

# An Interconnect-Centric Design Flow for Nanometer Technologies

Jason Cong

Department of Computer Science

University of California, Los Angeles, CA 90095 \*

## ABSTRACT

As the IC devices is scaled into nanometer dimensions and operates in giga-hertz frequencies, interconnect design and optimization have become critical in determining the system performance and reliability.

In this talk, I shall present our research effort at UCLA on developing an interconnect-centric design flow, including interconnect planning, interconnect synthesis, and interconnect layout. I shall focus on recent advancements on interconnect planning, including physical hierarchy generation, floorplanning with interconnect planning, and interconnect architecture planning, and discuss their impact on future circuit designs.

## 1 INTRODUCTION AND OVERVIEW OF THE FLOW

With rapid feature size scaling, the circuit performance is increasingly determined by the interconnects instead of devices. Given the dominating importance of interconnects in current and future generations of IC designs, our group at UCLA has been developing a new design flow, named *the interconnect-centric design flow*, in the past several years. In conventional VLSI designs, much emphasis has been given to design and optimization of logic and devices. The interconnection was done by layout designers or automatic place-&-route tools as an afterthought. In interconnect-centric designs, we suggest that interconnect design and optimization be considered and emphasized throughout the design process (see Figure 1). Such a paradigm shift is analogous to the one happened in the software design domain of 1970s. In the early days of computer science, much emphasis was placed on algorithm design and optimization, while data organization was considered to be a secondary issue. It was recognized later on, however, that the data complexity is the dominating factor in many applications. This fact gradually led to a data-centric and object-centric software design methodology, including development of the database systems and the recent object-oriented design methodology (see Figure 2). Although algorithms and data repre-

\*This research is partially sponsored by Semiconductor Research Corporation under Contract 98-DJ-605, National Science Foundation Young Investigator Award MIP-9357582, MARCO/Gigascale Research Center and grants from IBM, Intel, and Fujitsu Corporations.

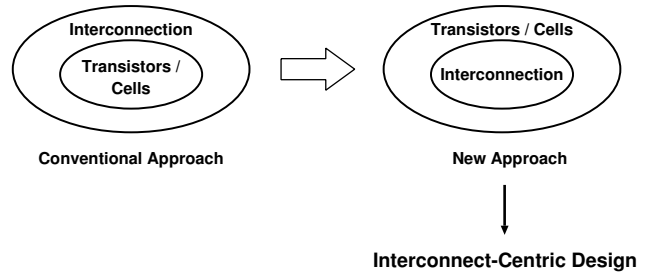


Fig. 1. Proposed paradigm shift for interconnect-centric VLSI design.

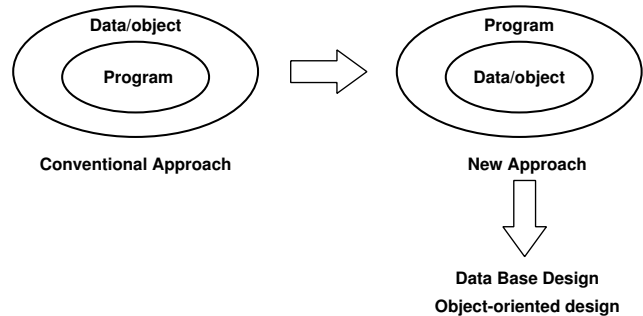


Fig. 2. An analogous methodology change in software design.

sentation/management are integral parts of any software system, the shift in viewpoint from algorithm-centric to data/object-centric designs allows us to effectively manage the design complexity in many large applications. We believe that development of the interconnect-centric design techniques and methodology will greatly impact VLSI designs, similar in the way that database design and object-oriented design methodologies have benefited software development.

Our interconnect-centric design flow and methodology emphasize interconnect planning and optimization throughout the entire design process. It goes through the following three major design phases: (1) interconnect planning, which includes physical hierarchy generation, floorplanning/coarse placement with interconnect planning, and interconnect architecture planning;

(2) synthesis and placement under the physical hierarchy, which performs behavior-level, RT-level, and logic-level synthesis with placement under the physical hierar-

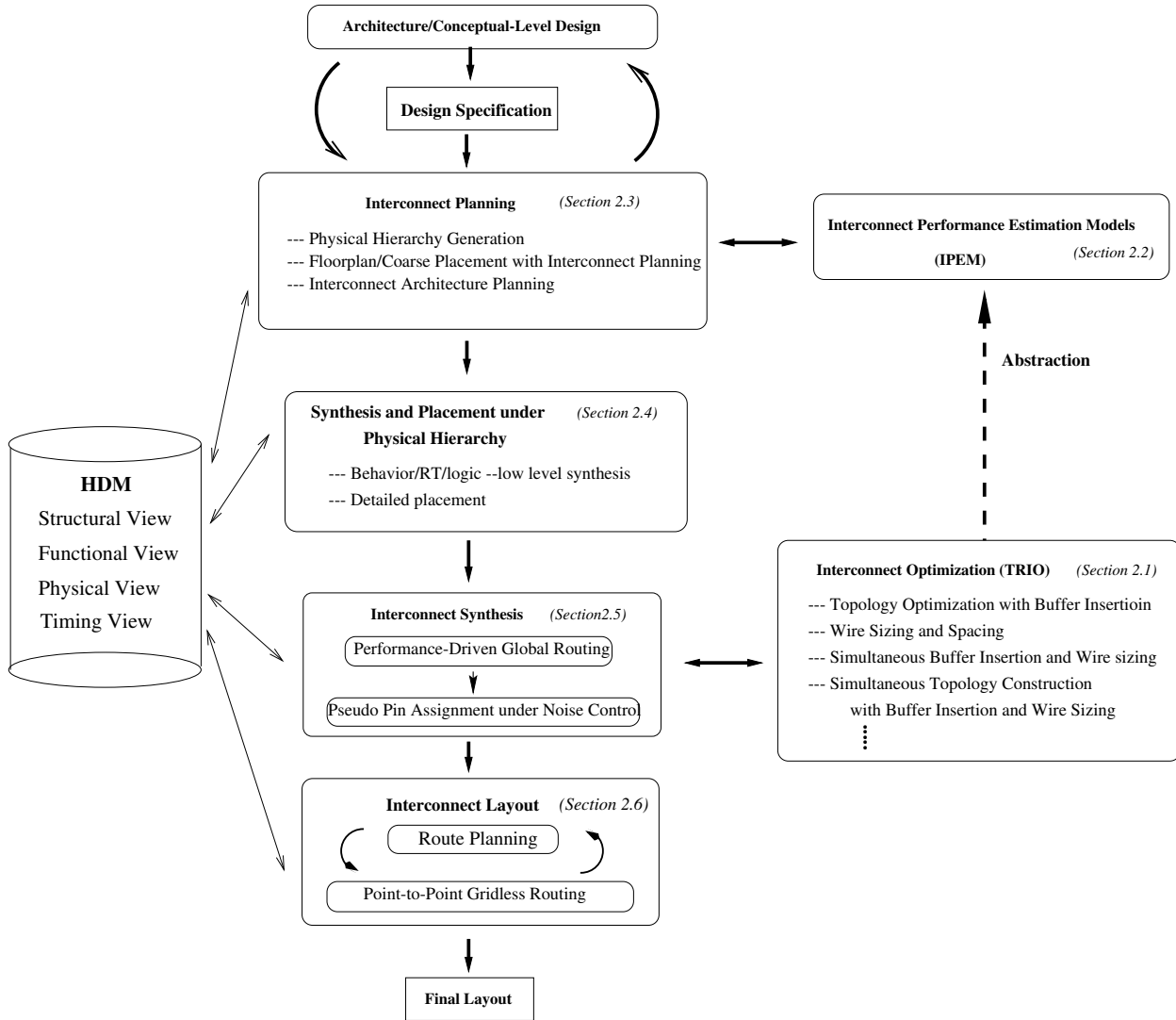


Fig. 3. Overview of our interconnect-centric IC design flow.

chy; (3) interconnect synthesis, which determines the optimal or near-optimal interconnect topology, wire ordering, buffer locations and sizes, wire width and spacing, etc., to meet the performance and signal reliability requirements of all nets under the area and routability constraints; and (4) interconnect layout, which carries out detailed routing to implement the complex width and spacing requirements of all wires using a flexible and efficient multi-layer general-area gridless routing system.

Figure 3 shows an overview of our proposed interconnect-centric design flow. Each building block in this flow will be discussed briefly in the next section. Section 2.1 presents a set of interconnect optimization techniques, which form a key building block in our interconnect-centric design flow. Section 2.2 presents a set of optimized interconnects, which are needed for interconnect planning. With these two building blocks, Sections 2.3 to 2.6 describes four major steps in our interconnect-centric design flow: interconnect planning,

synthesis and optimization under the physical hierarchy, interconnect synthesis, and interconnect layout. Section 2.3 presents our results on interconnect planning; this is the centerpiece of the interconnect-centric design flow, including physical hierarchy generation, floorplanning/coarse placement with interconnect planning, and interconnect architecture optimization. After interconnect planning, we have roughly determined the global interconnect structures for achieving the given performance target when it is possible. Otherwise, an early feedback is given to the circuit/system designer to revise the architecture design or/and the performance target. Section 2.4 describes the step of synthesis and optimization under the physical hierarchy. Section 2.5 describes the step for interconnect synthesis, which integrates various interconnect optimization techniques in multi-layer global routing and pseudo pin assignment (track assignment) for delay and noise control and optimization. After interconnect synthesis, we have basically completed the interconnect de-

signs to meet both the performance and signal reliability (noised related) requirements. Section 2.6 describes the final step of multi-layer gridless routing, which is needed to support interconnect layout of optimized interconnects with possibly complex geometries.

## 2 MAJOR STEPS/COMPONENTS IN THE FLOW

This section describes the major steps/components in the interconnect-centric design flow shown in Figure 3. A more detailed description of these modules are available from [1].

### 2.1 Interconnect Optimization

*Interconnect optimization* determines the optimal interconnect structure of each net in terms of interconnect topology, wire width and spacing, buffer locations and sizes, etc., to meet the performance and signal reliability requirements. Interconnect optimization is a key building block in the interconnect-centric design flow. Our group started a systematic study of the interconnect optimization problems since 1991 and has developed a number of efficient optimal or near-optimal algorithms for various interconnect optimization problems, including;

- interconnect topology optimization
- wire sizing optimization
- global interconnect sizing and spacing
- simultaneous driver, buffer, and interconnect sizing
- simultaneous interconnect topology construction with buffer insertion and/or wire sizing

and other possible combinations of these optimization techniques. These interconnect optimization algorithms have been integrated in the UCLA TRIO package (Tree, Repeater, and Interconnect Optimization) [2] for solving various interconnect optimization problems, and TRIO is used as a supporting block in our interconnect-centric design flow. Experimental results show that the interconnect optimization techniques in TRIO can lead to up to a factor of  $5\times$  delay reduction of global interconnects. Currently, the TRIO package is being extended for optimizing multiple physically related and temporal-related interconnect structures for both delay and noise optimization in nanometer designs. The TRIO package is available for download from the world-wide web [2].

### 2.2 Interconnect Performance Estimation

Given the fact that interconnect optimization may lead to significant global interconnect delay reduction, it is important to fully consider the impact of interconnect optimization during design planning. However, a brute-force integration by running existing interconnect optimization

algorithms directly during the synthesis and design planning stages will not be practical for the following reasons:

- Inefficiency: Although most of the interconnect optimization algorithms discussed in the preceding section have polynomial time complexity and are efficient to use during layout synthesis (For example, TRIO can optimize roughly 1 to 100 nets per second depending on the optimization algorithm being used), they are not efficient enough to be used *repeatedly* during interconnect planning where one would like to explore tens of thousands of floorplan configurations. For each configuration, the performance of tens of thousands of global and semi-global interconnects needs to be evaluated very quickly.
- Lack of abstraction: To make use of those optimization programs, a lot of detailed information is needed, such as the granularity of wire segmentation, number of wire widths and buffer sizes, etc. However, such information is usually not available during design planning.

Given these difficulties, existing design planning tools simply use wirelength based interconnect delay models and ignore the impact of various possible interconnect optimization operations. This may lead to very inaccurate results. To overcome this problem, we have developed a set of fast and accurate *interconnect performance estimation models* (IPEMs) with consideration of various optimization techniques, including optimal wire sizing (OWS), simultaneous driver and wire sizing (SDWS), and simultaneous buffer insertion, buffer sizing and wire sizing (BISWS) [3, 4]. These IPEMs are very efficient (constant run time in practice), and provide high-level abstraction. In addition, our IPEMs provide explicit relations between the interconnect performance and layout design parameters under various kinds of optimization; this helps to make design decisions at high levels. These models have been tested on a wide range of parameters and have about 90% accuracy on average compared with those running complex optimization algorithms in TRIO directly (in terms of the delay measured by HSPICE simulations). These performance estimation models have been implemented and integrated in The UCLA IPEM package. They form another supporting block of our interconnect-centric design flow, and are used extensively during interconnect planning. The UCLA IPEM package can be downloaded from the World Wide Web [5].

### 2.3 Interconnect Planning

Interconnect planning is the first step and also the centerpiece of our interconnect-centric design flow. It is applied very early on in the design process and has tremendous impact on the final result. We discuss interconnect

planning after interconnect optimization and interconnect performance estimation because interconnect planning makes use of various interconnect performance estimation models to consider the impact of interconnect optimization during the planning process.

We further divide the interconnect planning process into three steps: physical hierarchy generation, floorplanning/coarse placement with interconnect planning, and interconnect architecture planning. These are defined in the following paragraphs.

- (a) **Physical hierarchy generation:** Designs in the nanometer technologies are inevitably hierarchical given their high complexity. However, the HDL description provided by the architecture and/or circuit designers usually follows the *logical hierarchy* of the design which reflects the logic dependency and relationship of various functions and components in the design. Such logical hierarchy may not map well to a two-dimensional layout solution as it is usually conceived with little or no consideration of the layout information. This is further evident from the sub-optimal results produced by many existing hierarchical design tools which use the logic hierarchy for floorplanning and recursive synthesis, placement and routing. Their results can be considerably worse than those by (good) flat design tools (when the design complexity is still tractable). Figure 4 shows an example of the logic hierarchy in the final layout (obtained by optimizing directly on the flat design). Modules in the same block in the logic hierarchy have the same grey shading in the layout. As can be seen, the logic hierarchy does not map directly into the physical hierarchy. This suggests that enforcing floorplanning or placement algorithms to follow the logic hierarchy boundary can be harmful to the final layout. Therefore, the first step of our interconnect planning process is to generate a good *physical hierarchy* that is most suitable for being embedded on a two-dimensional silicon surface for performance optimization. Such physical hierarchy generation in fact defines the global, semi-global, and local interconnects (based on their levels in the physical hierarchy) and has significant impact on the final design quality. Recently, we have developed an efficient algorithm, for performance-driven partitioning and coarse placement with retiming, and use it as a possible approach to generating a good physical hierarchy [6]. Retiming is considered during partitioning and coarse placement so that flip-flops can be repositioned onto the global interconnects to hide (some) global interconnect latency. Experimental results show that combining partitioning, coarse placement, and retiming can provide an additional 23% delay reduction compared to the physical planning results obtained by separate performance-driven partitioning followed by floorplanning.

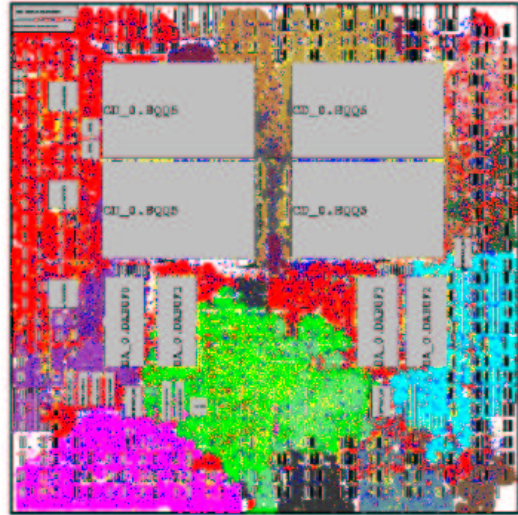


Fig. 4. An example of logic hierarchy in the final layout (Courtesy of IBM). It is a large ASIC design with over 600,000 placeable objects, designed using IBM's SA27E technology (a  $0.18\mu\text{m}$  technology with  $L_{eff} = 0.11\mu\text{m}$  and using copper wires).

- (b) **Floorplanning/coarse placement with interconnect planning:** After the physical hierarchy is generated, the second step is floorplanning with interconnect planning, which is also called *physical-level interconnect planning*. It interacts closely with the interconnect synthesis tools (to be presented in Section 2.5) and plans for the best interconnect topology, wire ordering, wire width and spacing, layer assignment, etc., for all global and semi-global interconnects to meet the required performance. For example, it is estimated that there will be a large number of buffers to be inserted for high-performance designs in future technology generations (close to 800,000 in 70nm technology [7]). If these buffers are distributed over the entire chip in an unstructured way, it will definitely complicate the layout design and verification. As a result, we have developed a method to automatically plan for buffer blocks during floorplan to achieve performance, area, and routability optimization [8]. Experimental results show that the proposed algorithm can reduce the number of buffer blocks by a factor of  $2.4\times$  with smaller chip area and a better chance of meeting timing constraints and smaller overall chip area.
- (c) **Interconnect architecture planning:** Due to the advance in VLSI fabrication technology, such as the use of chemical-mechanical polishing (CMP) for global and local planarization of insulator and metal levels, the design rules are no longer completely dictated by the manufacturing capability and leave large room for optimization. The goal of *interconnect architecture planning* is to take advantage of the degree of free-

dom in the process technology and determine various interconnect parameters for overall system-level performance, reliability and power optimization, subject to the manufacturing constraints. These parameters include the number of routing layers, the thickness of each interconnect and isolation layer, the metal resistivity and dielectric constant of each layer (assuming different material/process may be used for different layers for performance, yield, and cost considerations), the nominal width and spacing in each layer, vertical interconnection schemes (e.g., via dimensions and structures), and so on. Such interconnect architecture planning should consider a given design characterization (specified in terms of the target clock rate, interconnect distribution, depth of the logic network, etc.) obtained after physical hierarchy generation and floorplanning with interconnect planning. In some cases, such optimization requires adjustments in the fabrication process, which is more suitable and economical for high-volume designs (such as microprocessor designs) or a class of designs with similar design characterizations. One example of the studies that we made in this area is the investigation of the wire-width planning problem, where we showed that a small number of common wire widths in each layer can be used for optimizing interconnects of a wide range of lengths in that layer [4].

#### 2.4 Synthesis and Optimization Under Physical Hierarchy

After interconnect planning, the next of phase of the design flow is synthesis and placement for each module under the physical hierarchy, as shown in in Figure 3. This includes behavior-level (when applicable), RT-level, and logic-level synthesis in the presence of global interconnects defined from the physical hierarchy, followed by detailed placement. Currently, we are using off-the-shelf synthesis and placement techniques for this step (such as using the Design Compiler from Synopsys or the SIS/VIS package from UC Berkeley for logic synthesis and the TimberWolf or GordianL package for placement). We tend to believe once the physical hierarchy and global interconnects are defined, existing synthesis and placement algorithms can work well at the module level which contains mainly local interconnects, as argued in [9]. In particular, gain-based synthesis can be used to synthesize small to medium size logic blocks under the physical hierarchy [10, 11]. We are also starting a new project at UCLA on placement-driven synthesis to investigate if one can improve the result from this step significantly by combining synthesis and placement at the module level.

#### 2.5 Interconnect Synthesis

The third major component in our interconnect-centric design flow is interconnect synthesis. Given a logic synthesis and placement solution, interconnect synthesis determines the optimal or near-optimal interconnect topology, wire ordering, buffer locations and sizes, wire width and spacing, etc., to meet the performance and signal reliability requirements of all nets under the area and routability constraints. This is similar to the traditional global routing step, but with much emphasis on interconnect performance and signal reliability optimization. In our system, interconnect synthesis is achieved in two steps:

- (a) Multi-layer general-area global routing for delay and congestion optimization: It uses the multi-layer performance-driven global router developed in [12] as the basis and integrates various interconnect performance optimization techniques available in the TRIO package[2].
- (b) Wire ordering and spacing for noise and routability optimization: Given a set of routing tiles and a global routing solution associated with it, a *pseudo pin* of a net is a wire crossing point of the net at some tile boundary.<sup>1</sup> The *pseudo pin assignment problem* is to determine the locations of all pseudo pins of all the nets. Because pseudo pin assignment determines the wire ordering and spacing to a large extent, it can be used effectively for crosstalk noise control. Moreover, it provides an important bridge between global routing and detailed routing by specifying pseudo pin locations on the boundary of each tile. Otherwise, the detailed routing problem for each tile is not well defined. We used the PPA algorithm presented in [13] for pseudo pin assignment. Experimental results show that it provides good control of coupling noise without sacrificing routability.

The output of interconnect synthesis provides the topology, width, spacing, and ordering specifications of all the nets to the subsequent interconnect layout module for detailed routing.

#### 2.6 Interconnect Layout

The final step of our interconnect-centric design flow is interconnect layout. Aggressive interconnect synthesis and optimization often result in complex interconnect structures with many buffers, variable widths within the same net, or even variable widths within the same segment. Different spacing rules are also needed for crosstalk control and minimization. These requirements need to be supported by an efficient multi-layer gridless detailed routing system. In the past a few years, we have developed a novel gridless detailed routing system, named

<sup>1</sup>In contrast, the original pins in the design are called the “*real pins*” in order to be distinguished from the pseudo pins.

Dune, to support multi-layer, variable-width, variable-spacing routing. Dune has two major components: a point-to-point gridless routing engine and a route planning engine [14, 15]. The point-to-point gridless routing engine is based on path-searching in a non-uniform grid graph stored using implicit representation. Experimental results showed that the runtime of the gridless routing engine is 2-4 times faster than that of Iroute [16, 17], a well-known tile-based router for gridless routing. The route planning engine uses a coarse grid-based route planning algorithm. It uses a line-sweeping algorithm to find all routing obstacles in each grid cell and uses exact gridless design rules (variable width and variable spacing) to accurately estimate the available routing resources in each grid cell. It uses a multi-iteration planning method to overcome the net ordering problem, and evenly distributes the nets into routing regions. It plays a similar role as the conventional congestion-driven global router, but models the available routing resources more accurately and interacts with the underlying point-to-point gridless routing engine much more closely. Experimental results show that using the route planning algorithm in our gridless detailed routing can improve the routability and also speed up the run time significantly, by a factor of 3 to 17. A detailed description of our gridless detailed router can be found in [14, 15].

### 3 SUMMARY AND ONGOING WORK

With a good understanding of these building blocks, we can revisit the overall flow for interconnect-centric designs, as shown in Figure 3. Our flow can be summarized as follows: In order to cope with the design complexity of giga-scale integration in the nanometer technologies, we would like the designer (or the design team) focus on designs primarily at the architecture or conceptual level. Given a design specification (usually in a HDL specification such as Verilog or VHDL) as the output of the architecture or conceptual level design, our interconnect-centric flow first goes through the interconnect planning phase which transforms the functional hierarchy embedded in the HDL specification into a good physical hierarchy, performs coarse placement with global interconnect planning and interconnect architecture planning (when appropriate). It is possible for physical hierarchy generation to be performed together with coarse placement and global interconnect planning at the same time, as the global placement and interconnect planning usually influence the physical hierarchy generation. Interconnect performance estimation models are used extensively during interconnect planning for predicting the performance of the optimized interconnects. After physical hierarchy generation, coarse placement with global interconnect planning, we shall have a good first-order estimation of the overall circuit performance (which is determined primarily by global interconnects). We can quickly provide feedback

to the designer to indicate if the proposed architectural or conceptual level design is feasible. Therefore, the designer can quickly iterate with the interconnect planning tool to evaluate multiple architecture or micro-architecture designs, and converge to the most promising one(s) for further refinement. The end result of interconnect planning is a feasible "physical prototype" of the design.

After interconnect planning, the next phase of the design flow is synthesis and placement for each module under the physical hierarchy, including behavior-level, RT-level, and logic-level synthesis followed by detailed placement with consideration of delays associated with the long global interconnect defined by the physical hierarchy. Once synthesis and placement for each module is determined, we perform interconnect synthesis, which includes performance-driven global routing with various interconnect optimizations for delay minimization followed by pseudo pin assignment with noise minimization. Finally, a gridless routing system is used to complete interconnect layout to implement various kinds of optimized interconnect structures. It includes a coarse grid based route planning engine and an efficient point-to-point gridless routing engine working on the implicit representation of the underlying non-uniform grid graph.

All these modules have been implemented, and they interact through a common hierarchical data model (HDM). The HDM provides a complete functional and physical representation of the design, including the structural view, the functional view, the physical view, and the timing view so that logic transformation, interconnect planning/optimization, or layout design can be carried out at every phase of the design process.

Each module in this design flow has been fully verified and has shown very promising results, as presented in various sections throughout this paper. We are in the process of performing integrated test of the overall flow and design methodology. We are integrating all the modules into the proposed interconnect-centric design flow and running several complete designs through such flow. Our test suite includes the PicoJava processor from Sun Microsystems and a few designs from IBM. (IBM has installed IDM, the IBM Data Model, at UCLA and we are developing an interface to it.) We hope to report complete experimental results in the near future. We believe that such an interconnect-centric design flow will effectively bridge the gap between high-level design abstraction and physical-level implementation, reduce or eliminate the uncertainty due to interconnects on system performance and reliability, and assure design convergence between synthesis and layout.

### ACKNOWLEDGMENTS

The author would like to thank the students and researchers (current and former) at the UCLA VLSI CAD Laboratory who have contributed to the development

of the interconnect-centric design flow presented in this paper. These include Chin-Chih Chang, Lei He, Kei-Yong Khoo, Cheng-Kok Koh, Tianming Kong, Hardy Leung, Hon-Ching Peter Li, Sungkyu Lim, Patrick Madden, Takumi Okamoto, David Z. Pan, Toshiyuki Shibuya, Chang Wu, and Xin Yuan. The author would also like to thank Tony Drumm from IBM for stimulating discussions regarding hierarchical designs and for providing the design plot shown in Figure 4. The author is very grateful to Min Xie for his assistance in preparing this manuscript.

- [17] J. Ousterhout, G. Hamachi, R. Mayo, W. Scott, and G. Taylor, "Magic: A VLSI layout system," in *Proc. 21st Design Automation Conference*, pp. 152–159, Jun 1984.

## REFERENCES

- [1] J. Cong, "An interconnect-centric design flow for nanometer technologies," *Proceedings of the IEEE*, vol. 89, no. 4, pp. 505–528, 2001.
- [2] J. Cong, L. He, C.-K. Koh, D. Z. Pan, and X. Yuan, *UCLA Tree-Repeater-Interconnect-Optimization Package (TRIO)*, 1999, Available at [http://cadlab.cs.ucla.edu/software\\_release/trio/htdocs/](http://cadlab.cs.ucla.edu/software_release/trio/htdocs/).
- [3] J. Cong and D. Z. Pan, "Interconnect delay estimation models for synthesis and design planning," in *Proc. Asia and South Pacific Design Automation Conf.*, pp. 97–100, Jan., 1999.
- [4] J. Cong and D. Z. Pan, "Interconnect estimation and planning for deep submicron designs," in *Proc. Design Automation Conf.*, pp. 507–510, June, 1999.
- [5] J. Cong, W. Long, and D. Z. Pan, *Performance Estimation Models for Optimized Interconnects (IPEM)*, 1999, Available at [http://cadlab.cs.ucla.edu/software\\_release/ipem/htdocs/](http://cadlab.cs.ucla.edu/software_release/ipem/htdocs/).
- [6] J. Cong and S. K. Lim, "Physical planning with retiming," in *Proc. Int. Conf. on Computer Aided Design*, pp. 2–7, Nov. 2000.
- [7] J. Cong, "Challenges and opportunities for design innovations in nanometer technologies," in *SRC Working Papers*, [http://www.src.org/prg\\_mgmt/frontier.dgw](http://www.src.org/prg_mgmt/frontier.dgw), Dec. 1997.
- [8] J. Cong, T. Kong, and D. Z. Pan, "Buffer block planning for interconnect-driven floorplanning," in *Proc. Int. Conf. on Computer Aided Design*, pp. 358–363, 1999.
- [9] D. Sylvester and K. Keutzer, "A global wiring paradigm for deep submicron design," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 19, pp. 242–252, February 2000.
- [10] I. E. Sutherland, R. F. Sproull, and D. Harris, *Logical Effort: Designing Fast CMOS Circuits*. Academic Press, Inc., 1999.
- [11] Magma Design Automation, [http://www.magma-da.com/c/@4sqgeyXVifzmg/Pages/Gain\\_based\\_overview.html](http://www.magma-da.com/c/@4sqgeyXVifzmg/Pages/Gain_based_overview.html).
- [12] J. Cong and P. H. Madden, "Performance driven multi-layer general area routing for PCB/MCM designs," in *Proc. Design Automation Conf.*, pp. 356–361, 1998.
- [13] C.-C. Chang and J. Cong, "Pseudo pin assignment with crosstalk noise control," in *Proc. Int. Symp. on Physical Design*, pp. 41–47, 2000.
- [14] J. Cong, J. Fang, and K. Khoo, "DUNE: A multi-layer grid-less routing system with wire planning," in *Proc. International Symposium on Physical Design*, pp. 12–18, Apr 2000.
- [15] J. Cong, J. Fang, and K. Khoo, "An implicit connection graph maze routing algorithm for ECO routing," in *Proc. ACM/IEEE International Conference on Computer Aided Design*, pp. 163–167, Nov 1999.
- [16] M. Arnold and W. Scott, "An interactive maze router with hints," in *Proc. 25th Design Automation Conference*, pp. 672–676, Jun 1988.