Silicon Virtual Prototyping: The New Cockpit for Nanometer Chip Design

Wei-Jin Dai, Dennis Huang, Chin-Chih Chang, Michel Courtoy
Cadence Design Systems, Inc.

Abstract – A design methodology for the implementation of multi-million gate system-on-chip designs is described. The new methodology is based on the creation of a silicon virtual prototype early in the back-end design process. The prototype is generated in a fraction of the time required to complete the traditional back-end flow but still maintains very high correlation with the final design. The physical prototype becomes the 'cockpit' where many design implementation decisions can be optimized by leveraging the short iteration times. Hierarchical design methodologies benefit from the prototyping stage by enabling a more optimal partitioning.
The silicon virtual prototype also alters the nature of the hand-off model between front-end and back-end designers. The netlist can now be quickly validated using the prototype: the physical reality is being injected early in the design process resulting in fewer iterations between front-end and back-end.

I. Introduction

The successful design and implementation of nanometer generations of high-performance, multi-million gates, nanometer system-on-chip (SoC) integrated circuits (ICs) requires linking the front-end logic design and back-end physical design in a more collaborative manner. However, current EDA point tools are typically geared for use exclusively by one group or the other. Front-end logic designers resist engaging in physical design issues. Back-end physical designers in an ASIC flow rarely have sufficient design perspective to make high-level micro-architectural changes. The methodology relies on an iterative process between the front-end and the back-end. Moreover, in an interconnect-dominated design, simply iterating between existing tools can no longer ensure that the design process will converge to a physical implementation that meets both schedule and performance constraints. Performance constraints include timing, area, signal integrity and power requirements. Schedule constraints mean the time schedules demanded by the competitive time-to-market pressures. As a result, designers either resort to overly pessimistic designs (with wider than necessary guard banding), or suffer from the pain of many iterations between synthesis and layout. Each back-end iteration currently takes several days in order to discover if the chip can meet its constraints. Therefore, it is crucial to establish a more efficient and faster way to check design feasibility and allow trial-and-error analysis to a successful physical implementation.

On the other hand, given the size and complexity of today's chips, it is virtually inconceivable to flat-place and route all the layout objects. Thus a hierarchical design method has to be applied to manage the design complexity while preserving optimization quality comparable with the flat design approach.

II. Traditional Floorplanning

The problem with traditional floorplanning is its lack of accurate physical information. One approach to address this drawback is a faster physical implementation for better prediction of the performance given the area, timing, routing resource constraints. Based on the fast physical implementation, the circuit can be partitioned more effectively into a set of macro blocks, each of which is either a set of standard cells or an IP block (design reuse). A new approach to floorplanning and physical implementation, called alternatively *silicon virtual prototyping* or *physical prototyping,* has been gaining significant momentum [1][2]. The silicon virtual prototype (SVP) is a fast implementation of the physical design to evaluate design trade-offs and creates a realistic implementation plan.

Given a circuit netlist, traditional floorplanning tries to determine the chips' block location, shape, orientation, pin assignment, and timing budget, plus the power routing, and clock tree. The objectives are to meet the chip area, timing, power, and routing resource constraints. All of these problems have been studied for more than a decade (see [3] for detailed discussion), but the results still leave room for improvement due to the huge search space. Many approaches have been reported for floorplanning in the last few decades [4].

III. Continuous Convergence Methodology

A key benefit of a design methodology based on the silicon virtual prototype is that it provides continuous convergence. This begins with an initial full-chip design representation with wires, that treats all aspects of the design—logic, timing, SI, power drop, electromigration (EM), I/O issues, and manufacturability—concurrently [5]. Design teams use the SVP to identify and prioritize performance and manufacturing issues, and then individual designers work on the highest priorities. When ready, the team integrates all design changes into the SVP and re-analyzes the entire design. Design teams that use continuous convergence often standardize on a one-day turnaround—in essence, performing a virtual tape-out every day. Thus, every day they see predictable, measurable, systematic progress toward their goal of silicon closure and final tape-out.

The full-chip SVP is key to the continuous convergence methodology (Fig. 1). An SVP must be a complete full-chip implementation that is close enough to tape-out quality where designers can accurately assess all relevant aspects of the design. Yet, it must execute fast enough that designers can iterate rapidly in order to try different implementation directions. A prototype without detailed

wiring may help guide logic design, but it will not guide nanometer physical design adequately.

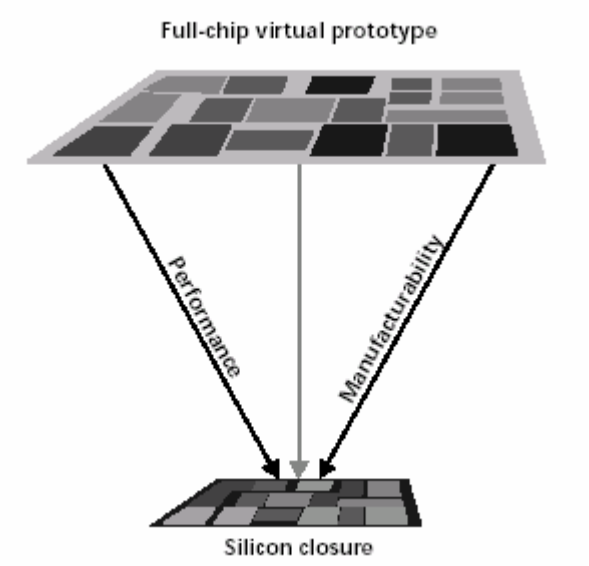

Fig. 1. Continuous convergence methodology.

The SVP must support clock structures, power grid, top-level interconnect, and other characteristics of the tape-out design. It must account for all relevant overhead in order to represent a known, physically feasible solution which can guide decisions such as timing-budget and pin assignments—a fully detailed layout with wiring is the only way to guarantee feasible budgets and assignments.

An SVP can serve as a universal cockpit for all tools and functions, combining all aspects of implementation and analysis within a single full-chip environment (Fig. 2). This environment can include implementation functions—floorplanning, placement, physical synthesis, routing, clock-tree synthesis, and power planning—and analysis functions—timing, signal integrity, routability, and power analysis.

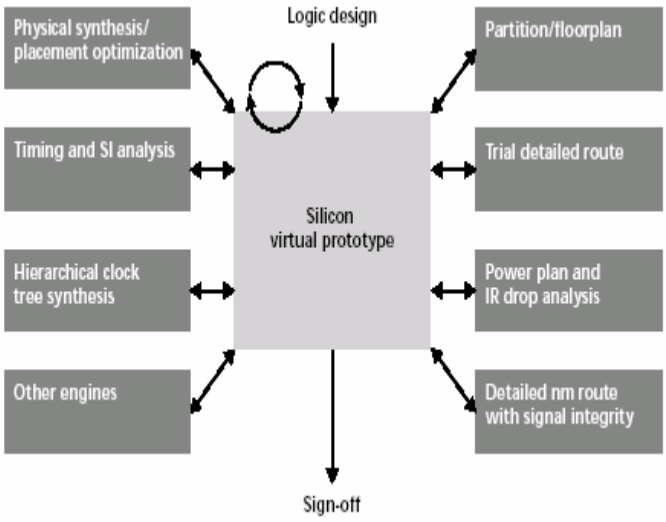

Fig. 2. An SVP can serve as a design cockpit.

IV. Correlation of Timing, Routability, Power and Signal Integrity Analysis

Correlation of the timing results (RC parasitics, delay and timing reports) between a silicon virtual prototype and tape-out-quality back-end tools is critical. The correlation can be verified by feeding a placement produced by a prototyping tool into a back-end flow and comparing the final timing information against the timing estimate at the prototyping. We have confirmed a tight correlation between the two for First Encounter. For a 200MHz design using a 180 nm process, engineers at Infineon concluded that the timing difference is only about 50-picosecond between the First Encounter prototyping tool and Synopsys' PrimeTime with Star-RCXT data [6]. Further analysis confirmed that the critical paths identified by the two are identical.

The physical prototype's timing calculations are based on the placement and trial routing produced by its fast engines. The trial route data approximate detail routing very closely. Therefore the extracted RC parasitics are also closely correlated.

Well-correlated timing relies heavily on three aspects: RC extraction, delay calculation and static timing analysis. A calibration procedure is required to fine-tune the RC extraction and the delay calculation. Actually, a simple constant scaling approach is very effective to centralize the error distribution [7]. Centralization of each delay stage's error distribution plays an important role to reduce the total path delay error, because the delay stages in a timing path may compensate each other. In addition, the speed can be dramatically improved by adaptively choosing different algorithms depending on the importance of each individual delay stage in the entire timing path.

Correlation between the prototype and the final tape-out version of the chip is required for power and signal integrity analysis as well. The prototype is used for power network planning, and noise prevention and repair. The accurate power analysis performed at the prototyping stage eliminates the need of the over-design of power and ground networks, which is practiced commonly in the traditional design flow without prototyping.

V. A New Approach To The Nanometer Design Flow

Silicon Virtual Prototyping can be used early in a design cycle to guide the implementation and partitioning of a whole chip. The proposed new design flow is shown in fig. 3.

A. Quick Logic Synthesis

The first step in this design methodology is to perform a quick logic synthesis of a netlist. The assumption at this stage is that the netlist is functionally clean, but that the timing is not necessarily accurate. Simple wire load models (WLM) can be used at this stage.

The gate-level netlist generated through this initial quick synthesis stage and the timing constraints form the inputs to the physical prototyping stage.

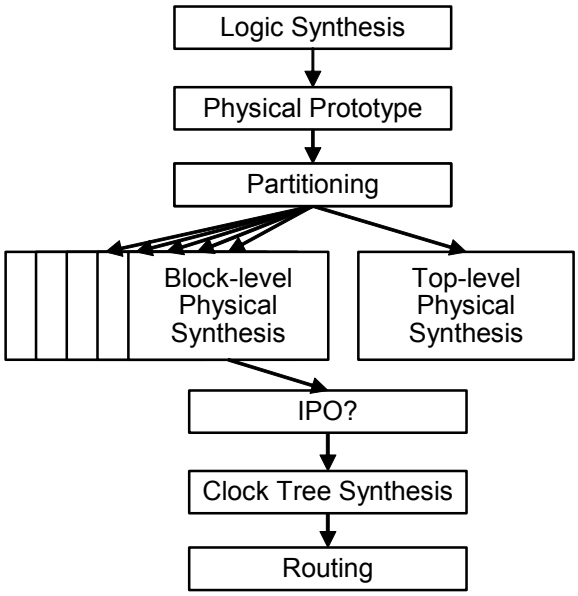

Fig. 3. Design flow with physical prototyping.

B. *Floorplanning*

The creation of the SVP starts with the floorplanning of the chip. This includes floorplanning tasks such as I/O placement, macro placement and power topology. Given the increasing importance of design reuse and the large number of macros being used, a combination of interactive and automatic floorplanning is required. The best results are achieved by a manual placement of the major design elements driven by the engineer's knowledge of the chip architecture, followed by an automatic placement of the remaining elements.

For the first pass at floorplanning, a netlist, physical libraries, corresponding synthesis libraries (.lib), top-level constraints, and a technology file (process description) are created and then imported. Interactive or constraint-driven placement of I/O pads is then used to meet the chip's specification. Placement guides or constraints are usually created for the major modules and used to 'guide' the placement engine where to roughly place the module cells.

C. *Quick Physical Implementation*

Next the remaining standard cells are placed using a timing-driven algorithm. The placement includes a routing stage ('Trial Route') that ensures the elimination of major congestion issues. The design is then RC extracted and timing analyzed, followed by In-Place Optimization (IPO) and clock tree synthesis. All of the complex physical effects are then analyzed and the performance objectives are verified. This prototype thus serves as a design basis.

The prototype is used for the creation of a netlist plus physical and timing design constraints for each block. The block-level constraints are used for design refinement of each block through logic or physical synthesis. The refined blocks can be assembled with other blocks to check timing and physical closure. This process is repeated until the designers achieve their design objectives. The results can be taken back through the backend design flow. Any IPO should be done within the prototyping tool to finalize timing closure.

Note that every cell is placed during the creation of a full-chip prototype, and a trial routing that closely approximates the final routing is also performed. This guarantees that the prototype is physically feasible.

D. *Hierarchical Design Flow*

The advantage of the hierarchical approach is that a design is decomposed into modules of manageable size, which are then refined in parallel by multiple teams. Controlling the block size is the key to the productivity of the design teams because back-end physical implementation tools such as detailed place-and-route and verification tend to be capacity-limited. Partitioning based on a flat full-chip physical prototype benefits from the optimization of a flat design while supporting a hierarchical design methodology.

This global perspective enables designers to experiment, using fast placement, with optimizations such as determining the most desirable aspect ratio for each block and the optimum location of the block relative to other blocks. Using fast routing and timing analysis, designers can also get very realistic pin assignments and timing budgets for each block before proceeding to the detailed design of the block. The prototype tools provide a high-capacity environment to build a flat chip model, from which the optimal physical hierarchy can be created. Fig. 4 shows the hierarchical design flow used in Cadence's First Encounter silicon virtual prototyping tool.

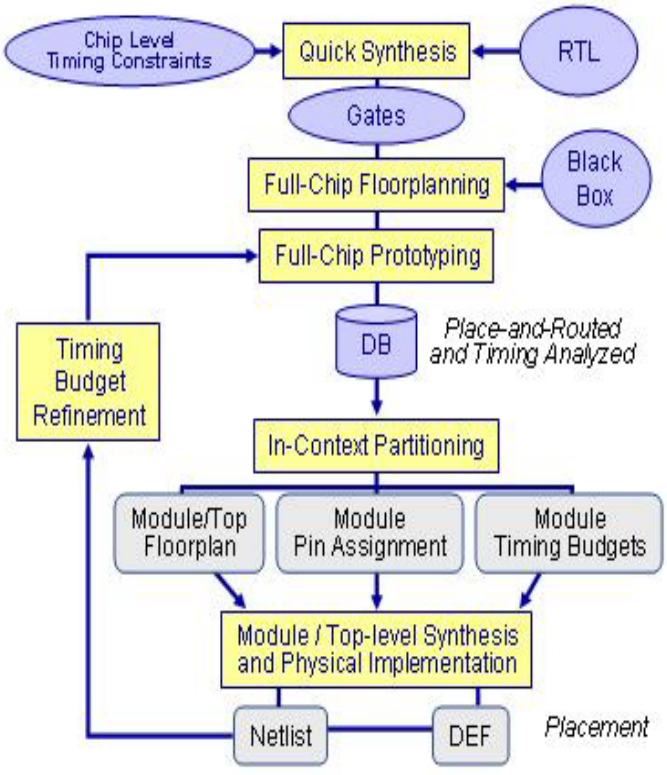

Fig. 4. First Encounter hierarchical design flow.

During the creation of the hierarchy, the aspect ratio, the pin assignment and the timing budgets are generated for each design module. Again, because they are based on the aforementioned physical prototype, the elements created are guaranteed to result in physical feasibility of the chip in the back-end. Optimal pin assignment is critical to reduce the complexity of the routing between the modules [8]. This directly translates to narrower channel width, which results in smaller die size. The allocation of the optimal timing budgets between the partitions is also critical to guarantee the timing closure of the chip. Without the full-chip prototype, the timing budgets are assigned arbitrarily to the modules. This leads to the situation where some modules will be impossible to implement while some other modules will have timing to spare. The result is an unnecessary iteration on under-budgeted modules, which delays the tape-out of the whole chip.

E. Timing Budget Generation and Refinement

The flow from the prototype to a final implementation needs to ensure predictable design convergence at all stages of the design flow. During the early phases of the design process, most of the elements of the design are still 'black boxes' or RTL code. At this early stage, the initial timing budgets created by the prototype are rough estimations. As more portions of the design get completed, a larger fraction of the design netlist becomes available in gate-level format. During this process, the full-chip physical prototype is built on a daily basis and the timing budgets are continuously refined based on the latest prototype. When the gate-level netlist is completed, final optimized timing budgets are generated. The key to this implementation process is that the most accurate view of the design is available at all stages through the quick generation of the prototype.

The faster turn-around time of the tool will help front-end engineers effectively develop chip-level timing constraints by taking into account physical design data. The prototyping tool will also generate hierarchical timing and physical constraints for each block automatically from the chip-level constraints. This is a labor-intensive and error-prone process if done manually. The prototype's fast turn-around time and strong correlation with the back-end implementation simplify the above two tasks.

VI. SVP Performance

Silicon virtual prototyping must show its value by improving turn-around-time. The prototyping system should provide an order-of-magnitude speed-up in comparison to existing physical design tools. The turn-around time has significant impact on designers' productivity and design schedules. The turn-around time from a netlist through placement, routing, RC extraction, delay calculation and timing analysis within a day for a multi-million-gate design is possible with today's leading-edge prototyping tools such as Cadence's First Encounter.

Table 1 shows data on three designs of increasing complexity from a networking company. The top section of the table shows the physical prototyping runtime using First Encounter physical prototyping tool running on a desktop workstation. All the times shown are the total times for the iterations actually required (4, 6 and 8 for design A, B and C respectively). This demonstrates how the quick turn-around time of prototyping makes it possible to optimize the design through multiple iterations without delaying a design schedule. Timing reports are shown before and after In-Place Optimization (IPO) indicating that the prototype is a key tool to achieve timing closure; all or most timing problems are solved at this stage.

The bottom section of the table summarizes the back-end implementation steps starting from the floorplan and placement produced by the prototyping tool. The tool set includes Synopsys Apollo router, Star-RCXT extractor and PrimeTime timing analyzer, and Mentor Graphics Calibre physical verification tool.

TABLE I
Performance data for the First Encounter prototyping tool

Design	Design A	Design B	Design C
Prototyping Tool			
Gate count	73K	330K	1142K
Components count	49K (+ 1 memory)	86K (+ 6 memories)	145K (+ 62 memories)
Floorplan (initial creation + manually adjustment)	30 min + 3 min	1 hr + 5 min	1.5 hr + 5 min
Placement	30 min (non timing driven)	1 hr (timing driven)	1.2 hr (timing driven)
Clock tree synthesis	20 min	30 min	40 min
Trial route	2 min	5 min	10 min
Extraction	1 min	2 min	3 min
Timing analysis	10 min	15 min	20 min
Timing report	-10 ns setup, 9 ns transition (clock 6.5 ns)	-5.5 ns setup, 10 ns transition (clock 6.7 ns)	-1.9 ns setup, 17 ns transition (clock 6.7 ns)
IPO	1 hr	1.5 hr	2 hrs ~ 30 min
Timing report	-2 ns setup, 3 ns transition	-1 ns setup, 3 ns transition	-1.2 ns setup, 3 ns transition
Number of iterations	4	6	8
Traditional Back-End			
Apollo routing + antenna (non timing driven)	2.5 hrs	4 hrs	5 hrs
Star-RCXT extraction	50 min	1.5 hr	1.8 hr
PrimeTime timing analysis	45 min	1.5 hr	2 hrs
Timing results (clock 7.5 ns)	0 violation, +0.67 ns setup, 2.3 ns transition (clock 7.5 ns)	no setup & hold, +0.21 ns setup, 2.7 ns transition (clock 7 ns)	no hold, -0.15 ns setup
Calibre DRC	1.5 hr	2 hrs	NA
Calibre antenna check	1 hr	1.5 hr	NA
Calibre LVS	1 hr	1.5 hr	NA

Comparison of the routing, extraction and timing analysis times between the prototyping environment and the traditional back-end implementation tools shows that the productivity gain with quick prototyping in the design cycle is significant.

VII. Summary and Conclusions

Detailed physical effects dominate performance and manufacturability in nanometer designs, making traditional linear flows obsolete. Wires are so important that performance analysis or optimization without detailed physical information is essentially meaningless. Successful nanometer physical IC design requires a continuous convergence methodology presented in this paper. Design teams that use continuous convergence see predictable, measurable, systematic progress toward their goal of silicon closure and final tapeout.

Unlike the traditional floorplanning approaches that are blind to the underlying complex nanometer physical effects, silicon virtual prototyping provides a playground where any design trade-off and constraints can be constantly monitored and verified. Moreover, this prototype can be refined using a detail implementation tool without losing consistency of design quality. Therefore, design closure can be achieved with high efficiency and predictability.

Physical prototyping tools also help to find better partitioning into manageable blocks, resulting in a hierarchical design methodology. Because the partitioning is based on the physically feasible prototype, the block-level timing budgets created are realistic and lead to a much easier task for traditional logic synthesis. The prototyping tool thus can become the hub of the design environment, covering partitioning, generation of block-level constraints, top-level design closure, clock-tree synthesis and power grid design.

References

[1] W.J. Dai, "Hierarchical design methodology for multi-million gates ASICs," DesignCon 2001.

[2] First Encounter User Manual V2002.1. Cadence Design Systems, Inc., March 2002.

[3] N. Sherwani, *Algorithms for VLSI Physical Design Automation, 3rd Ed*., Kluwer Academic Publishers, 1999.

[4] S.M. Sait and H. Youssef, *VLSI physical design automation, theory and practice*, IEEE Press, 1995.

[5] K. Bernstein, K. M. Carrig, C. M. Durham, P. R. Hansen, D. Hogenmiller, *High Speed CMOS Design Styles*, Kluwer Academic Publishers, 1998.

[6] V. Gerousis and W.J. Dai, "Frond-end physical design solutions provides hierarchical methodology", Design Automation and Test in Europe Conference, 2002.

[7] M. Selzer, B. Birkl and F. Lenke, "Silicon virtual prototyping using First Encounter", Evaluation Report, Motorola, Feb. 2002.

[8] J. Cong, "Pin assignment with global routing for general cell design", *IEEE Trans. Computer-Aided Design*, vol. 28, no. 20, pp. 1882-1884, 1992.