Research Needs in Computer-Aided Design: Logic and Physical Design and Analysis

Logic synthesis and physical design, once considered separate topics, have grown together with the increasing need for tools which reach upward toward design at the system level and at the same time require links to detailed physical and electrical circuit characteristics. This list is a summary of important areas of research in computer-aided design seen by SRC members, ranging from behavioral-level design and synthesis to capacitance and inductance analysis. Note that research in CAD for analog/mixed signal is particularly encouraged, in addition to digital and hybrid system-on-chip design and analysis tools. Not included here are research needs in test and in verification, which are addressed in separate needs documents, and needs in circuit and system design itself, rather than tools research; these needs are assessed in the science area page for Integrated Circuits and Systems Sciences. Earlier needs lists with additional details are found in the "Top Ten" lists in synthesis and physical design developed by SRC task forces.

System-Level Estimation and Partitioning

System-level design tools require estimates to rapidly select a small number of potentially optimum candidates from a possibly enormous set of possible solutions. Models should consider power consumption, performance, die area (manufacturing cost), package size and cost, noise, test cost and other requirements and constraints. Partitioning of functional tasks between multiple hardware and software components, considering trade-offs between requirements and constraints, must be made using reasonably accurate models. Standard methods are also needed to estimate the performance of microprocessors and DSP cores to enable automated core selection. Methods should be able to select (one or multiple) cores to best satisfy the requirements and constraints and to show the consequences of the selection. The trade-off characteristics are application dependent and cannot be done in the abstract. Research must find methods of capturing basic core characteristics.

Behavioral Synthesis and Incremental Floorplanning

The links between behavioral synthesis and floorplanning work two ways. On one side, to reason about delays in a reasonably accurate fashion, the (very) global nets need to be identified before behavioral synthesis. Having a reasonable floorplan is one of the approaches. On the other side, behavioral synthesis can affect the floorplan drastically by trading off time (schedule) against area. Area can be either the size of individual floorplanned items where a faster implementation needs more area, or the addition of additional units to schedule operations in parallel. The interaction of the two, the forward constraint generation and backward estimation propagation needs more attention. RTL level libraries mostly include only area and delay numbers. To combine floorplanning and behavioral synthesis the libraries need to be extended to including physical information. Suitable abstractions need to be developed.

Physical and Floorplan Synthesis

Considerations in this topic include initial placement tied into synthesis, getting initial estimates of interconnect loading, taking into account routability/congestion; global (top level) routing to get wire lengths; accounting for power so that local 'hot spots' are spread out, net sizing, etc. Models for initial exploration need to be fast, while final implementation needs to be more accurate. Post-place-and-route optimization for performance/area, wire sizing, routing estimation, and some timing tweaks are usually needed in order to meet 100% of the setup/hold requirements. Some effort is needed to understand how to modify the netlist (move cell, buffer/split net, optimize cell type, etc.) to get the best performance out of the circuit in the smallest amount of area possible. . For floorplan synthesis, automated aids could make recommendations on changes to an initial floorplan to improve performance/area/optimization/power results. Traditionally, synthesis has been performed while respecting the artificial logical hierarchy that the designer has chosen to implement. This has placed limitations on both the implementation of the logic, as well as the methodology used to estimate the timing of the circuit. Reexamination of requiring the logical hierarchy to match the physical hierarchy during place and route could be valuable.

Floorplanner Shape Generator

A design tool, which understands the type of block, cell count, interconnect requirements and process technology to generate, area of the block, possible shapes or aspect ratios for the block. As time to market is reducing, it is critical to get an early floorplan. Such a floorplanner needs accurate shapes and areas for the blocks. The shape generator tools provide such an input to the floorplanner. In the past, such a topic has been studied under area estimation; however, given the complexity in terms of differences in arrays, datapath, random logic, domino logic etc., it is critical to develop a tool which can provide such a function. Moreover, in early floorplanning it is critical to explore alternatives, so questions like "what happens if this block is turned sideways" must be examined. Tools should support predefined wiring plans and block aspect ratios, and could consider logic resynthesis/restructuring as anther degree of freedom.

Synthesis of Multilevel Memory Structures

A well designed memory hierarchy can drastically improve the performance and have a large effect on the placement and routing complexity. Behavioral synthesis algorithms need to understand these trade-offs and be able to explore a variety of memory hierarchies and partitions. Much of the area of modern chips (and therefore power) is being spend on memories. Better memory partitioning and accessing to reduce power consumption is one of the low-power opportunities with the highest pay-off.

Behavioral Synthesis for Low Power Architectures

Special architectures or architectural configurations are known to be good for low power applications. Power saving transformations at the behavioral level pay off quite a bit more than at the logic level. Combining the two in efficient algorithms that take advantage of low power architectures could lead to substantial savings. Clock gating and precomputation are the first things that can be investigated in the behavioral context. Better allocation to minimize average power or better scheduling to minimize peak power are other opportunities. Both dynamic and static trade-offs can be explored

Effective Library Design

A well designed library can significantly improve the quality (performance, power, area) of the synthesized designs. The key issues are what should be the functionality, number of power levels, how to balance the output slews, what topologies are desired, how they should be laid out, etc. Systematic studies are needed to understand these effects and their implications to the technology independent and technology dependent optimizations currently in use in synthesis. Library design with automatic placement and routing for high-performance datapath units is encouraged

High-Speed Clock Net Design with Power and Skew Constraints

Develop a clock net designer which can provide a reliable (jitter within budget) clock signal to all the required clock terminals with the required skew (with skew budget) and within the specified power requirements. The clock designer tool should take process variation into account. Research in this area should extend to dynamic logic, gated clocks, pre-charge circuitry, etc. Clock gating, sink parity, multiple clock domains, robustness under manufacturing variability, novel distribution topologies and methodologies, early estimation , interaction with test and test clock distribution are topics that might be considered.

Block-level Placement

Develop algorithms for performing block level placement. They must handle a variety of weighted constraints such as area, timing, and coupling (or shielding), and treat interconnect as an integral part of floorplanning and placement. Such algorithms must understand area, shape, alignment and proximity constraints for the blocks and timing, coupling, noise and shielding constraints for the interconnect, and consider repeater insertion. They should also consider anticipated placement of repeaters where applicable. By block level is implied 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 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 and place and routing at the next lower level, and then iterating to an optimized solution. Optimization priority is delay followed by area minimization. In general, timing issues due to coupling, noise and power should be considered.

Layout for Control of Manufacturing Variability

Systematically address the variations in subwavelength optical lithography, in CMP and other pattern-dependent process steps, and within the reticle, to assess the magnitude and correlations of such variability sources, and trend analysis into future ITRS technology nodes. Consider synthesis and layout techniques to mitigate impact of such systematic variability with minimum design turnaround time cost and quality impact, novel co-optimizations , new flow integrations (from library design through floorplanning and performance estimation through layout-stage extraction and timing/noise analysis)

Design Constraint Driven Place and Route for Mixed Signal Designs

In mixed signal designs there are many interactions between the physical layout and specific design constraints. The place and route engine must have access to and an understanding of these design constraints, which include substrate coupling, IR drop, and parasitic interaction. Design constraints should be managed so that for a given layout it is possible to know what constraints are met and not met, with the place and route engine able to observe and be guided by the design constraints.

Leakage Aware Synthesis

Logic family exploration and algorithms for optimal usage of multiple Vt devices which would allow timing/performance to be met while minimizing leakage.

Variable-Accuracy Inductance Extraction

The effect of inductance is increasingly felt on-chip, mainly due to wide long interconnect and increasing speed of operation. Accurate inductance extraction requires full 3D analysis and is very time consuming. For the purposes of PD (routing, noise avoidance, etc.) there is a need for a variable accuracy inductance modeling technique that can range from extremely quick but inaccurate to medium-slow CPU speed but accurate. Valuable would be a tool which, given a three dimensional set of routed wire shapes and their nets, the material of the wires, and the surrounding dielectric, could produce line inductance estimates for the nets and their neighborhood ranging from a very rough and quick estimate to a detailed but time consuming result. Extraction taking into account coupling between capacitances and inductances, and model order reduction that includes mutual inductance, are other parts of the extraction problem.

Placement and Routing Techniques for Large Blocks

Develop new placement techniques for up to 1,000,000 placeable components flat. The data requirements for the netlist must be within the allowed memory capacity of the popularly used workstations. The run times must be acceptable. Placement must take into account timing constraints and allow for local optimization of timing/area/power. Optimization priority should be timing, then area, then noise, then power. The techniques should also accommodate manufacturability constraints. Considerations should be ability to co-optimize for multiple concerns (routability, noise, power, timing), algorithmic techniques for, e.g., multi-threshold, multi-supply, and incremental large-scale placement and routing taking inductance into account.

Power Management, Distribution, and Isolation

Power distribution optimization, especially in the presence of isolation requirements; decoupling and isolation methodologies/techniques; co-optimization with signal distribution needs (e.g., shielding of clock and critical global nets, management of inductance); co-optimization with layout enhancements for management of manufacturing variability.