POSITION STATEMENT

A CONTEMPORARY PERSPECTIVE
ON
DESIGN AUTOMATION AND VLSI IN THE 80's

by

Jonathan Allen
Department of Electrical Engineering and Computer Science
and
Research Laboratory of Electronics
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139

The goal of design automation for VLSI is to provide tools which permit rapid, correct, and economical design of custom circuits subject to constraints of area, time, or power. This design process can be seen as the transformation of a given input specification (e.g. a formal representation of an algorithm) through a succession of representations (architecture, functional elements, gates, circuits, layout) until a mask specification is obtained which can then be interpreted by a fabrication process to yield correctly working physical circuits. The major problems faced, then, are the selection of these representations together with the means to create, modify and transform them. Hovering over these needs is the growing cloud of complexity, which demands organized systems design such that the resulting specification can be easily understood, changed, and extended along with other designs to form still larger circuits. One term that ties all of these requirements together is productivity, and IC design is currently facing all the problems encountered in software system design. Just as programmers produce very few debugged instructions per day, today's IC designers usually produce less than ten "debugged transistors" per day when creating new logic circuits.

While it may seem that the invocation of modularity, hierarchy, and regularity as cures for complexity is a tiresome litany, these principles are the only ones we have to deal with large systems. The use of standard cells, gate arrays, PLAs, register files, course grid layout frameworks, rectangle placement and routing, and other compiled structures indicates that these principles are at least tacitly accepted, if not fully understood and exploited. Architectures are specified in terms of a high-level block design with interconnect. Descending from this level of the hierarchy, we are faced with the design of the basic blocks. In general, the designer can use three kinds of blocks. Fixed cells (e.g. pads, or standard cell circuits) can be obtained from a library. Parameterizable cells (e.g. PLAs and register files) can be generated by programs from a logical specification quickly and accurately. Finally, new cells can be created directly using a layout language (possibly with graphics).

Each of these cells should be checked (at the *cell* level) for well-formedness at a number of different levels of representation. Each cell must permit derivation of all the needed representations via programmed transformations. Hence, at the artwork level, design rule checking is used to check acceptability of the geometric layout representation, but it must be done hierarchically, so that when cells are modularly interconnected, it is not necessary to re-check the constituent cells. The topological network of transistors is another useful representation, permitting node tests and logic simulation. The circuit representation permits detailed simulation and timing verification. Artwork, topology, and circuit representations are clearly intimately related, but each permits the designer to examine and modify different aspects of the cell. Of course, higher level (more abstract) representations at the gate, register, ALU, and functional element level provide the means for test generation and functional simulation. The important conclusion is that each cell has a number of different representations associated with it, and they each provide the means for establishing partial correctness. Obviously, it must be possible to transform between them reliably in order to insure that *each* such view is a facet of the *same* overall design.

Once we know that the individual cells are well-formed, placement and routing techniques are used to create the next higher level of the hierarchy, and this process will in general require some modification of the constituent cell boundaries to accommodate shared busses, required pitch, or other constraints. We are just now beginning to see how to do this in a way that preserves the well-formedness of the cells. It also seems clear that routing capability must be provided within the layout language, where it can naturally be used for this hierarchical construction. The use of these interconnect techniques yields a new set of cells which must in turn be checked for well-formedness. What emerges is a design process that is characterized by alternate analysis/synthesis operations. Cells are synthesized, analyzed for correctness, and then grouped together to synthesize larger cells, which in turn must be analyzed for correctness. Critical to this process is the means to represent the cells with well-defined modular interfaces, that provide for the evolving satisfaction of well-formedness.

In addition to the provision for hierarchical design, many other design choices must be faced, some of which can be briefly mentioned here.

1. A good design system should provide means to explore basic space/time architectural tradeoffs. A given functional specification can thus be explored along several performance biases.

2. Algorithms have both a competence ("what must be done") and a performance ("how it may be done") aspect. It is a fundamental problem to learn how to separate these so that performance options can be explored. This is particularly important for the detection of latent parallelism.

3. Designers must be sensitive to the time during the design process at which they must bind decisions. Experience in computing has tended to indicate that deferred binding provides essential flexibility (e.g. in virtual memory maps). For example, much useful design specification can be accomplished at the topological level without the need to bind the detailed geometrical layout.

4. There are many issues of representation that have large implications for computational performance of a design system. For example, boxes may be represented as line figures or by areas (using "fill" symbols) and the ease of algorithm specification (for, e.g. design rule checking) varies greatly between these cases.

5. New types of testing need to be explored. As wafer-scale circuits are introduced, the need to test and reconfigure large systems arises. This is such a large and important problem that new schemes for electrically monitoring circuit performance over the entire wafer, coupled with the means to alter interconnect patterns after wafer fabrication, will become essential.

6. The attack on complexity must be pursued both in terms of algorithm structures and by the use of special hardware. There is much opportunity to cut polynomial time algorithms down to logarithmic or linear time dependence (e.g. design rule checking), but special hardware for simulation and other tasks will be able to cut down run times by at least two orders of magnitude.

7. There is a tension between synthetic and analytic techniques, as we have mentioned above. While there is a tendency to instantiate correct designs by synthetic programs, the need for analytic checks will probably never disappear.

Using powerful computing facilities including interactive graphics, new and useful design systems will rapidly evolve in the next decade. It is important that these systems provide for easy interaction by the designer so as to influence the direction of optimization. The number of gates, speed, area, power, and extent of interconnect, all are factors that the designer should be able to manipulate. The representations that we have mentioned are key to the ability to manipulate these design parameters, but there is a question as to how they should be controlled. Perhaps designs can be successfully achieved by automatic compilation procedures, but a great deal of study of high-quality designs will be necessary before we can rely on these techniques. For at least the early years of the coming decade, design systems that interactively complement human design choice and expertise, while minimizing the cost of complexity and insuring the well-formedness of modular structures, will play a useful role in the emerging VLSI design process.