Physical Layout Automation for System-On-Packages

Sung Kyu Lim

GTCAD Laboratory
School of Electrical and Computer Engineering
Georgia Institute of Technology
Atlanta, GA, 30332-0250
limsk@ece.gatech.edu

Abstract

System-On-Package (SOP) paradigm proposes a unified chip-plus-package view of the design process, where heterogeneous system components such as digital ICs, analog/RF ICs, memory, optical interconnects, MEMS, and passive elements (RLC) are all packaged into a single high speed/density multi-layer SOP substrate. We propose a new chip/package co-design methodology for physical layout under the new SOP paradigm. This new methodology enables the physical layout design and analysis across all levels of the SOP design implementation, bridging gaps between IC design, package design, and package analysis to efficiently address timing closure and signal integrity issues for high-speed designs. In order to accomplish a rigorous performance and signal integrity optimization, efficient static timing analysis (STA), signal integrity analysis (SIA), and thermal and power analysis (TPA) tools are fully integrated into our co-design flow. Our unified wire-centric physical layout toolset that includes on-chip/package wire generation, on-chip/package floorplanning, and on-chip/package wire synthesis provides wire solutions for all levels of the design hierarchy—including cell, block, and chip level for pure digital and mixed signal environment. In addition, on-chip hard/soft IP (Intellectual Property) integration is supported in our co-design flow for shorter design times through design reuse. To the best of our knowledge, this paper is the first to address the chip/package co-design issues in System-On-Package (SOP) physical layout.

1. Introduction

1.1. System-On-Package

The increasingly higher integration of transistors at an increasingly lower cost per transistor has resulted in a capability of placing billion transistors on a single chip. Many in the industry believe that this progress will lead to the System-On-Chip (SOC) in most application areas: microprocessors, DSPs, wireless systems, multiprocessor servers, military systems, and computer peripherals. ASIC foundries and EDA vendors see a promising new business opportunity in the SOC paradigm, which extends ASIC design from component to system level. On the other hand, the systems integration community and electronics packaging design vendors see the systems market as an extension of their current business, via the System-On-Package (SOP) [1]. As they see it, SOP will increase their importance in the product supply chain linking electronics packaging directly to product specification, early design, and ASIC design. The SOP paradigm extends the role of electronics packaging from the later stages of the manufacturing process (in the current chip-centered design universe) to the front-end and conceptual phases of the design process. The SOP design paradigm, which facilitates rapid reengineering via reuse libraries, promises a high return on investment at a very low risk within shorter time-to-market cycle, compared to the System-On-Chip (SOC) paradigm.

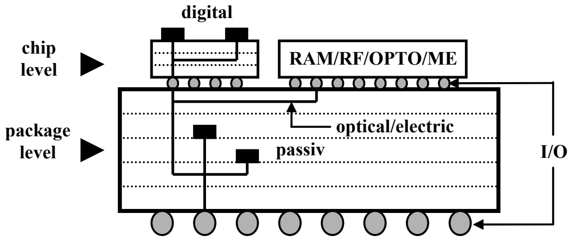

Figure 1 Mixed signal component integration and multi-layer (= 2.5 dimensional) physical layout resource environment of System-On-Package

System-On-Package (SOP) paradigm proposes a unified chip-plus-package view of the design process, where heterogeneous system components such as digital ICs, analog/RF ICs, memory, optical interconnects, MEMS, and passive elements (RLC) are all packaged into a single high speed/density multi-layer SOP substrate as illustrated in Figure 1. The physical layout resource environment of SOP is 2.5-dimensional (= multi-layer) in nature. At the chip-level, traditional 2-dimesional placement and 2.5-dimensional routing is applied to digital subsystem. At the package-level, however, both the placement and routing are done in 2.5-dimensional environments since the passive elements are embedded into the multi-layer SOP substrate.

A high performance mixed signal system employs a lot of passive components up to 30 passive components per an IC [2]. Such passive components are used for simultaneous switching noise reduction, cross talk reduction, network matching, and signal integrity [3]. This is largely because of the inability to successfully integrate all of the needed passive components on the same silicon. As a result, the discrete passive components increase overall system complexity, total cost, and thus compromised performance [4]. The complexity of a radio frequency front-end IC is considerably simpler with high Q passive components. Quality and functionality of RF circuits is extremely sensitive to any unforeseen parasitics. On the other hand, decoupling capacitors perform well when they are close to the source of simultaneous switching noise. Hence, performance and cost of high end microprocessor is also benefit from close vicinity capacitors [5], which effectively stabilize supply and ground noise.

1.2. Physical Design for SOP vs MCM

The SOP can be seen as a convergent technology among packaging, PCB and MCM. However, the physical layout structure of the SOP is more general than that of the MCM, PCB or packaging technology taken alone. There are significant similarities as well as differences between MCM and SOP. Both technologies use more than the usual four layers (in IC designs) for routing. Among the differences, SOP can have a number of placement layers unlike the MCM. It has one I/O pin layer at the bottom through which various components can be connected to the external pins. The placement layers contain the IC or embedded passive blocks, which from the point of view of physical design is just a geometrical object with pins. In some cases where these blocks are a collection of cells, the pins may not be assigned and pin assignment needs to be done to determine their exact location. The interval between two placement layers is called the routing interval. The routing interval contains a stack of signal routing layers sandwiched between pin redistribution layers. We also allow routing to be done in the pin redistribution layers.

However, the most important difference pertains to the nets. The nets in MCM have all their pins located in the top layer or the I/O pin (= bottom) layer. SOP nets can have pins in any of the floorplan (= intermediate) layers. Although some of the methodologies used for physical design of MCM can still be applied to the SOP, the unique and distinguishing features for the SOP necessitates tuned approaches. The number of layers for routing is a very important optimization objective, since the number of layers required may be much more than what is currently being used in the MCMs.

1.3. Wire-centric Chip/Package Co-design

A traditional IC design flow moves from circuit design to package design to board design and then to manufacturing. At each step, designers and engineers try to optimize their segment of the process, unaware of the impact their decisions have on the other steps. Until clock rates headed to the microwave range and package areas plummeted, this worked. But today such an isolationist approach does not reliably and efficiently produce high-performance products. Because of the complexities involved, SOP design engineers need to understand and analyze the influence of the IC and its package on system performance early in the design cycle. To meet this need, we need to co-design the chip and the package, reducing product development times. Designing chip and package concurrently for SOP physical layout can save significant time and money when compared to a traditional isolationist design flow. Product reliability can also be improved when all members of the design team reach across the design flow to consider chip and package for SOP technology.

Implementing nanometer-scale ICs begins and ends with wires. Wires are so dominant that little is known about a design's performance without them. In fact, nanometer design strategies that are not clearly focused on rapid wire creation, optimization, and analysis are destined to fail. A new generation of tools exists to address the problem of design closure for the physical implementation of nanometer designs. These tools consider the wire and interconnect dominance that characterizes the small geometries of nanometer design. The scaling of these geometries is paralleled by an increase in the capacity of the chips, which exceeds the capacity of most design tools.

We propose a new chip/package co-design methodology for physical layout under the new SOP paradigm. This new methodology enables the physical layout design and analysis across all levels of the SOP design implementation, bridging gaps between IC design, package design, and package analysis to efficiently address timing closure and signal integrity issues for high-speed designs. In order to accomplish a rigorous performance and signal integrity optimization, efficient static timing analysis (STA), signal integrity analysis (SIA), and thermal and power analysis (TPA) tools are fully integrated into our co-design flow. Our unified wire-centric physical layout toolset that includes on-chip/package wire generation, on-chip/package floorplanning, and on-chip/package wire synthesis

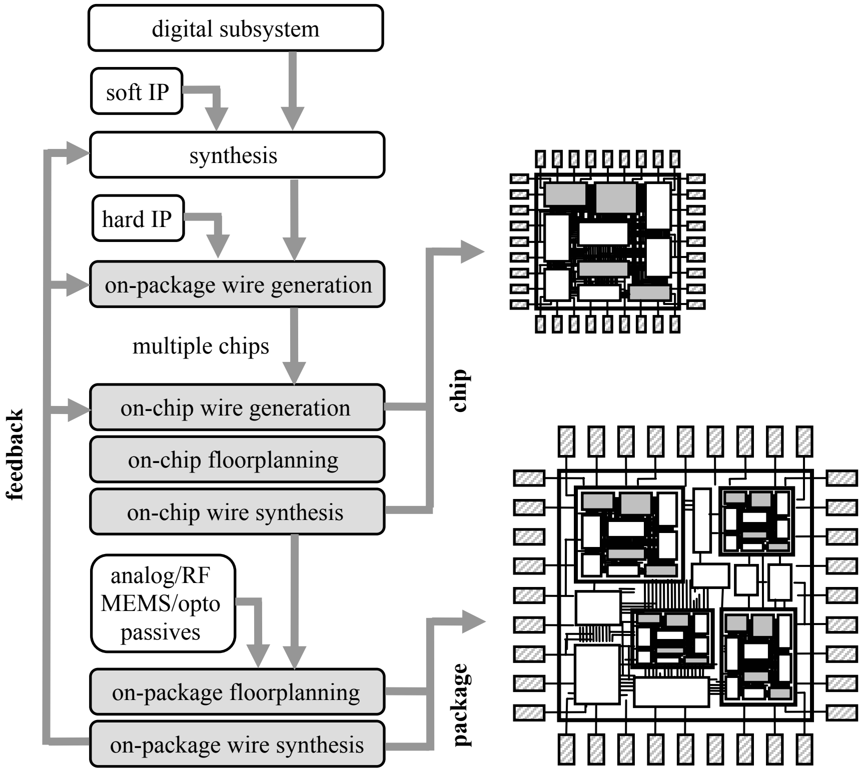

Figure 2 Overview of our wire-centric chip/package co-design flow for System-On-Package physical layout. Hard/soft IP integration is also shown.

provides wire solutions for all levels of the design hierarchy—including cell, block, and chip level for pure digital and mixed signal environment. In addition, on-chip hard/soft IP (Intellectual Property) integration is supported in our co-design flow for shorter design times through design reuse.

2. Overview of the Co-Design Flow

An overview of our wire-centric chip/package co-design flow for SOP physical layout is shown in Figure 2. The digital subsystem is first synthesized together with soft IP to generate place-and-routable gate-level netlist. This netlist along with hard IP is the input to our on-package wire generation step.

1. *On-package wire generation*: during this step, partitioning is performed to divide the netlist into multiple chips. Under our wire-centric design paradigm, partitioning is seen as the crucial step that defines the local and global wires—intra-partition connections become on-chip wires, whereas inter-partition connections become on-package (or off-chip) wires in this step. The objective is to minimize the amount of wires, the longest path delay, and power consumption induced by the partitioning under pin, area, and thermal constraints.

Each chip obtained by the prior package-level partitioning is the input to our front-end chip-level planning that consists of the following three steps.

2. *On-chip wire generation*: the netlist contained in each chip is further divided into multiple blocks for divide-and-conquer during this step. Under our wire-centric design paradigm, intra-partition connections become on-block wires, whereas inter-partition connections become on-chip (or off-block) wires in this step. The objective is to minimize the amount of wires, the longest path delay, and power consumption induced by the partitioning under area and thermal constraints.

3. *On-chip floorplanning*: the blocks generated during the prior partitioning and IPs find their locations during this step. The dimension of hard IP blocks is fixed, and the remaining blocks assume a certain set of possible aspect ratios. Thus, the placement of the blocks is of the primary concern during this step. In addition, I/O pin assignment is performed. The main objective during this step is to minimize the area of the final chip, total estimated wirelength, and the longest path delay. The major

constraints to be considered are routability and power-ground noise.

4. *On-chip wire synthesis*: pin assignment and global routing are performed during this step. Our goal is to perform global routing and pin assignment simultaneously for more rigorous performance optimization in a shorter runtime. In order to model and handle congestion and obstacles efficiently, we model the routing resource with graph. In addition, our graph-based Steiner tree generation is based on the computation of Steiner Arborescence, where the distance between every source-sink path is the shortest. The objective is then to minimize the total wirelength, which has a direct impact on capacitive load for the driver. The major constraints to be considered are congestion and cross-talk noise.

Note that we do not perform gate placement and detailed routing during our front-end chip-level planning. The purpose is to obtain a quick physical layout prototype of the chips and to use them in our back-end package-level planning so that a fast convergence is maintained within our feedback loop between chip-level and package-level planning. When our package-level prototype reaches the desire quality, gate placement and detailed routing are performed for the blocks in the chips.

The digital ICs obtained during the chip-level planning together with analog/RF ICs, MEMS, Optical interconnect components, and embedded passive elements are the input to our package-level planning that consists of the following two steps.

5. *On-package floorplanning*: I/O pin location and chip/component dimension are already determined. Thus, the placement of these mixed signal components is of the primary concern during this step. The embedded passive elements are placed in the intermediate layers within the SOP substrate while other components are mounted on the surface. The main objective during this step is to minimize the area of the final chip, total estimated wirelength, and the longest path delay measured in a multi-layer environment. The major constraints to be considered are routability and power-ground noise. In addition, some analog ICs and passive elements need to be placed together in order to maintain high Q, and analog and digital ICs will need to be separated apart due to various noise issues.

6. *On-package wire synthesis*: global routing and layer assignment are performed during this step. Unlike the traditional multi-layer routing, pins are possibly located at all intermediate layers in SOP substrate rather than top-layer only since chips now have connections to embedded passives. In addition, these embedded passives are obstacles during routing. Our goal is to perform global routing and layer assignment simultaneously for more rigorous performance optimization in a shorter runtime. The objective is to minimize the total wiring cost in terms of the number of layers and vias and the longest path delay. The major constraints to be considered are congestion and cross-talk noise.

Upon the termination of package-level planning, layout information and its measurement on the area, speed, and the power consumption of the package are fed back to our front-end chip-level planner or even to synthesizer. Then the front-end steps use this feedback to improve the last solution through incremental updates. When our package-level prototype reaches the desire quality, we optionally perform various post layout interconnect optimization such as buffer insertion, wire sizing, etc, on our package-level layout during the last iteration of the co-design flow.

3. Co-design Flow Components

3.1. On-Chip/Package Wire Generation

We formally state the definitions of the on-package and on-chip wire generation as follows:

Definition: On-package Wire Generation

- Instance: (i) a set of gates and IP blocks in a digital subsystem and its netlist, (ii) area and delay information of the cells (= gates and IP blocks), (iii) *K* (= # of chips desired), and (iv) area, pin, and thermal constraints for the chips.
- Question: Is there a partition *P* of cells into *K* chips such that cutsize(*P*), delay(*P*), and power(*P*) are minimized?
- Constraints: the *K* chips satisfy the area, pin, and thermal constraints.

Definition: On-chip Wire Generation

- Instance: (i) a set of gates and IP blocks in a chip and its netlist, (ii) area and delay information of the cells (= gates and IP blocks), (iii) *K* (= # of blocks desired), and (iv) area and thermal constraints for the blocks.
- Question: Is there a partition *P* of cells into *K* blocks such that cutsize(*P*), delay(*P*), and power(*P*) are minimized?
- Constraints: the *K* blocks satisfy the area and thermal constraints.

The delay minimization during partitioning is computationally harder than cutsize optimization since delay optimization involves heavy use of path-based timing analysis, which involves an analysis of the entire netlist even for a small local change in the current solution. On the other hand, cutsize optimization is done efficiently without relying on any global netlist analysis. Thus, an efficient timing analysis engine such as incremental STA (static timing analysis) is inevitable to efficiently adopt local perturbation and give quick feedback to the optimizer. The problem becomes complicated, however, if the given circuit is sequential since feedback loops via latches or flip-flops exist in most of the sequential circuits. A recent advancement in handling loops in sequential circuit timing analysis is proposed [6]. The RTA (Retiming based Timing Analysis) engine [6] not only performs timing analysis to produce timing slack information, but it also predicts the timing given that an optimal retiming is to be applied afterwards. Our research focus is on the extension of the retiming based partitioning is to consider the power consumption and thermal management. If we ignore these issues during partitioning, we might generate blocks or chips that consume more power and thus dissipate more heat than others. These “hot blocks" or “hot blocks” in turn cause more problems during the subsequent placement steps—the subsequent floorplanning has to prevent these hot blocks from being placed together, which increases the computational burden of the placement.

We note that the power consumption profile of the blocks can be obtained from switching activities of the gates partitioned into them. There are several works proposed in the literatures [7][8] that compute these switching activities from the given gate-level netlist for combinational and sequential circuits. Thus, the power consumed by the blocks during partitioning is the sum of switching activities of the gates partitioned into them. Then our partitioning formulation is as follows: minimize the cusize and delay under the area and switching activity constraints, where the switching activity constraint is in the form of upper bound in switching activity of the blocks. In addition, we distribute some “hot cells" evenly to the blocks and lock them during the partitioning so that these cells with higher switching activities remain evenly distributed throughout the optimization process. It is challenging to observe the impact of this new thermal constraint introduced to partitioning and prevent the potential negative impact on cutsize and/or delay.

In terms of power minimization, we plan to pay closer attention to nets that are driven by cells with high switching activities. If these nets are cut during the partitioning, the longer length and thus the bigger capacitance of these global wires will cause the driving gates to consume more power. Thus, the switching activity based net-weighting is a simple and useful technique to guide the partitioning process for power optimization. The challenge, however, is that the cutsize or delay might be compromised while the partitioner is giving more attention to power optimization. Therefore, identifying a tradeoff among cutsize, timing slack, and switching activity based objectives is an important issue to be addressed.

3.2. On-Chip/Package Floorplanning

We formally state the definitions of the on-chip and on-package floorplanning as follows:

Definition: On-chip Floorplanning

- Instance: (i) a set of blocks to be placed on a chip and its netlist, (ii) dimension and delay information of the blocks, and (iii) chip aspect ratio (= width/height) and noise constraints.
- Question: Is there a floorplan *F* of the blocks onto a 2-dimensional chip such that area(*F*), wirelength(*F*), and delay(*F*) are minimized?
- Constraints: the chip satisfies the aspect ratio and noise constraints.

Definition: On-package Floorplanning

- Instance: (i) a set of modules (= digital ICs, analog/RF ICs, memory, MEMS, passive elements, and decoupling capacitances) to be placed onto a multi-layer SOP package and its netlist, (ii) dimension, delay, and power consumption information of the modules, and (iii) package aspect ratio (= width/height) and noise constraints.
- Question: Is there a floorplan F of the modules onto a multi-level SOP package such that area(*F*), wirelength(*F*), and delay(*F*) are minimized?
- Constraints: (i) the ICs, memory, and MEMS are placed on the top layer, and the passive elements and decoupling capacitances in the multi-layer SOP substrate, (ii) the package satisfies the aspect ratio and noise constraints.

We handle the following types of floorplanning constraints [9][10] existing in a high performance mixed-signal SOP design:

a) signal integrity, where decoupling components are placed near I/Os or ICs: we use *point constraint* to specify/enforce which block needs to make contact with which point (x,y,z) in multi-later SOP.
b) power integrity, where digital and analog ICs are placed in different voltage islands: we use *region constraint* to specify/enforce which blocks need to be contained in the given bounding box.
c) timing convergent, where blocks from a critical path are placed one after the other: we use *abutment constraint* to specify/enforce which blocks need to be placed next to each other.
d) interposer/interface, where I/O blocks are placed near boundaries of a peripheral pin package: we

use *boundary constraint* to specify/enforce which blocks need to be placed nearby which boundary of the SOP.

e) physical hierarchy, where related functional blocks are placed close together: we use *group constraint* to specify/enforce which blocks need to be placed together.

We use a penalty cost as a measurement of constraint violation and minimize it during simulated annealing [11]. This approach simplifies the complexity of perturbation and the handling of infeasible floorplan. In addition, it reduces the additional run time required to build and modify graphs for representing constraints.

3.3. On-Chip/Package Wire Synthesis

We formally state the definitions of the on-chip and on-package wire synthesis as follows:

Definition: On-chip Wire Synthesis

- Instance: (i) a set of blocks and its netlist, (ii) location, dimension and delay information of the blocks.
- Question: Is there an assignment B of the pins to the boundary of the blocks and a routing *R* of the nets onto a 2-dimensional chip such that cost(*R*), delay(*R*), and noise(*R*) are minimized?
- Constraints: the routing solution satisfies the congestion constraints and avoids obstacles

Definition: On-package Wire Synthesis

- Instance: (i) a set of modules (= ICs, memory, MEMS, embedded passives) and its netlist, (ii) location, dimension and delay information of the modules.
- Question: Is there a routing R of the nets onto a multi-level SOP package such that cost(*R*), delay(*R*), and noise(*R*) are minimized?
- Constraints: the routing solution satisfies the congestion constraints and avoids obstacles

We model the placement layer in the SOP as a floor connection graph [12]. The routing layer in the SOP can be modeled either as a uniform or a non uniform gird graph. These two kinds of are connected through via edges. For large SOPs memory will be a concern. An alternative idea for routing layer resource representation is as a collection of net entry/exit points. The routing will be done by area router, which will intuitively result in finer routing assignment, but at the expense of larger runtimes. The use of grid graph facilitates development of simple and efficient algorithms with good runtimes. The advantages of using this model is that algorithms can consider block pin assignment, global routing, via assignments simultaneously.

The purpose of this routing resource model is to be able to handle pin assignment and global routing simultaneously. The components which must be taken into account in the model are the regions through which the nets can be routed and coarse location from where the nets can originate. To this end, we model the blocks in the floorplan as Block Nodes (BN) as illustrated in Figure 3. The nets can cross over to the adjacent routing layers only through the regions in the channel. The channel itself is represented by Channel Nodes (CN). The actual blocks form blockages for the nets, which cannot be routed through them. The nets can switch from floorplan layer to the routing layer only through designated regions which are represented as Layer Switch Nodes (LSN) in the resource graph. The LSN in this case are simply four corners of the blocks. They denote regions rather than points through which nets will traverse to adjacent routing intervals. The routing layers can be represented by a grid graph, each node specifying a region in the layer and edges representing the adjacency between regions. The nodes are called Routing Nodes (RN).

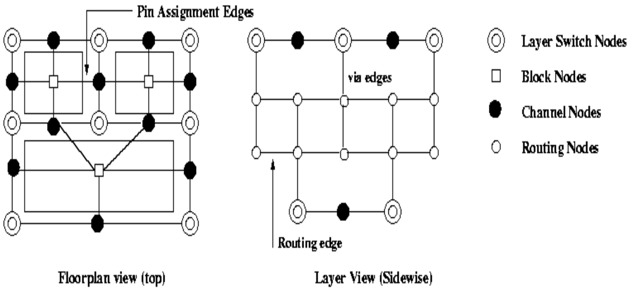

Figure 3. The floorplan and the layer view of the resource graph with node and edge types.

The edges between channel nodes and block nodes are called Pin Assignment Edges (PE). This makes it possible to do pin-assignment while doing global routing. The pin assignment capacity is the maximum number of pins which can be assigned towards a particular channel. The edges between the layer switch node and routing node is defined as Via Edges. The capacity of this edge is the maximum number of nets which can cross between two regions in the two layers. The via edges also exist between two adjacent routing layers (actually layer pairs). Finally, the routing edge capacity is the number of nets which can pass through the routing regions.

In the SOP model the nets can be classified into two categories. The nets which have all their terminals in the same floorplan are called i-nets, while the ones having terminal in different floorplans will be referred to as x-nets. The i-nets can be routed in the single routing interval or indeed within the placement layer itself. However, for high performance designs routing such nets in the routing interval immediately above or below the placement layer maybe desirable and even required. On the other hand, the x-nets may span more than one routing intervals. The only case where one routing interval may suffice is when the terminals of the net are located in either of the floorplans immediately above or below the routing interval. We associate two integers l and h with each net such that is the lowest floorplan and is the highest ordered floorplan in F containing pins of the net. If l and h are equal for a particular net, the net is i-net else the net is x-net. The difference between the two is the span of the net. Greater the span of the net, more number of routing intervals (and placement layers) the nets must go through leading to increased demands in the actual number of layers required per routing intervals. The nets encountered in the MCM model are i-nets and nets with span utmost one. The SOP algorithms must handle x-nets in all levels of physical design. For example one of the objectives of the SOP floorplanner may be to reduce the span of the nets while assigning blocks to different floorplan layers.

4. Conclusions

In this paper, we have emphasized the need for new techniques and models to solve the new and emerging SOP technology. The physical design of SOPs is significantly different from the traditional way in which the physical design is done for the existing technologies such as PCB and MCM. Although conventional approaches can still be used to solve some of the problems, there is a big scope for achieving higher efficiency by independently investigating the issues involved. Our unified wire-centric physical layout toolset that includes on-chip/package wire generation, on-chip/package floorplanning, and on-chip/package wire synthesis provides wire solutions for all levels of the design hierarchy—including cell, block, and chip level for pure digital and mixed signal environment.

To the best of our knowledge, this paper is the first to address the chip/package co-design issues in System-On-Package (SOP) physical layout. The concept of SOP is still under rapid development, and the proposed research tries to identify and formulate new problems existing in this emerging technology. We believe that this research promotes collaborative research efforts between IC design community and IC packaging community to face new challenges emerging in the new SOP technology together.

5. Reference

[1] Rao Tummala and Vijay Madisetti, “System on Chip or System on Package?”, IEEE Design & Test of Computers, pp 48-56, 1999.

[2] S. Dalmia, J.M. Hobbs, et al, “Design and optimization of high Q RF passives on SOP-based organic substrates," Electronic Components and Technology Conference, pp 495-503, 2002.

[3] Robert C, Frye, “MCM-D Implementation of Passive RF Components: Chip/Package Tradeoff,” IEEE Symposium on IC/Package Design Integration, pp 100-104, 1998.

[4] Lesley Polka, Shamala Chickamenahalli, et al., “Package-level Interconnect Design for Optimum Electrical Performance," Intel Technology Journal, Microprocessor Packaging, 3rd Quarter, 2000.

[5] Alex Waizman, Chee-Yee Chung, “Package Capacitors Impact on Microprocessor Maximum Operating Frequence," Electronic Components and Technology Conference, 2001.

[6] J. Cong and S. K. Lim, "Physical Planning with Retiming", IEEE International Conference on Computer Aided Design, pp 2-7, 2000.

[7] C-Y. Tsui, J. Monteiro, M. Pedram, S. Devadas, A. M. Despain and B. Lin, "Power estimation in sequential logic circuits," IEEE Trans. on VLSI Systems, Vol. 3, No. 3 (1995), pp. 404-416.

[8] M. Pedram, "Power estimation and optimization at the logic level," Int'l Journal of High Speed Electronics and Systems, pp. 179-202, 1994.

[9] E. Young and C. Chu and M. Ho, “A unified method to handle different kinds of placement constraints in floorplan design”, Proc. International Conference on VLSI Design, pp 661—667, 2002.

[10] F. Young and D. Wong, “Slicing Floorplans with Pre-placed Modules,” Proc. IEEE International Conference on Computer-Aided Design, pp 252–258, 1998.

[11] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, “Optimization by Simulated Annealing,” Science, pp 671-680, 1983.

[12] J. Cong, “Pin Assignment with Global Routing for General Cell Design,” IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, pp 1401—1412, 1991.