

Mastering the I/O planning puzzle

Kevin Rinebold

Page 1 of 2

[EE Times](#)

(09/24/2007 9:00 AM EDT)

Systemwide [I/O](#) planning is an exercise in coordinating device placement with associated pin and net assignments across the chip-package-board system to maximize system quality for the target application. Achieving this goal is a multidomain balancing act of trade-offs and iterations.

What might be the ideal pad ring layout from a [chip](#) perspective may be less than ideal from a package routing or manufacturing perspective. The package ball pad assignment that minimizes layer count at the board level might prove inadequate from the standpoint of chip-level power distribution.

The key to systemwide I/O planning is achieving balance across all domains of the system so key design criteria can be weighed and evaluated in full system context to guide design decisions.

Attempting any type of coordinated design planning across the chip, package and board using traditional tools and [serial](#) methodologies can be frustrating at best. One problem is separate design environments and databases—one for the chip, a second one for the package, and a third for the board. Even in this situation, it's not uncommon for design teams to collaborate using spreadsheets to communicate pin assignments. The shortcoming is that it is based on snapshots of static data, resulting in a highly iterative, error-prone process that does little to reduce cycle time or cost of results.

Device integration at the package level in the form of System-in-Package (SiP), stacked-die and Package-on-Package (PoP) further challenge traditional tools and methodologies. The multichip aspect of these packages adds the dynamic of chip-to-chip connectivity in addition to chip-to-package. Designers often use the fixed I/O of one chip to influence I/O and connectivity assignments on adjacent chips.

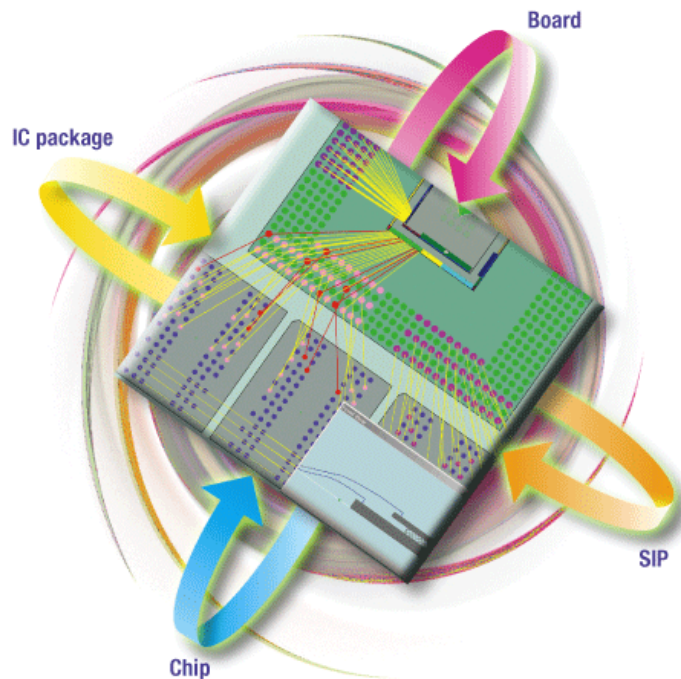
While chip floor planning and implementation tools work well for their intended application, they lack the ability to deal with multiple chips simultaneously. On the other hand, package- and board-level tools that support multiple chips lack the needed gate and [macro](#) visibility necessary for chip-level I/O placement and assignment.

Several other factors and considerations figure into the systemwide I/O planning puzzle. Some are well known, such as system compatibility, completeness and availability of [IP](#) and package lead times; but there are other, more oblique, factors. Designing for application-specific packaging, detecting and resolving differences between logical and physical data, and correlating data across domains are just a few examples.

Serial approaches

The historic approach to chip-package-board design has been a serial top-down flow where the chip drives package connectivity and, in turn, the package drives board connectivity. Increasingly, there are situations requiring compatibility with existing systems where the board becomes the influencing factor driving upstream connectivity through the package and back into the chip. This bottom-up approach necessitates some element of package design to derive the starting point for pad ring layout on the chip. This is especially true for flip-chip packaging, where a high degree of coordination is necessary to derive a bump pattern that is mutually conducive to route both chips and packages.

During feasibility studies or early-stage design planning, it's not uncommon for design decisions to be made in the absence of detailed information. An example is the availability and completeness of silicon intellectual property (IP). In many cases, IP isn't available during early stages because it is not yet designed or purchased. Therefore, a placeholder is inserted that approximates the size and pin count so feasibility or planning can proceed. Effective systemwide I/O planning requires the flexibility to instantiate missing data on-the-fly, work at different levels of abstraction, and use best available data.



Once companies embark on a process for systemwide I/O planning, they're quickly confronted with the issue of syntactical differences in net names between domains. It's not uncommon for a logical net to be referred to by three different substrate-specific names. For example, the logical net for Address [bit 0](#) may be called ADDR(0) on the chip, A(0) on the package and AD(0) on the board. From the system perspective, these are all the same net, but for I/O planning they must be correlated and mapped without changing their respective net lists.

Differences in how data is represented in logical and physical environments create another mapping headache. For example, physical data in the form of a spreadsheet references 32 instances of an I/O cell called Datapad as Datapad(0-31). The corresponding Verilog contains two blocks called Word(0) and Word(1) each referencing Datapad(0-15). In this case, the Verilog references Datapad(0) twice at different levels of the hierarchy, while Datapad(16) goes unreferenced. Mapping and correlating differences between logical and physical data is critical to systemwide I/O planning.

Technology and market trends validate the need for systemwide I/O planning tools and methodologies. Even though the benefits are easily quantified, companies struggle with how best to incorporate these tools and methodologies into existing design flows with minimal disruption. For many, the first step is internally developed tools or scripts based on a spreadsheet to communicate pin and net assignments between design groups. It's a step in the right direction but has minimal impact on cycle time or cost.

In the last 12–24 months, the EDA industry has responded with commercial solutions to address the growing challenge of coordinated I/O design planning. The first generation has been the linkage of traditional chip, package and board design tools by a common conduit, which facilitates the exchange of pin and net information between the domains, but in a serial fashion using static data.

Success or failure for many consumer applications is determined by their ability to meet product introduction targets. In some cases, the schedule dictates the package is sent to manufacturing before chip completion, due to complexity of the package substrate and assembly processes. This requires a high degree of confidence in the chip and package working together, achieved only through coordinated design planning.

Another twist for I/O planning is the use of application-specific packaging in which one chip is designed into multiple package configurations depending on end-market applications. The challenge is to derive an I/O pad ring layout that works equally well for the range of configurations. Accomplishing this is a highly iterative, cumbersome process due to the "one die, one package" orientation of traditional tools and methodologies.

A newer generation of I/O planning solutions takes a more revolutionary approach, bringing all data sources together into a common, unified planning environment. It facilitates placement and connectivity scenarios easily derived and evaluated in the context of the full system.

A unified chip-package-board [data model](#) enables changes automatically to propagate to adjacent domains, providing instantaneous feedback on their systemwide impact. This ability to optimize the I/O and connectivity design plan for performance, cost and manufacturability prior to detailed implementation is a process improvement that can positively impact business.

Systemwide I/O planning tools require innovative functionality to manage and manipulate a range of data at various stages of completeness. Ease of use and implementation are crucial because these tools must plug into existing flows with minimal disruption. Instantiating data on-the-fly is critical for timely planning and feasibility in the absence of detailed data. They must be vertically aware for stacked package applications and support [attachment](#) technologies such as wire bonding and flip chip.

[File](#) formats such as LEF/DEF, Verilog and VHDL are common chip-related data sources used during I/O planning. Original data sources and content must be tracked throughout the planning process, so when the time comes for back annotation, the appropriate information is communicated.

LEF/DEF readers must be robust to accommodate various approaches and data constructs with the ability to extract the I/O-related information. Spreadsheet support for I/O pad ring definition is still necessary as companies take incremental steps toward systemwide I/O planning. Bidirectional support of package data is needed for tools from vendors such as Sigrity, Cadence and Zuken, either in their native file formats or using industry-standard files such as AIF. The ability to import PCB data from popular [CAD](#) systems also is required.

Once the data exists within the I/O planning environment, functionality is needed to define and manage the relationships between devices, i.e., which chips go in which package and in what order. Automated hierarchy management can help expedite this process and must be an underlying [function](#) in any I/O planning tool. Another critical requirement is the ability to correlate and map differences between domains, as in the case of net name syntax or mismatched Verilog data.

A combination of automated and interactive features is needed for device placement and optimization. Ideally, these features are transparent across domains and easily applied to various situations. Results must adhere to substrate-specific technology rules or region-based personalities, as in the case of multiple [voltage](#) planes. Device placement and pin assignments must consider requirements for high-speed interfaces that utilize differential signals or special net topologies. For I/O cell placement, a simple sequencing mechanism is needed to ensure the appropriate ratio of signal to power/ground cells and other devices that occur in a regular pattern such as ESD control or filler cells.

Evaluating manufacturability during early planning can have a significant impact because meaningful change can still take place then. For example, engineers have the ability to evaluate wire bond feasibility while constructing the pad ring. While it's unlikely the feasibility results would be used for manufacturing, they provide necessary feedback on the quality of pad ring placement.



Kevin Rinebold
(Rinebold@sigrity.com) is a senior product marketing manager at Sigrity.