 SRC Physical Design Top Ten Problems

Jeff Parkhurst  Naveed Sherwani  Sury Maturi  Dana Ahrams  Eli Chiprout
Intel Corporation  Intel Corporation  LSI Logic Corp.  National Semiconductor  IBM
Folsom, CA 95630  Hillsboro, OR 97124  Milpitas, CA 95035  Santa Clara, CA 95051  Austin, TX 78660
Jeff.Parkhurst@intel.com  Naveed.Sherwani@intel.com  smaturi@lsil.com  dana.ahrens@ns.com  chiprout@us.ibm.com

I. Abstract

Transistor density on an integrated circuit continues to grow at an exponential rate. The associated complexities when applied to synthesis, layout, and timing analysis will become untenable within the framework of our current technological capabilities. This paper briefly outlines the problems facing the semiconductor industry early into the next decade. A SRC research thrust team consisting of members from various companies prioritized these problems and established a top ten list. This paper includes both problem descriptions as well as metrics to benchmark proposed solutions.

II. Introduction

The 1997 NTRS[1] states that integrated circuits will contain 40 million transistors by the year 2006 operating at frequencies near 4GHz. The challenges associated with the layout and testing of these designs are numerous at all aspects of the design flow. This will also impact the various elements of the design including clocking, power delivery, interconnect modeling and synthesis, noise avoidance, floorplanning, placement, as well as timing verification. VLSI tools of the future will likely be highly integrated from Synthesis to Layout to Timing Analysis and Verification increasing productivity to keep pace with design complexity. Certain underlying breakthrough technologies supporting this must be discovered in order for the exponential trend in integrated circuits to continue. This paper lists the areas of research critical to the semi-conductor industry where breakthrough technologies need to occur.

This list is derived from prioritizing the many research needs in the Physical Design area. The list consists of medium term research in the sense that it enumerates those technologies critical to the industry in the next 5-7 years. It is not meant to address tool requirements per se, but their fundamental underlying technologies. The authors are also responsible for generating the original list[2] which may be found on the SRC web site.

This list is follow on work to the original list[3] presented by Naveed Sherwani at the 1997 International Symposium of Physical Design. While this previous work was put together by a consortium of professors and industry leaders, the current focus of the SRC research thrust team is to provide a list primarily from an industrial perspective. The purpose is to capture what the industry rates as important problems rather than what is doable or exciting from a research perspective. The design drivers for these problems vary from IP blocks in SLI chips to high performance microprocessors and large ASICs. The details of each of these problems are outlined next and are not in any prioritized order.

III. Top Ten Problems

Block Level Placement

A significant amount of time in the design of an integrated circuit is spent iterating between placement and timing analysis/verification to meet timing requirements. Considering those parameters that affect timing, noise, and area during the placement activity can minimize this iterative process. Block level placement addresses the 40 million transistor placement problem via a hierarchical approach. This alleviates the need to handle the placement of an entire sea of transistors, simultaneously.

To address this productivity issue, a placement algorithm needs to be developed that handles a variety of weighted constraints such as area, timing, and coupling (or shielding). It should consider anticipated placement of repeaters where applicable and use a hierarchical design [Giga block > Mega block > block > cell] to reduce the number of components being placed at one time. One of the unique problems in this environment is to traverse the hierarchy to better optimize the resulting layout. This includes an initial estimation of the area and delay characteristics of a block, traversing down into the block for place and route at the next lower level, and then comparing initial estimates to the actual layout and iterating to an optimized solution.

The optimization priority is delay followed by area minimization. In general, timing issues due to coupling may be approached by minimizing those effects including maximal spacing within the given area constraints. Tighter area bounds could be achieved by considering the switching nature of the net and its impact on timing delay. Optionally, noise and power should also be considered to optimize within the delay and routing constraints for an individual net.

The scenario to test the algorithm would involve a Giga block with particular aspect ratio requirements and would contain a hierarchy with functional blocks and cells that reside inside it. The netlist and timing constraint data at the inputs and outputs of the Giga block would also be given. These then would be propagated down the hierarchy to verify internal (intermediate) timing requirements are also met. Additionally, noise constraint data on a net by net basis, the switching nature including activity factor and/or time window, and the general coupling attributes of adjacent nets for the different metal layers are optional but recommended to be considered. Finally, in addition to the timing constraints, there should be a set of given amounts of maximum cross coupling allowed between nets that are of the same timing class. A separate noise analysis algorithm will estimate these allowed cross-couplings and then the layout algorithm would need to respect these constraints.

The proposed algorithm will suggest routing of nets, placement of blocks to meet overall Giga block area and shape requirements, and preferred placement of repeaters for purposes of meeting timing and noise criteria. Scalability of the algorithm should be documented as well.
Any existing microprocessor or large ASIC chip may be used as a test case to benchmark the algorithm. The design should contain greater than 25 Mega blocks and greater than 25 blocks per Mega block. Number of cells within a block should be greater than 10,000. Benchmarked results could include various weighting of cost functions to observe sensitivity to the different objective function components.

**SOI-specific PD which would yield more than Remapping**

Silicon on Insulator technology provides its own unique set of benefits as well as problems over bulk CMOS. The unique problems resulting from floating body effects (FBEs) as well as increased coupling at the lower metal layer interconnects do not get properly addressed by simply design rule remapping from Bulk CMOS to SOI. This necessitates using unique and diverse design styles and layout styles for different types of circuits to both take advantage of SOI benefits and avoid pitfalls due to FBEs.

A PD algorithm and/or flow that would leverage off of the unique issues pertaining to the performance benefits found with SOI CMOS is needed to address this issue. It should incorporate design rules as well as avoiding particular design styles that lend themselves to functional deficiencies during operation. It should optimize area, timing, coupling, and power.

The proposed algorithm or flow could be tested on any existing ASIC design with greater than 1,000 devices where the design is specified in either RTL or as a gate level netlist. The coupling characteristics of lower level metals, operation of the different portions of the design, including functionality as well as frequency is supplied along with area, shape and power goals.

The resultant design and layout should meet timing and area goals as well as avoid deleterious effects on the operation of the circuit due to floating body effects, such as hysteretic behavior, frequency-dependent delay time, pulse stretching, etc. Functionality should be tested to verify these conditions are met.

**Interconnect Driven Floorplanning**

Interconnect is anticipated to dominate overall design delays (Cu provides only a temporary relief to this trend) as processes move into very deep submicron processes. Hence, interconnect must be taken into account at every stage of the design flow; the earlier the stage, the more effective the results. It is now necessary to account for interconnect in micro-architecture decisions, to avoid wire dominance that would make the design of a part either unable to meet its timing goals or unable to meet its power or die size goals.

A Floorplanning algorithm that treats interconnect as an integral part of the Floorplanning stage would address this issue. All interconnect must therefore be planned in three dimensions, as logic blocks are planned in two dimensions (with appropriate layer assignment). The floorplanner should comprehend area, shape, alignment and proximity constraints for the blocks and timing, coupling, noise and shielding constraints for the interconnect and should also consider repeater insertion for meeting the timing goals for long interconnect.

A set of blocks with area, allowable aspect ratios, locations (if necessary), constraints between blocks (alignment, abutment, proximity), and connectivity between blocks (with timing constraints) should be provided to test the floorplanner. The overall chip area and possible aspect ratios are also required as well as an objective function specifying the mix of area, timing trade-off. Any existing microprocessor or large ASIC chip could be used as a test case. The design should contain 300 to 400 blocks and 500-1000 busses with small individual or non-critical signals ignored. Objective function weighting and constraints such as overall area could be changed to observe and report convergence properties of the algorithm(s).

The resultant output should consist of a feasible floorplan with all blocks shaped, sized, and meeting as many constraints as possible. Interconnect should be routed in the appropriate layers along with proper placement of any required repeaters.

**High Speed Clock Net Design with Power and Skew Constraints**

The clock signal is used to define the time reference for the movement of data within a synchronous system. Much attention has been given to the characteristics of these vital clock signals and the networks used in their distribution. Clock signals are often regarded as simple control signals; however, these signals have some very special characteristics and attributes. Clock signals are typically loaded with the greatest fanout, travel over the greatest distances, and operate at the highest speeds of any signal, either control or data, within the entire system. Since the data signals are provided with a temporal reference by the clock signal, the clock waveforms must be particularly clean and sharp. Furthermore, these clock signals are strongly affected by technology scaling in that long global interconnect lines become highly resistive as line dimensions are decreased. This increased line resistance is one of the primary reasons for the increasing significance of clock distribution networks on synchronous performance. The control of any differences in the delay of the clock signals can also severely limit the maximum performance of the entire system and create catastrophic race conditions in which an incorrect data signal may latch within a register.

A clock net designer providing a reliable clock signal to all the required clock terminals within a required skew and power budget while considering process variation is needed to address this problem.

The algorithm would be provided with clock entry driver (drivers), clock terminals (with location, load and edge rate information), layout information (layers available, obstructions present, pre-routed signals), number of buffering levels required, possible buffer location information, and a specification of the power budget. This power budget excludes latches or flops being driven by the clock net. The clock net designer should produce a properly sized, buffered clock net meeting the skew, edge rate and load constraints at all clock terminals, routed in allowed layers, and placement of buffers in allowed areas.

The test case for the clock net designer could be any large ASIC chip requiring 2000 clock endpoint drivers. The algorithm should provide a solution for clock frequencies greater than 1 GHz. The clock designer should consider the trade-offs between skew, area, edge-rate and power.
Floorplanner Shape Generator

Early Floorplanning activity is becoming critical as time to market reduces. Floorplanners need accurate shapes and areas for the blocks and shape generator tools may provide such data. Given the complexity in terms of differences in arrays, datapath, random logic, domino logic etc., it is critical to develop a single tool that can provide such a function. This also adds value in that it enables exploration and examination of alternative layout strategies such as block rotation at a very early stage of the design.

To address this need, an algorithm needs to be developed which understands the type of block, cell count, interconnect requirements, and process technology to generate area and possible shapes or aspect ratios for the block. It would be provided the type of block (or sub types, array, random control logic, domino, datapath etc.), cell count (in terms netlist of such estimated counts), and a netlist (or estimated netlist) and would provide area and shape alternatives for the block.

The test case for the tool could be any large block with single and/or mixed sub-block types. Testing on blocks with different mix sub-types checking co-relation with final solutions is suggested.

Layout Driven Synthesis

Currently, logic and layout are performed separately in order to make the sub-problems computationally feasible. The design flow consists of creating a netlist using logic synthesis tools followed by layout. However, with the increase in dominance of interconnect delay, it has become critical to take layout information into account while synthesizing logic for sizing reasons. Recent attempts to improve the wire load model and feed it back into an optimization loop are steps in the right direction but do not solve the problem for high performance which requires increased integration between synthesis and layout. Additionally, present synthesis systems are not incremental, making this optimization loop costly.

This problem may be addressed by developing an integrated logic and layout synthesis algorithm and/or flow for (random) control logic. RTL in VHDL or Verilog, with area, shape, power and timing constraints for the blocks with feedthrough and other global layout requirements supplied. The tool would then provide a tape-out quality place and route block meeting the area, power and timing constraints.

Any large ASIC test case, with frequency requirement in 1-3 GHz range and greater than 100K cells may be used to test the tool. The complete solution (with logic and layout synthesis) should be possible in a few hours.

Placement Techniques for Large Blocks

The NTRS roadmap (1997 edition) predicts that by year 2002, an estimated 20M usable transistors (table 1, page 12) will be available for auto layout. Some ASIC vendors are claiming such densities today. While traditional microprocessor design styles call for custom layout and "hand packing" for density, many new products using "ASIC" methodology are compelled by market dynamics to work with few resources in a very short period of time. These ASIC products require a semi-custom approach. Hierarchical approaches do not obviate the need for blocks to be as large as possible in order to reduce the Giga block placement problem size to manageable quantity. It is imperative that the placement problem for up to a million placeable components is solved so as to take advantage of the next generation technology capabilities as they become available and to pave the way for the larger designs planned for subsequent generations. This technology would be used in conjunction with Floorplanning solutions to provide a complete layout solution for next generation designs.

The development of new placement techniques for 1,000,000 placeable components flat addresses this problem. The data requirements for the netlist must be within the allowed memory capacity of the popularly used workstations. The run times must be acceptable as typical placement should not take more than 12 to 16 hours elapsed time; however, this is not a firm rule and trade-offs of execution time versus layout "quality " are encouraged. Placement must take into account timing constraints and allow for local optimization of timing, area, and power prioritizing in this order. The techniques should also accommodate manufacturability constraints.

The algorithm is supplied a netlist[cells and connectivity], timing constraints, coupling characteristics for the different metal layers [optional], and switching nature (activity factor and switching window) [optional] which should produce a suggested placement of devices and routing of nets, including any repeaters, to meet overall timing requirements.

Any ASIC flattened database with 1 million placeable standard cells or greater can be used as a test case both with and without optional parameters for optimization [noise and power]. Consider execution time versus ‘quality’ of solution. That is, given that timing is met in all cases, tradeoff of reduction of area vs. execution time. The tool should require no special hardware to run. Using a normalized area router, at least 80% utilization (average) must be achieved for the test cases. Using a normalized channel router, a routing ratio of no more than 0.2 (channel ht/row ht) must be achieved for the test cases.

RLC Routing

The effect of inductance is increasingly being felt on-chip due mainly to long wide interconnect and the increasing speed of operation. At the same time, coupling capacitance is also increasing among interconnect lines. All routing algorithms thus far have been based on either Elmore’s delay, moments or RC lines. Routing algorithms can no longer afford to be constrained by only the R and C of the line but also by L and coupling Cc as well. A quick RLC timing/noise estimator is necessary.

It is necessary to develop a routing methodology where the inductance (L) and coupling capacitance estimates of nets are taken into account during routing. This requires a quick timing and noise estimate of general RLC models. The router would be given a routing region with a netlist, obstacles, some routed nets, pin locations for remaining unrouted nets, allowable layers for routing per net, noise margins for inductive/capacitive coupling, delay constraints, the RC per unit length/width of the lines, and some quick inductance and Cc per geometry extractors. It should produce a routing of nets in the netlist with appropriate width and spacing in order to account for the effects of inductance as well as coupling capacitance and RC per square of the lines to meet delay constraints and avoid noise. L and Cc are not provided since they
are a function of the entire geometry and must be quickly generated on the fly and then refined. Consider only inductive constraints and line RC at first and then bring in coupling capacitance.

The router may be tested using a standard industrial design with approximately 1000 nets using extraction and simulation to verify results running at a frequency of 4GHz and above.

**Variable Accuracy Inductance Extraction**

The effects of on-chip inductance are becoming more pronounced due to long wide interconnect and the ever increasing speed of operation. Additionally, accurate inductance extraction requires full 3D analysis and is very time consuming.

For the purposes of Physical Design (i.e. routing, noise avoidance, etc.), there is a need for a variable accuracy inductance modeling technique that can range from an extremely quick but coarse estimation to a longer but more accurate result.

The algorithm should be given a three-dimensional set of routed wire shapes, the nets they belong to, the material of the wires and the surrounding dielectric. The algorithm should produce line inductance estimates for the nets and their neighborhood ranging from a very coarse and quick estimate to a detailed but time consuming result. This algorithm would be applied to a 5+ layer, 1000+ net and 5000+ shape layout. It could be tested using any standard industrial design applying the extraction algorithm in addition to the usual R and C extraction. The results and CPU times should be compared with an accurate inductance extractor. We recommend starting with only line inductance and only later considering coupling inductance.

**Coupling Aware Static Timing Enhancements**

Coupling capacitance is increasing among interconnect lines and can seriously degrade static timing estimates. The delay through a net can be reduced or increased drastically depending on the switching pattern of its neighbors. Signal delay errors can affect PD constraints. Currently, static timing analysis takes into account only the RC of the line and the delay through the devices and does not take into account the coupling capacitance of the nets.

This can be addressed by developing an algorithm that produces point-to-point-timing windows of lines. The algorithm would be given the variable switching static timing window of their neighbors and provided a set of nets with both the RC of the lines and coupling capacitance between adjacent lines as well as switching speeds and strengths of drivers and receivers. A design of 500+ nets with long coupling of at least two other nets on a subset of those nets with fanouts of 3 or more should be used as well as several buffers/repeaters along the long nets. In general, the algorithm could be tested on any standard industrial design with the above characteristics. A standard transistor-level static timing analysis should be performed followed by an enhanced run considering the coupling information about the nets. Comparison of full simulation to the non-enhanced static timing should be performed.

**IV. Conclusion**

This paper has discussed the top ten problems in physical design as prioritized by an SRC research thrust team. The paper outlines the issues, deliverables, as well as test cases that should be used to benchmark proposed solutions.

**Acknowledgements**

The authors would like to thank Shishpal Rawat for his technical input in reviewing this paper.

**References:**

