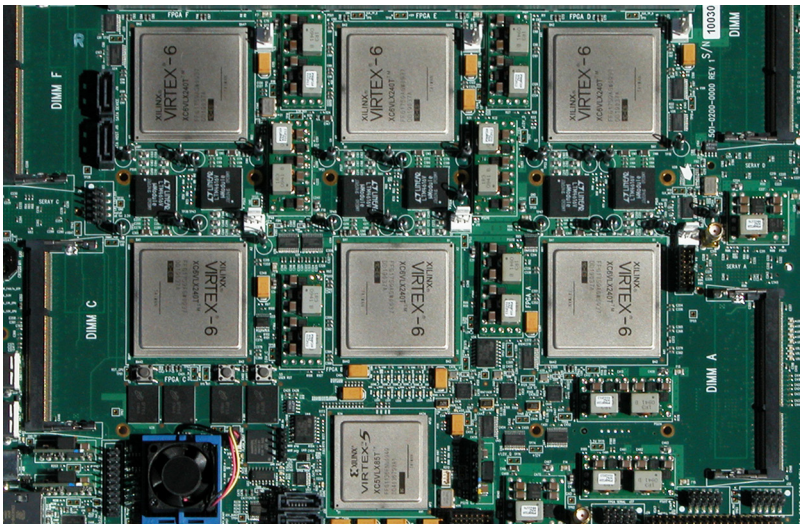

Figure 8 – Even a relatively small ASIC emulation board can be extremely complex to envision and optimize manually.

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 9) 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 revisions match design revisions: BOM, ball-out, connection netlist, placement, etc.

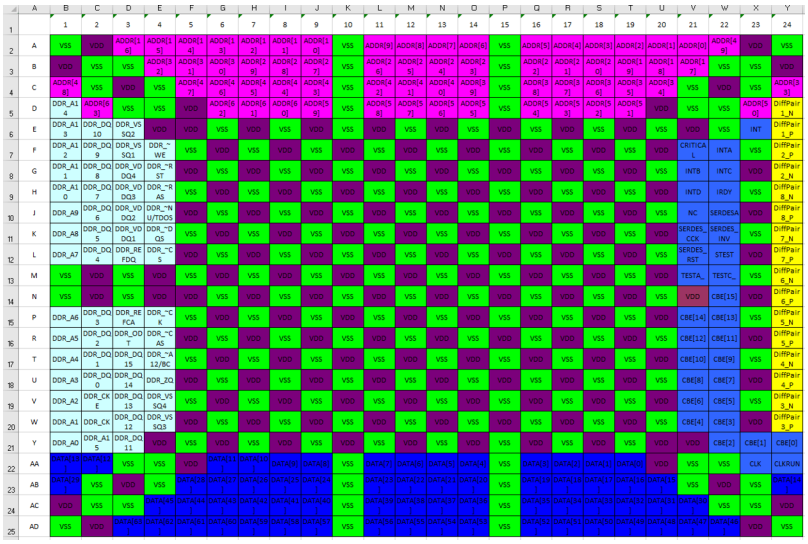

Figure 9 – Exporting a ball map is quick and easy when using design software.

THE NEXT GENERATION OF HIGH-DENSITY-ADVANCED-PACKAGING

As next-generation High-Density-Advanced-Packaging (HDAP) designs become more common, PCB designers and engineers can use system-level co-design to tie multi-substrate visualization, planning, and optimization, into a complete multi-board design.

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

The Xpedition® IC Package Design flow provides a co-design methodology that allows design teams to easily plan and optimize I/O and connectivity from a chip through multiple packaging scenarios and on to printed circuit boards.

For more information, go to https://www.mentor.com/pcb/ic-packaging/overview.

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

MGC 6-17 TECH15640-w