DesignCon 2009

System IO Planning and Design Feasibility – Challenges and Solutions

Kevin Rinebold
Product Marketing Manager
rinebold@cadence.com

Abstract

This paper will briefly examine the current trends driving the need for coordinated system IO planning with early design prototyping and feasibility. It will identify the key challenges in adopting and implementing an IO planning solution, and will review common flows and methodologies in use today. It will introduce a new methodology for dynamic system IO planning across the multiple domains of chip, package, and printed circuit board (PCB) based on newly available EDA technology.

Author Biography

Kevin Rinebold is a Senior Product Marketing Manager at Sigrity responsible for advanced packaging and design planning products. He has more than 15 years of experience in advanced packaging, working closely with key industry leaders.

Prior to Sigrity, Kevin has held positions at Synopsys, Avant!, and Xynetix. He has authored and presented numerous technical papers on advanced packaging and design planning and is a regular contributor to industry journals.

Introduction

A view of trends driving the need for system IO planning shows a convergence of increasing complexity, cost, and time-to-market. IO count, high-speed interfaces, and power distribution often are cited in the need for coordinated design planning – and correctly so – but they're not the only factors driving companies to adopt new tools and methodologies for system IO planning. System compatibility, package level integration in the form of system-in-package (SiP) or package-on-package (PoP), availability and completeness of silicon IP, and package lead-times further fuel the tooling and methodology transition. Many companies are abandoning their traditional serial flow for chip-package-board design in favor of more holistic and concurrent approaches to keep pace with technology and market conditions.

The objective of system IO planning is to coordinate device placement with associated pin and net assignments within the chip-package-board system – prior to detailed chip implementation – when the options to effect change are greatest and cheapest to implement. Achieving this objective is a multi-domain balancing act of evaluation and trade-offs. It requires early insight into aspects of the detailed package layout like wirebond configurations, flip-chip escape patterns, and route feasibility. Visibility into board level connectivity and its influence when developing socket-compatible devices is imperative. Ability to understand chip-level logic restrictions like hard macros or high-speed interfaces and their impact on IO pad ring layout and subsequent net list is also a must. Not only does this require the ability to simultaneously work with multiple data sources, but also the ability to work through incomplete and changing design content.

Successful adoption of a system IO planning methodology can result in fewer and faster design iterations reducing project complexity and cost. Finding and fixing cross domain issues while still in the design planning stage avoids overdesign and its associated cost impact. This may come in the form of package cost savings from fewer layers and vias, or reduction in decoupling capacitors due to more efficient specification of driver strengths. For many companies, achieving these goals remains elusive due to limitations in tools and methodologies. The resulting IO plan is more often "it's as good as it gets because we're out of time" rather than a conscious coordinated design effort.

Tool and Methodology Limitations

Attempting any type of coordinated design planning across the chip, package, and board using traditional tools and serial methodologies can be challenging and frustrating at best. One problem is separate design environments and databases – one for the chip, a second 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 short-coming is it's based on snap-shots of static data, resulting in a highly iterative, error-prone process that does little to reduce cycle-time or cost of results.

Package level integration in the form of SiP, PoP, or thru-silicon via (TSV) packaging creates additional challenges for traditional tools and methodologies. The multi-chip aspect of these packages adds the dynamic of chip-to-chip connectivity in addition to chip-to-package. Designers often use the fixed IO of one chip to influence IO and connectivity assignments on adjacent chips. While chip floorplanning 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 IO placement and assignment.

Applications and Challenges

Project imperatives such as designing for system compatibility, coordinating flip-chip bump patterns, or dealing with package lead-times necessitate IO planning and design feasibility. Other situations not as obvious but have just as big impact on process and productivity include; designing for multiple package variants, addressing missing or incomplete libraries or silicon intellectual property (IP), or resolving differences between logical and physical data. Furthermore, new 3D package technologies like TSV and PoP demand coordinated IO planning across their multiple substrates for successful implementation.

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 device socket on the board becomes the influencing factor driving connectivity upstream through the package and back into the chip. This bottom-up approach necessitates elements of package design to derive a 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 suitable for both chip and package.

The current pitch of flip-chip bumps is in the 170-200um range and is expected to decrease to 150-160um with the next generation. This could further drop to 130-150um in the 2009-2010 timeframe as copper pillar technology is adopted. As semiconductor process nodes continue to shrink, chips are increasingly becoming IO pad limited with a clear impact on die size. Given these trends, developing an efficient flip-chip bump interface is only going to get harder. Multiple facets of chip and package design must be evaluated simultaneously, ideally within the same tool, to derive the ideal bump pattern.

During feasibility studies or early design planning, it's not uncommon for design decisions to be made in the absence of IO libraries or detailed IP information. Frequently the targeted IP hasn't been designed or purchased at the time preliminary judgments must be made. Therefore, a placeholder is inserted that approximates the size and pin count so feasibility or planning can proceed, with replacement and validation occurring later once the IP is available. Effective system IO planning requires the flexibility to instantiate missing or virtual data on-the-fly, work at different levels of abstraction, and use best available data.

Another case for system IO planning is the use of package variants, sometime referred to as application-specific packaging. In this case, one chip is designed into multiple package configurations depending on end-market applications. The challenge is to derive an IO pad ring layout that works equally well for the range of configurations. Traditional flows with a "one die, one package" orientation are cumbersome and highly iterative.

Syntactical differences in net names between domains can be encountered during system IO planning. 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 IO planning they must be correlated and mapped without changing their respective net lists. A similar situation exists when there's a need to map several instances from the chip to one instance on the package, as in the case of multiple on-chip VDD nets connecting to a common VDD net on the package.

New Tool Options

Technology trends and market conditions validate the need for system IO 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 various design groups. It's a step in the right direction, but has minimal impact on cycle-time or cost.

Commercial solutions have emerged to help address the growing challenge of coordinated IO design planning. The next generation has been the linkage of traditional chip, package, and board designs tools by a common conduit. This conduit or wrapper facilitates the exchange of pin and net information between domains, but in a serial fashion using static data.

A new generation of IO planning solutions, such as Sigrity's OrbitIO Planner, takes a more revolutionary approach, bringing all data sources together into a common, unified planning environment. Placement and connectivity scenarios are easily derived and evaluated in the context of the full system. Feasibility functions provide the means to incorporate aspects of detailed implementation while still in the early stages of design planning. A unified chip-package-board data model facilitates the seamless flow of data between domains allowing changes to automatically propagate to adjacent designs where their impact can be immediately evaluated. This ability to optimize the IO and connectivity design plan for performance, cost, and manufacturability prior to detailed implementation will result in fewer and faster design iterations with an overall reduction in complexity and cost.

System IO planning and feasibility tools require innovative functionality to manage and manipulate a range of data at various stages of completeness. Ease of implementation and usability are crucial because these tools must plug into existing flows with minimal disruption. They must be able to instantiate design information on-the-fly for timely planning and feasibility in the absence of detailed data. Solutions must be vertically-aware to support the growing use of 3D packaging and provide the versatility to model die attachment scenarios utilizing wire bonding or flip-chip. In the case of OrbitIO Planner there are four major components that comprise the solution; data management and integration, device placement, feasibility tools, and connection planning.

Data Management and Integration

The ability to make placement and net changes in one domain and immediately see the impact on adjacent domains is made possible by an underlying data model that unites chip, package, and board data. The unified data model not only serves as a data repository, but also tracks the originating data formats and any incremental changes to ensure proper back-annotation. The model supports data mapping across designs while maintaining integrity of the individual databases. Finally, it provides extendibility to support multiple instances of similar design types within the same model.

Populating the unified data model is accomplished utilizing a variety of standard data formats. File formats such as LEF/DEF, Verilog, and VHDL are the common chip-related data sources. LEF/DEF readers accommodate various data constructs and are capable of extracting the pertinent IO related information to avoid excessive data size. Verilog/VHDL readers possess intelligence to identify discrepancies between existing physical data and incoming logic. Spreadsheet support for IO pad ring definition is still necessary as companies take incremental steps to improve methodology. Native file formats from Sigrity, Cadence, or Zuken are sometimes used for packaging data. In their absences, an industry standard format like AIF can be used or even a simple spreadsheet that describes the ball pad map of the BGA. Ability to import PCB data from popular CAD systems also is supported.

Third generation IO planning tools provide capability to instantiate missing or virtual data on-the-fly to work through missing or incomplete design content. This can range from defining a simple die outline or BGA pattern, to adding IO pads cells on-the-fly, or even defining a placeholder for a chip-level macro. As detailed content becomes available the virtual content is replaced and validated using the tool’s engineering change functions.

Functionality to process engineering changes is utilized as design content is constantly changing during IO planning. These functions are capable of processing incremental updates so previous work is preserved. Common change scenarios include; LEF template updates, deleting or replacing contents of a device, net list changes, replacing one device with another, or the simple deletion of a device. In certain situations a compare and merge function can be helpful in identifying data differences.

Once the unified data model is populated, the relationships between devices must be managed, i.e., which chips go in which package and in what order. Automated hierarchy management is the fabric that ties everything together in the IO planning solution. It enables representation of the complete system from the gate level through the PCB while maintaining integrity of the individual designs. Almost every function within the IO planning tool will references the hierarchy before performing its designated task. (Figure 1)

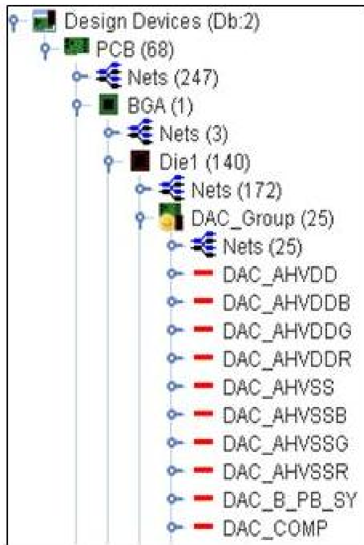

Figure 1: Design Hierarchy

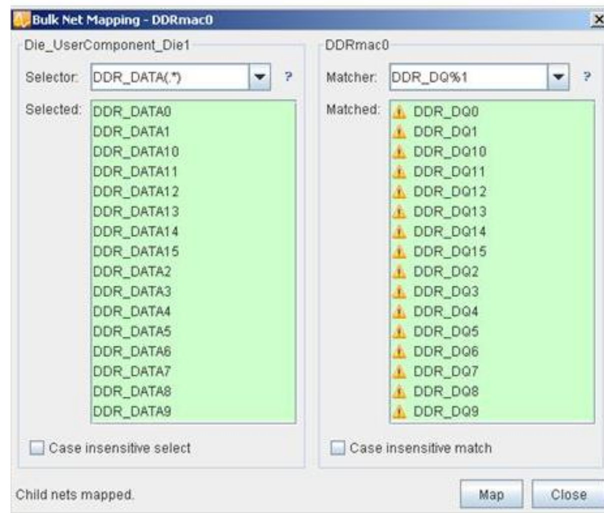

Figure 2: Mapping Net Names

When designs from different sources come together it's not unusual to encounter differences in net naming conventions. These differences must be mapped and correlated between domains before IO panning can take place. This is done through automated pattern recognition or with guidance from user specified regular expressions. (Figure 2)

Tracking incremental changes as well as remembering original data formats is a function of the unified data model that comes into play when it's time to export data. The target tool and scenario will dictate the scope of content and file format. Some scenarios may require incremental changes to be exported as a script for an IC layout tool while others may require complete content in LEF/DEF or spreadsheet formats. There are situations that require filtering of content as in the case of excluding physical-only or PG devices from a Verilog file. Exporting die pad content from the IO pad ring requires collapsing multiple device instances into a single multi-pin component for use in package implementation tools. Not only do export functions support a variety of formats, they possess intelligence to recognize hierarchy and present the appropriate options.

Device Placement

A combination of automated and interactive features are used for device placement and optimization. These features are transparent across domains and easily applied to various situations. Adaptability to tasks ranging from rough IO pad ring layout to detailed placement of overlay cells is required. Results adhere to substrate-specific technology rules or region-based personalities, as in the case of multiple voltage planes. Device placement and pin assignments consider requirements for high-speed interfaces that utilize differential signals or special net topologies.

Pad ring placement can be accomplished in a variety of manners and is dictated by design flow and methodology. In some flows, initial placement originates from a spreadsheet while others instantiate and place pad cells on-the-fly to construct the pad ring. Third generation IO planning tools include functionality to automatically place IO pad cells for specific scenarios. For example, the ability to tightly pack cells together to determine minimum die size or ability to optimize cell spacing for wirebond technology.

Sequence based automated placement helps expedite pad ring creation for flows that instantiate and place IO pad cells on-the-fly. A user-definable sequence file contains multiple strategies or recipes defining how various cell types are placed relative to one another. Sequences range from simple ordering of signal, power, and filler cells to more complex sequences that define ordering for a DDR2 interface.

In the course of IO planning for SiP it's fairly common to use fixed IO from one or more chips to influence IO and connectivity assignments on an adjacent chip. Finding the right combination of locations, rotations, and pin assignments can be a time-consuming task. Tools like OrbitIO Planner include automation to simultaneously evaluate placement and pin assignments to quickly derive the optimal solution.

Feasibility Tools

Effective IO planning requires insight into aspects of the detailed layout while early enough in the planning stages to effect meaningful change. Wirebond configurations or flip-chip escape routing behave as intermediate connection points between die and package pads. These points can act as a redistribution mechanism with the unintended consequence of reordering the connection schedule. Some third generation IO planning tools include innovative functionality to properly model these scenarios to ensure realistic and usable results. (Figure 3)

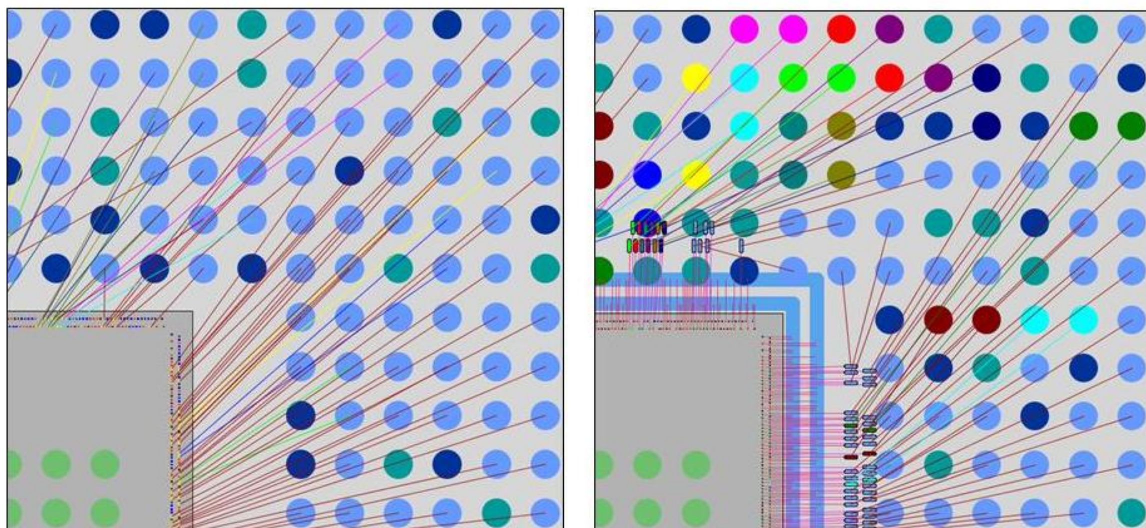

Figure 3: Connection schedule without and with wirebonds

Modeling wirebond configurations not only impacts connection planning, but also helps validate the quality of a pad ring layout. Converging on a mutually acceptable pad ring layout can be a major time-sync and is one of the leading causes of overly complex package layouts. Evaluating wirebond manufacturability while still in the pad ring layout stage will lead to optimal configurations in a much shorter timeframe.

Incorporating powerful feasibility engines into third generation planning tools requires a delicate balance of capability and ease-of-use. Automatic generation routines are completely rules driven as users may not be intimate with a given process, but posses enough knowledge to evaluate feasibility. Wirebond feasibility engines are capable of determining the proper ring configuration and number of bonding tiers. Routing requirements are also taken into account so bond finger spacing reflects the appropriate clearances.

The requirements for redistribution layer (RDL) and bump escape routing must be considered during IO planning for flip-chip devices. RDL routing takes place on the upper most layers of the chip and is used to connect the IO pad cells to their respective bump cells. The required route resources are a direct result of the placement quality of the pad and bump cells. Bump escape routing is preformed on the package substrate and directly impacts layer count and design complexity. Via placement for the escape routes not only impacts routeability, but greatly influences connection scheduling. Just as the case with wirebonding, evaluating flip-chip feasibility in conjunction with IO planning will lead to shorter cycle-times and optimized designs.

Connection Planning

Connectivity planning and creation functions work in concert with the placement and feasibility features of the IO planning tool. These are not isolated events but rather a tightly intertwined puzzle with the results of one impacting the others. Deriving balanced solutions requires a great deal of automation and is scalable to various situations across interconnect domains.

Consider the example of propagating connectivity from flip-chip bumps to package ball pads. One objective is achieving the shortest connections with the least number of cross-overs. Use of differential signals will mean some nets must be on adjacent ball pads to minimize skew. In turn, this can trigger a need for PG balls to be adjacent to the diff pairs. PG ball pad placement must consider the location of its respective voltage plane. This quickly becomes a multi-faceted problem that uses innovative functions to evolve a solution.

Most third generation IO planning tools include robust mechanisms to properly describe the complex interactions and behaviors of connection planning. This starts with the automated definition of differential signals or defining the ratio of signal/power/ground pads. For flip-chip, connections comprehend their routing layers or specific route topology. Mechanisms can be used to group design objects based on common characteristic independent of net assignments. For example, the ability to reserve specific pin groups for certain types of signals.

40% Cycle-Time Reduction

A 40% reduction in IO planning cycle-time has been realized using the new generation of system IO planning and design feasibility tools. This was demonstrated using a 1400 pin flip-chip design that had to be socket-compatible with a printed-circuit board. One design team used their traditional serial methodology, while a parallel effort employed a concurrent design planning methodology using a system IO planning tool. The team using the serial methodology took approximately twenty weeks to sign-off the IO floorplan. The team using the system IO planning tool achieved sign-off in less than twelve weeks.

This case study was preformed by a medium-sized fabless semiconductor company. The package substrate utilized an expensive build-up technology that resulted in it being half the finished device cost. The design experienced eight die size revisions as functionality and cost were tweaked. Close to 30 different bump maps were evaluated with 40+ iterations of the package layout. Most package iterations were feasibility studies to evaluate new bump maps and their impact on flip-chip escape routing.

The team using the IO planning tool had the advantage of cycling through the flow of IO pad placement, bump layout, and package feasibility within one environment. The other team performed similar tasks but had to iterate between multiple tools. They also went through a re-spin to resolve RDL routeability issue which the other team identified during IO pad and bump placement. (Figure 4)

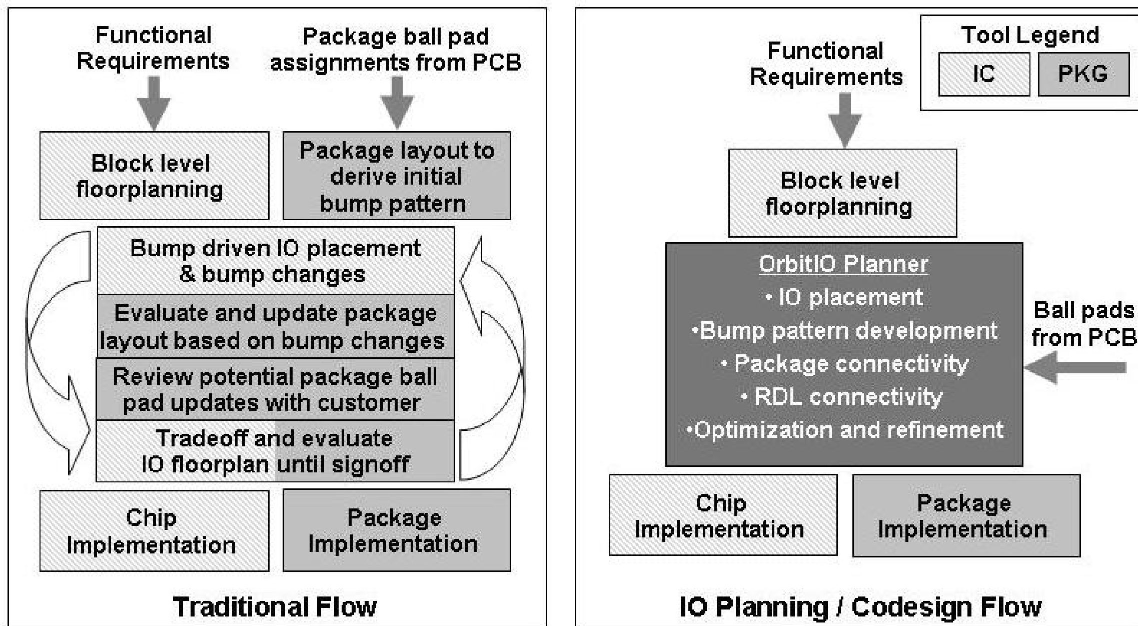

Figure 4: Comparison of design flows

Conclusion

Companies abandoning their traditional tools and methodologies in favor of more holistic and concurrent approaches are realizing the benefit of fewer and faster design iterations with reduction in complexity and cost. In addition to adopting new solutions, these companies recognized the challenges associated with methodology change and the hesitancy to cross domains to resolve design issues. For some, the result is a dedicated codesign team while others task their packaging team to take an active role in IO pad ring placement.

Market conditions and technology trends are converging, driving the need for system IO planning and design feasibility solutions. Inability to plan and coordinate designs for TSV packaging is already identified as a limiting factor in their adoption. Developing an efficient flip-chip bump interface is only getting harder with shrinking process nodes and decreasing bump pad pitch.

Progressive EDA vendors recognize these multi-substrate, multi-domain challenges and view the status-quo of sequential design flows with separate tools and databases as a severely limiting factor. The new generation of planning and feasibility tools supports critical decision making on issues that impact performance, complexity, and cost at a point in the design process when it’s most economical to effect change.