

# Chapter 6. Custom Integrated Circuits

## Academic and Research Staff

Professor Jonathan Allen, Professor John L. Wyatt, Jr., Professor Srinivas Devadas, Professor Jacob White

## Visiting Scientists and Research Affiliates

James H. Kukula<sup>1</sup>

## Graduate Students

Robert C. Armstrong, Donald G. Baltus, Cyrus S. Bamji, Curtis S. Chen, Steven J. Decker, Ibrahim M. Elfadel, Marina Frants, Juha M. Hakkarainen, Craig L. Keast, Songmin Kim, Kevin Lam, Steven B. Leeb, Jennifer A. Lloyd, Andrew Lumsdaine, Ignacio S. McQuirk, Keith S. Nabors, Joel Phillips, Khalid Rahmat, Mark W. Reichelt, Mark N. Seidel, Amelia H. Shen, Luis M. Silveira, David L. Standley, Ricardo Telichevesky, Christopher B. Umminger, Filip Van Aelten, Woodward Yang, Paul C. Yu

## Technical and Support Staff

Dorothy A. Fleischer, Susan E. Nelson

## 6.1 Custom Integrated Circuits

### Sponsors

Analog Devices, Inc.  
IBM Corporation

### Project Staff

Professor Jonathan Allen, Robert C. Armstrong, Donald G. Baltus, Cyrus S. Bamji, Mark W. Reichelt, Filip Van Aelten

The overall goal of VLSI CAD research is to provide the means to produce custom integrated circuits correctly, quickly, and economically. Traditionally, correctness is checked at several representational levels, such as layout (via design rule checking), and circuit and logic (both via simulation). The techniques for checking correctness are usually local to the particular representational level involved. While these techniques are important components of the design testing, they do not attempt to provide alignment and consistency checks between the different representational levels and an input behavioral specification. In addition, they do not characterize the set of possible designs at each representational level corresponding to the initial functional specification while ranging over a variety of performance levels. For this reason, there is an increasing need to provide CAD tools that can serve as the framework

for design exploration, providing the desired performance together with consistently aligned representations at all levels.

In this research group, we are studying a variety of research topics with an emphasis on performance-directed synthesis of custom VLSI designs. An overview of the viewpoint that motivates these projects has recently been provided in a major survey paper<sup>2</sup> in which the need for coordinating the design optimization process over the several levels of representation is emphasized. Since design exploration is so central to the production of high-performance designs, emphasis is placed on how performance can be characterized at the several levels of representation and how overall optimal performance can be achieved in an integrated way. In addition to the basic issues of circuit optimization, architectures for digital signal processing have been studied because of the highly parallel nature of the algorithms involved and the need for a very high level of real-time performance in these systems. We have increased our emphasis on developing the means for formally specifying designs that facilitate performance-directed design exploration as well as efficient verification of correctness of the coordinated set of design-level representations. All of these studies are taking on an increasingly theoretical approach, signifying the transition of digital system design from an art to a science.

<sup>1</sup> IBM Corporation.

<sup>2</sup> J. Allen, "Performance-Directed Synthesis of VLSI Systems," *Proc. IEEE* 78(2): 336-355 (1990).

In previous work, techniques were developed to automatically transfer a netlist of an MOS circuit (either NMOS or CMOS in any circuit style) to a highly efficient, compacted layout. Providing algorithmic control over this important transformation is essential in overall design optimization. This procedure has been used as a "back end" for technology mapping processes, and the corresponding program has been very successful in benchmark tests.<sup>3</sup>

Given this assurance of the generation of high-quality layout for circuit netlist specification, attention is now being turned to performance exploration at the architectural level, with a focus on digital signal processing applications which are highly parallel. In this research, architectures that repeatedly use a small set of circuit cells in a regular array are being emphasized, such as systolic arrays. While these arrays, which utilize nearest neighbor communication, have been successfully used for a wide variety of signal processing tasks, it is of interest to explore more relaxed models where communication is allowed to extend beyond nearest neighbors to a larger local area. This extension admits a wider class of architectures and invites a more comprehensive level of design exploration where architectural flexibility is traded off against more complex, local delay models. Formal techniques for specifying these signal processing tasks at a behavioral level have been introduced, and means for mapping the algorithm onto multiple index spaces have been devised, together with the means to explore these mappings. An affine delay model has been introduced to quantify communication delay, and an overall integer linear programming approach has been used to optimize the overall design on a timing performance basis. In addition, branch and bound techniques have been used to prune the possibly large search space to a smaller and more manageable set of alternatives that can be quickly assessed. In this way, global optimization is being directly addressed through the linking of architectural exploration and delay models in the context of a large but restricted design space. Since these circuit models are well characterized, the previously developed techniques for automatic conver-

sion between circuit netlist representation and layout provides an overall means for simultaneous optimization across all representational levels, from layout through circuit, to logic, and architecture. This investigation aims to provide one of the first instances of overall global optimization across all representational levels, avoiding premature local optimization at any single level of representation.

There is increasing interest in extending circuit design and optimization techniques to the device level by refining the device models, and also by simulating them directly rather than using approximate average device model parameters in a circuit simulator. While the device-level equations can be solved directly to provide the needed accuracy, their solution is generally found to be too time-consuming on conventional architectures. Accordingly, Reichelt has been studying techniques for the parallelization of transient two-dimensional simulation of MOS devices using waveform relaxation techniques.<sup>4</sup> Uniform convergence of the waveform relaxation algorithm in this application has been demonstrated, and speedups between five and eleven times have been found using these new algorithms. These techniques have recently been implemented on three-dimensional parallel architectures in a way that exploits the inherent parallelism of the waveform relaxation techniques.

In earlier work, Bamji demonstrated how context-free grammars could be used to formally represent large classes of circuit netlists, such as ratioed NMOS designs, static CMOS designs, and precharge/evaluate circuit designs.<sup>5</sup> He also represented the class of all layouts as a regular structure grammar and showed how to verify the correctness of the resulting layout using parsing techniques. This formalism has essentially provided a new technique for hierarchical design rule verification using a cell library through formal characterization of the interactions of cell templates.<sup>6</sup> Design rule verification is achieved by covering the layout with these templates, which are defined in terms of graphs, and all operations are performed in the graph domain. The verification procedure is incremental, and because the number of cell instances is usually much smaller than the number of mask geometries, it is much faster than

---

<sup>3</sup> D.G. Baltus and J. Allen, "SOLO: A Generator of Efficient Layouts from Optimized MOS Circuit Schematics," Proceedings of the 25th Design Automation Conference, June 1988.

<sup>4</sup> M. Reichelt, J. White, and J. Allen, "Waveform Relaxation for Transient Two-Dimensional Simulation of MOS Devices," *IEEE Trans. Comput-Aided Des.*, forthcoming.

<sup>5</sup> C.S. Bamji and J. Allen, "GRASP: A Grammar-based Schematic Parser," Proceedings of the 26th Design Automation Conference, June 1989.

<sup>6</sup> C.S. Bamji and J. Allen, "GLOVE: A Graphic-based Layout Verifier," forthcoming.

techniques that directly manipulate mask geometries. A current emphasis is on the automatic discovery of cells from nonhierarchical layout. Through these means we expect to derive design hierarchy automatically, and hence, suggest new cells for a library. The graph-based template techniques are similar to those previously used in a regular structure generator,<sup>7</sup> which has proved to be a convenient method for the elaboration of layout corresponding to higher level architectural connectivity.

In earlier studies, Van Aelten showed how to perform the verification of circuit properties based on a grammar-based schematic parser developed by Bamji. These formal techniques are now being extended to the verification of relations between synchronous machines. Since the machines may be implemented in many different ways, it is important to be able to transform representations of these machines into alternate forms that can be directly compared. Using a new set of primitive relations between string functions, an arbitrarily long sequence of behavioral transformations can be compressed into a single, composite relation that is guaranteed to correctly represent the given sequence of primitive transformations.<sup>8</sup> In this way, it is possible to use these composite transformations to verify the behavioral specification against the resultant logic implementation with one automata equivalence check, using any one of several available methods. In this way, a computationally expensive verification technique is reduced to a much simpler verification task.

We continue to focus on issues of design database and framework specification. It is increasingly recognized that a design database is the central component of an overall design system, along with a variety of specialized CAD algorithms that utilize one consistently maintained overall design representation. Nevertheless, maintenance of this database so that all levels of representational view are consistent with one another and are hence projections of one (and only one) underlying design is an exceedingly difficult problem. One single representational formalization has recently been introduced by Armstrong, and it is expected to support all levels of representation currently utilized or anticipated.<sup>9</sup> By setting up a series of

explicit correspondences between representational levels, consistency can readily be verified and maintained in the face of design modifications at any of the several different representational levels. This automatic consistency maintenance is highly desirable in any design system, and can be readily coupled to performance exploration strategies at levels where a change at a higher level of abstraction gives rise to many possible lower level representations, as in the transition from a circuit specification to layout. A preliminary version of this procedure has now been implemented, and is being tested over a substantial variety of cases.

With the advent of a new emphasis on central databases and their connections to specialized CAD algorithms, there is new concern for the study of integrated, cooperative CAD computing environments involving both workstations and mainframes. A proposal has been developed for the study of coordinated workstation, mainframe, and large disk system environments that are interconnected in a flexible way using fiberoptic switching. In this way, large, high-performance disk systems are flexibly coupled between both mainframe servers and high-performance workstations, and a means is provided to explore the optimal distribution of tasks over such a distributed framework. This environment also provides a structure in which distributed parallel algorithms can run in a coherent way within an overall environment that supports other specialized nonparallel CAD algorithms.

## 6.2 The MIT Vision Chip Project: Analog VLSI Systems for Fast Image Acquisition and Early Vision Processing

### Sponsors

National Science Foundation/Defense Advanced Research Projects Agency  
Grant MIP 88-14612

### Project Staff

Professor John L. Wyatt, Jr., Professor Berthold K.P. Horn, Professor Hae-Seung Lee, Professor

<sup>7</sup> C.S. Bamji, C.E. Hanck, and J. Allen, "A Design-by-Example Regular Structure Generator," Proceedings of the 22nd Design Automation Conference, June 1985, pp. 16-22.

<sup>8</sup> F. Van Aelten, J. Allen, and S. Devadas, "Verification of Relations between Synchronous Machines," 1991 MCNC Logic Synthesis Workshop, forthcoming.

<sup>9</sup> R.C. Armstrong, *A Formal Approach to Incremental Consistency Maintenance in Multirepresentation VLSI Databases*, Ph.D. diss. proposal, Dept. of Electr. Eng. and Comput. Sci., MIT, 1987.

Tomaso Poggio, Professor Charles G. Sodini, Professor Jacob White, Steven J. Decker, Ibrahim M. Elfadel, Juha M. Hakkarainen, Craig L. Keast, Ignacio S. McQuirk, Mark N. Seidel, David L. Standley, Christopher B. Umminger, Woodward Yang, Paul C. Yu

## 6.2.1 Introduction

In real-time machine vision, the sheer volume of image data to be acquired, managed and processed leads to communications bottlenecks between imagers, memory, and processors and also to very high computational demands. Our group is designing experimental analog/VLSI systems to overcome these problems. The work is presently concentrated entirely on early vision tasks, i.e., tasks that occur early in the signal flow path of a machine vision system. Designs of chips for certain tasks in velocity estimation, image moment calculations, depth from stereo, and edge detection are currently underway or have been completed.

This project began in September 1988, and the faculty involved are Professors Berthold K.P. Hae-Seung Lee, Tomaso Poggio, Charles G. Sodini, Jacob White, and John L. Wyatt, Jr., who is the principal investigator. This work was inspired by Professor Carver Mead's pioneering efforts at the California Institute of Technology, although our project methods and goals are different from Mead's.

The goal of this project is to design and build prototype early vision systems that are remarkably low-power, small, and fast. The typical system will perform one or more computation-intensive image-processing tasks at hundreds to thousands of frames per second using only tens to hundreds of milliwatts. The entire system with lens, imager, power supply and support circuitry could fit inside a cigar box.

In this project, we are exploring the various types of analog processing for early vision. There is no single design strategy, but each design has many of the following features:

- sensors and processing circuitry integrated on a single chip,
- parallel computation,
- analog circuits for high speed, small area and low power,
- selection of tasks and algorithms requiring low to moderate precision (roughly equivalent to

6-8 bit fixed point precision in a digital machine),

- special emphasis on computations that map naturally to physical processes in silicon, e.g., to relaxation processes or to resistive grids,
- emphasis on charge-domain processing, e.g., CCD and switched-capacitor implementations, for maximal layout density and compatibility with CCD sensors,
- sufficiently fast processing that no long-term storage circuitry is required, and
- careful matching of algorithms, architecture, circuitry and (often custom) fabrication for maximum performance.

The advantages of this analog design approach to early vision hardware are (1) high speed (both in the sense of high throughput and low latency), (2) low power, and (3) small size and weight. High throughput can be important in high speed inspection processes, e.g., for printed materials or PC boards. Low latency is very important for closed loop systems because delays are destabilizing. Relevant examples might include vehicle navigation and robot arm guidance. Low power together with small size and weight are important for airborne and space applications. And finally, small systems tend to be affordable.

These advantages are achieved at a cost. One problem is that a high degree of care and expertise is required for high performance analog design. Another is that performance and functionality of analog integrated circuits is critically dependent on exact fabrication process parameters. For these reasons, the first design of an analog chip, even by an experienced designer, often does not work, so that repeated efforts are required.

Another problem is that there is a large amount of labor and frequent delays involved in custom fabrication on campus. Finally, analog systems have little flexibility compared with their digital counterparts. Thus analog design, often coupled with custom fabrication, is only appropriate when extremely high performance is required.

Chips are fabricated through Mosis whenever possible. Chips requiring special processing are fabricated on campus at MIT's Microsystems Technology Laboratories where Craig Keast and Professor Charles Sodini have developed a special CMOS/CCD fabrication process for vision applications.

Sections 1.1.2 and 1.1.3 describe chips that have been designed and tested by doctoral students David Standley and Woodward Yang.

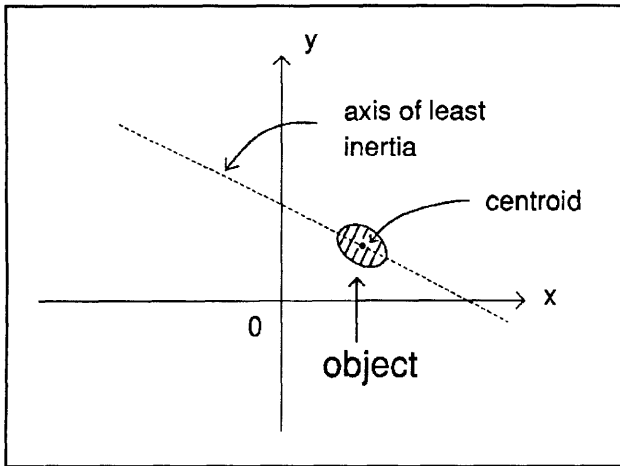


Figure 1. Example of object centroid and axis of least inertia.

### 6.2.2 Fast Imager and Processor Chip for Object Position and Orientation

This analog VLSI chip finds the position and orientation of an object's image against a dark background. The algorithm is based on finding the first and second moments of the object intensity. These moments allow the centroid (an indicator of position) and the axis of least inertia (an indicator of orientation) to be computed; see figure 1. If  $I(x,y)$  is the intensity as a function of position, and we (initially) assume that  $I(x,y) = 0$  outside the object, then the required quantities are given by

$$\iint I(x,y)h(x,y)dA$$

for all of the following  $h$ :

$$h(x,y) = 1, x, y, xy, x^2 - y^2$$

( $x^2$  and  $y^2$  are not needed separately). All of the weighting functions  $h$  are harmonic; i.e., the Laplacian vanishes identically:

$$\Delta h(x,y) \equiv 0.$$

This observation is a key to a scheme proposed by Horn, in which an analog computer based on a resistive sheet (or resistor grid) can be constructed, in principle. In the implementation described here, an  $N \times N$  array of discrete intensity data is reduced to a set of  $4N$  quantities by a 2-D resistor grid and is subsequently reduced to a set of just eight quantities by 1-D resistor lines, all in a continuous-time, asynchronous fashion – no clocking required. The eight outputs can be

digitized, and the centroid and orientation can be found using simple expressions. While resistive sheets have been used in earlier systems to compute position, none have been previously used to perform the orientation task, which requires computing second moments.

Figure 2 shows the resistor grid and its associated array of photoreceptor cells, which are uniformly spaced and occupy most of the chip area. The object image is focused onto the surface of the chip, inside the array. Each photoreceptor cell contains a phototransistor and processing circuitry; it converts the incident light intensity into a current that is injected into the grid. Thresholding is available to remove a dim (yet nonzero) scene background, so it does not interfere with the calculation. If the intensity  $I$  at a particular cell is below an adjustable threshold value  $I_{th}$ , then no current is injected. If  $I > I_{th}$ , then the output current, which is analogous to the gray-level weighting at that location, is proportional to  $I - I_{th}$ . The result is a continuous, piecewise-linear response characteristic. The array size is  $29 \times 29$ ; intentional image blurring over a few elements gives substantially increased resolution.

The perimeter of the grid is held at a constant voltage by the current buffers in figure 2; this ensures proper operation of the grid as a "data reduction" computer. The buffer outputs are simply copies of the currents flowing into them from the grid; the buffers are needed to isolate the grid from the effects of other circuitry. Figure 3, which shows the complete architecture of the chip, indicates how the (copied) grid currents are fed into resistor lines on the perimeter, how the ends of these lines are connected, and where the eight output currents exit the chip near the corners. These currents are measured by external circuitry (which also holds the ends of the lines at ground potential). In this setup, there are two lines on each side: one uniform and one quadratic line. These calculate weighted sums of the grid currents, where the weighting is (respectively) a linear or square-law function of the position along the line. The buffer outputs are steered either to the uniform or quadratic lines, so that four outputs are available at a time; i.e., multiplexing is required here (but is not necessary in general).

Working chips have been fabricated using the MOSIS service. The die size is 7.9 mm x 9.2 mm, and the imaging array is a 5.5 mm square. Accuracy is dependent on the object. For moderately sized and sufficiently elongated objects, e.g. a diamond of diagonal dimensions 25 by 50 on a (normalized) 100 by 100 image field, orientation is typically determined to within  $\pm 2^\circ$ . The speed is 5000 frames per second, and power consumption is typically 30 mW.

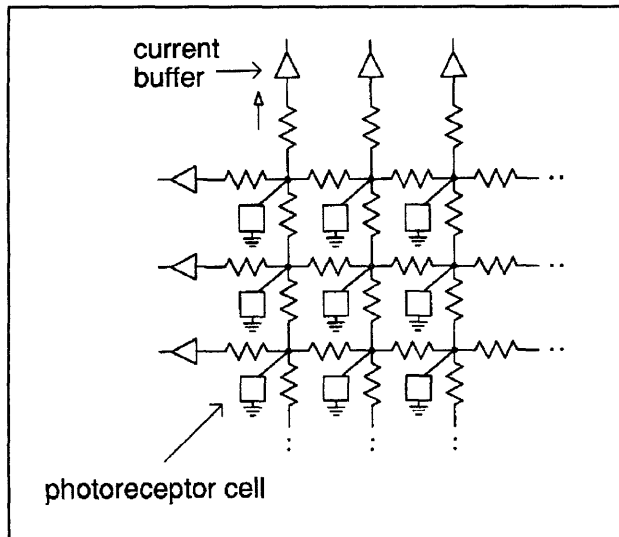


Figure 2. Resistor grid and photoreceptor cell array.

### 6.2.3 Integrated CCD Imager and Processors for Edge Detection

#### *Parallel, Pipelined Architecture*

Many image processing algorithms and machine vision tasks consist of calculations or operations performed on spatially localized neighborhoods. Therefore, the highly parallel nature of these algorithms can be effectively exploited if they are performed directly on the imaging device before the parallel structure of the data is disrupted by serial data output.

For a typical CCD imaging array, the parallel, pipelined architecture (see figure 4) provides a balance of computation speed, internal clock rate, internal storage, and I/O bandwidth without degrading the imager fill-factor (sensitivity). As columns of image data are clocked from left to right in parallel, local interactions between neighboring column elements are directly computed by the processor elements. As row values are sequentially clocked through the processor, local interactions between neighboring row elements are similarly performed by utilizing delay elements. This architecture is able to efficiently implement linear convolutions that are separable and recursively defined. This architecture is also suitable for certain types of nonlinear filtering operations that perform image segmentation, a basic machine vision task.

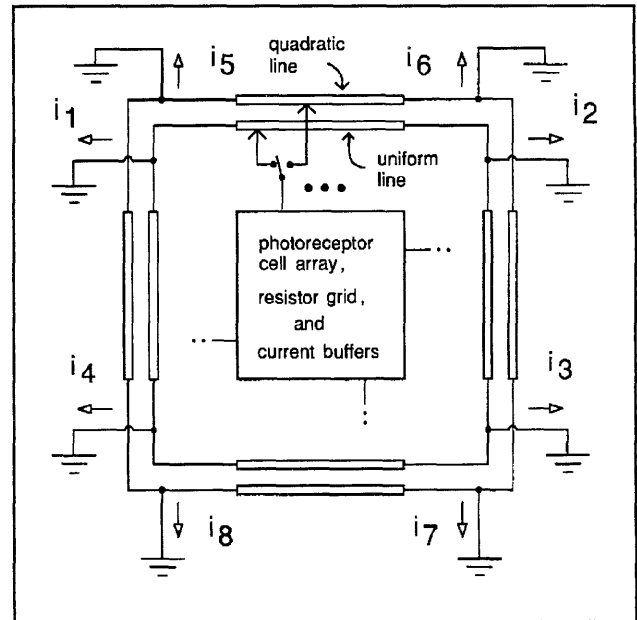


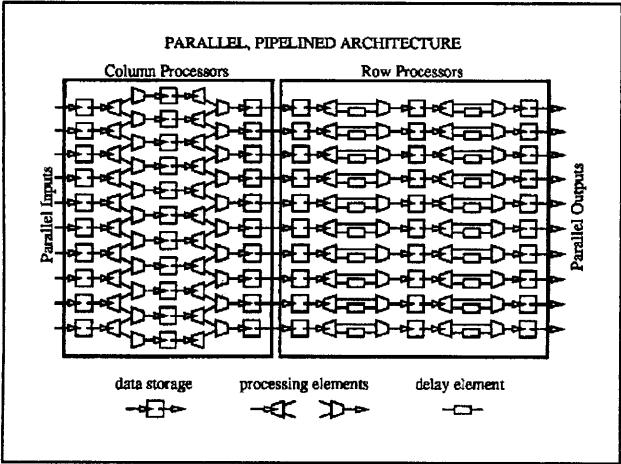
Figure 3. Main chip architecture.

#### *CCD Processors for Edge Detection*

A full fill-factor CCD (64 x 64) imager with an integrated, analog CCD signal processor for edge detection was implemented using standard 4  $\mu\text{m}$ , CCD/NMOS technology. By combining simple charge-domain, analog circuitry in a parallel, pipelined architecture, the prototype image processor was capable of performing a simple edge detection algorithm at 1000 frames/second. Furthermore, this signal processing capability was achieved with only a 15% increase in area, without any additional fabrication steps, decreasing the fill-factor of the CCD imager, or a significant increase in power dissipation.

While there are many edge detection algorithms, the LoG (Laplacian of Gaussian) of the image was chosen for its ease of implementation. The Laplacian is a rotationally symmetric, scalar operator. The zero-crossings of the Laplacian correspond to maxima in the image gradient which are interpreted as object edges. The Gaussian acts as a low-pass image filter that reduces the effects of noise in the image and has several desirable properties such as (1) rotational symmetry, (2) separability into 1-D convolutions within row and column elements, and (3) approximation by a binomial distribution which is recursively defined. The Gaussian (actually, binomial) convolver on this chip is a parallel, pipelined version of a CCD architecture developed first by Sage and Lattes at MIT Lincoln Laboratories.

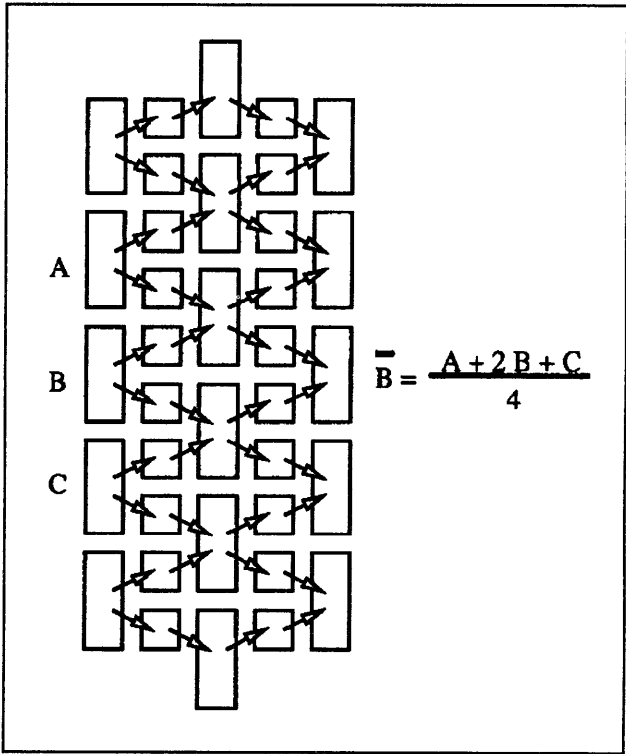
In addition to image sensing, CCDs are also capable of performing simple analog charge-domain computations such as charge packet split-



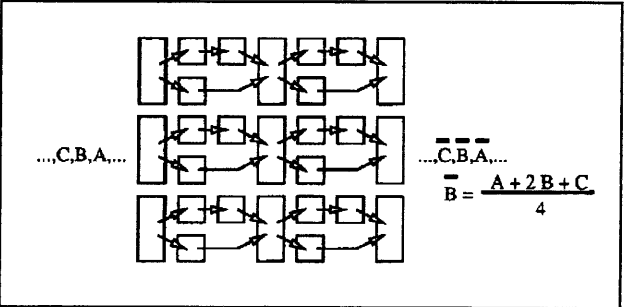
**Figure 4.** The parallel, pipelined architecture for high throughput image processing. The entire image is transferred through the column and row processors as a pipelined sequence of column vectors of image pixels. The column processors compute local interactions between neighboring column pixels in parallel. The row processors pipeline the computation of local interactions between neighboring row pixels by using additional delay elements.

ting, charge packet summation, charge packet subtraction, and delay. By combining the CCD structures for charge packet splitting and charge packet summation, a 1-D binomial convolution within parallel column elements can be directly implemented as shown in figure 5. With the addition of a delay element, a 1-D binomial convolution within sequential row elements can be implemented as shown in figure 6.

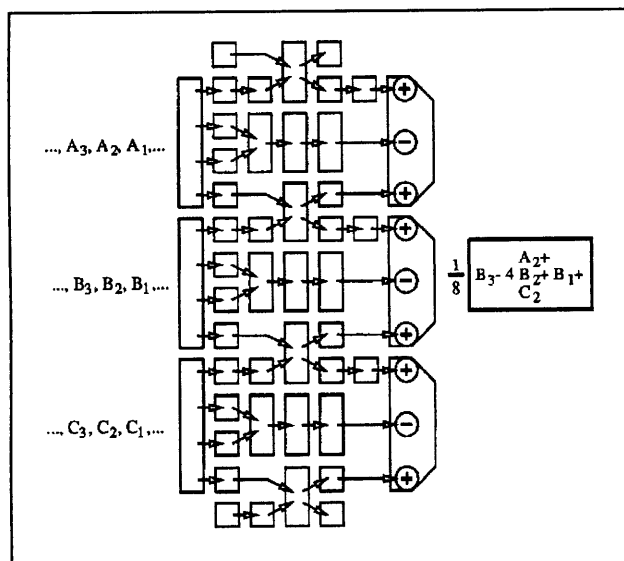
Similarly, CCD structures for charge packet splitting, charge packet summation, charge packet subtraction and delay elements can be combined as shown in figure 7 to perform a Laplacian convolution. Thus, the LoG processor can be simply realized by cascading the CCD structures for Laplacian convolution, 1-D binomial convolution within column elements, and 1-D binomial convolution within row elements. Notice that the parallel, pipelined architecture allows efficient integration with a CCD imager, as a column of image data can be directly clocked out of a CCD imager into the LoG processor. Furthermore, the parallel, pipelined architecture can be extended to the implementation of other linear and nonlinear image filters by proper modification of the processing elements. This prototype device demonstrates the feasibility and computational power of integrating CCD signal processors with CCD imaging arrays for real-time image processing.



**Figure 5.** CCD structure for 1-D binomial convolution within column elements. As the column elements are clocked in parallel from left to right, the analog charge packets are split into halves. Subsequently, the halved charge packets are summed in parallel to compute the average of neighboring column elements. Repeated operations with cascaded structures increase the order and extent of the 1-D binomial distribution within column elements.



**Figure 6.** CCD structure for 1-D binomial convolution within row elements. As row elements are clocked from left to right, the analog charge packet is split into halves. Half of the charge packet is delayed, and the other half of the charge packet is summed with the half of the preceding charge packet to compute the average of sequential row elements. Repeated operations with cascaded structures increase the order and extent of the 1-D binomial distribution within row elements.



**Figure 7.** CCD structure for Laplacian convolution. As image data are clocked from left to right, the charge packets are split, delayed, summed, and subtracted to compute the discrete convolution approximating the Laplacian computation.

### 6.2.4 Overall Project Scope

Listed below are the individual efforts that are underway or have been completed in this project. The single-chip systems are chips with imagers for input that produce highly processed, low-bandwidth output that can be readily handled by conventional digital systems. David Standley's chip in section 1.1.3. is an example. The "modular vision system component chips" may or may not have imagers on board. They produce high bandwidth output such as filtered images or depth maps that require specialized processing hardware for high speed operation.

#### Single-Chip Systems

CMOS Image Moment Chip (completed)  
 CCD/CMOS Focus of Expansion Chip (in progress)

#### Modular Vision System Component Chips

CCD/CMOS Analog Convolver (completed)  
 CCD/CMOS Image Smoothing and Segmentation Chip (in progress)  
 CCD/CMOS Stereo Chip (in progress)  
 Switched-Capacitor Surface Reconstruction Chip (in progress)

#### Technology Development

CCD/CMOS Fabrication Process (completed)  
 Switched-Capacitor Resistive Grids (completed)

CMOS Image Segmentation Circuits (in progress)

#### Theory

Stability of Active Resistive-Grid Systems (completed)  
 Texture and Halftoning Using Markov Random Fields (in progress)  
 Switched-Capacitor Network Settling Time (in progress)  
 Least-Squares Camera Calibration Method (in progress)

#### Simulation

Parallel Simulator for Large Arrays of Analog Cells (in progress)

#### Publications

- Elfadel, I., and R. Picard. "Miscibility Matrices Explain the Behavior of Grayscale Textures Generated by Gibbs Random Fields." *SPIE Proceedings on Intelligent Robots and Computer Vision IX*, OE/Boston '90, Boston, Massachusetts, November 4-9, 1990.
- Keast, C.L., and C.G. Sodini. "A CCD/CMOS Process for Integrated Image Acquisition and Early Vision Signal Processing." *Proceedings SPIE Charge-Coupled Devices and Solid State Sensors*, Santa Clara, California, February 1990, pp. 152-161.
- Lumsdaine, A., J.L. Wyatt, Jr., and I.M. Elfadel. "Nonlinear Analog Networks for Image Smoothing and Segmentation." *J. VLSI Sig. Process.* Forthcoming.
- Lumsdaine, A., J. Wyatt, and I. Elfadel. "Nonlinear Analog Networks for Image Smoothing and Segmentation." *Proceedings of the IEEE International Symposium on Circuits and Systems*, New Orleans, Louisiana, May 1990, pp. 987-991.
- Sarpeshkar, R., J.L. Wyatt, Jr., N.C. Lu, and P.D. Gerber, "Mismatch Sensitivity of a Simultaneously Latched CMOS Sense Amplifier." *IEEE J. Solid-State Circuits.* Forthcoming.
- Standley, D. "Stability in a Class of Resistive Grid Networks Containing Active Device Realizations of Nonlinear Resistors." *Proceedings of the IEEE International Symposium on Circuits and Systems*, New Orleans, Louisiana, May 1990, pp. 1474-1477.



Umminger, C.B., and C.G. Sodini. "Switched Capacitor Networks for Monolithic Image Processing Systems." Submitted to *IEEE J. Solid-State Circuits*.

Wyatt, J.L., Jr., and M. Ilic. "Time-Domain Reactive Power Concepts for Nonlinear, Nonsinusoidal or Nonperiodic Networks." *Proceedings of the IEEE International Symposium on Circuits and Systems*, New Orleans, Louisiana, May 1990, pp. 387-390.

Wyatt, J., D. Standley, and B. Horn. "Local Computation of Useful Global Quantities Using Linear Resistive-Grid Networks." Poster session, Conference on Neural Networks for Computing, Snowbird, Utah, April, 1990.

Yang, W., and A.M. Chiang. "A Full Fill-Factor CCD Imager with Integrated Signal Processors." *Proceedings of the ISSCC*, San Francisco, California, February 1990, pp. 218-219.

### Patents

Decker, S., H.-S. Lee, and J.L. Wyatt, Jr. "Resistive Fuse Circuits for Image Segmentation and Smoothing." Submitted to U.S. Patent Office 12/14/90, receipt no. 628-340.

Standley, D., and B.K.P. Horn. "Analog VLSI Chip for Object Position and Orientation." Submitted to U.S. Patent Office 9/28/90, receipt no. 591,145.

### Theses

Umminger, C.B. *Switched Capacitor Networks for Machine Vision*. S.M. thesis. Department of Electrical Eng. and Comput. Sci. MIT, 1990.

Yang, W. *The Architecture and Design of CCD Processors for Computer Vision*. Ph.D. diss. Dept. of Electr. Eng. and Comput. Sci. MIT, 1990.

## 6.3 Techniques for Logic Synthesis, Verification and Testing

### Sponsors

Analog Devices Career Development Assistant Professorship  
U.S. Navy - Office of Naval Research  
Contract N0014-87-K-0825

### Project Staff

Professor Srinivas Devadas, James H. Kukula, Curtis S. Chen, Marina Frants, Kevin Lam, Amelia H. Shen, Filip Van Aelten

### 6.3.1 Introduction

In order to design and build large-scale computers that achieve and sustain high performance, it is essential to carefully consider reliability issues. Although these issues have several aspects, our focus here is on the important problem of detecting failures in a computer's VLSI components which are caused by errors in the design specification, implementation or manufacturing processes.

Design verification involves ensuring that the specification of a design is correct prior to carrying out its implementation. Implementation verification ensures that the manual design or automatic synthesis process is correct, i.e., checking that the mask-level descriptions obtained correctly implement the specification. Manufacture testing involves checking the complex fabrication process for correctness, i.e., checking that there are no manufacturing defects on the integrated circuit. It should be noted that the three verification mechanisms described above deal not only with verifying the functionality of the integrated circuit but also with its performance.

In the Custom Integrated Circuits Group at MIT, we are attempting to develop synthesis, specification and verification mechanisms that are versatile as well as efficient. More importantly, we are developing strategies that address reliability issues in the many steps of the complex VLSI design process, and this involves varied levels of design abstraction from behavioral specifications to mask-level layout. Because many aspects of reliability cannot be investigated by focusing on a single level of design abstraction, such a complete program is essential when addressing reliability.

In the following sections, we elaborate on our current work in developing CAD algorithms and tools for highly reliable VLSI circuits and on our plans for the future. Included in the description is our research in the development of provably correct behavioral transformations with applications to design and implementation verification, synthesis of testable VLSI circuits in the area of manufacture test, and the development of a framework for logic synthesis, verification and testing.

In Sections 6.3.2 through 6.3.4, we describe work categorized largely by the mathematical and algorithmic techniques used in each problem domain.

## 6.3.2 Design Verification

### *Introduction*

The specifications for a large VLSI circuit are typically expressed in a high-level language by a computer architect or VLSI designer. Then the architect uses high-level simulators to check whether the exact functionality of the circuit has been captured in the design. Unfortunately, an exhaustive simulation that can guarantee correctness is possible only for the smallest circuits. Ideally, the architect would like to formally verify and guarantee that the specifications are correct, so that, if they are implemented and fabricated without error, a functionally-correct IC will result.

Several errors may occur in the manual process of design specification. For example, in a recent personal computer (PC) product by a major company, a particular number which was supposed to be squared was multiplied by two instead. This was tracked to an error in the specification. The specification had been rigorously (but obviously not exhaustively) simulated. Protocol chips are typically small compared to general-purpose  $\mu$  processors, but experience has shown that they are particularly prone to deadlock errors, due to incorrect specifications.

While we believe that formal verification methods will never replace simulation, they can augment simulation-based correctness checking in design verification. Verification methods could uncover specification errors without simulating huge numbers of test cases or using exorbitant amounts of CPU time. We have recently begun work in the area of design verification, specifically in the areas of temporal-logic-based model checking and protocol verification.

### *Temporal Logic Model Checking*

Model checking, a design verification method developed initially by Clarke et al. at CMU,<sup>10</sup> (1) specifies a property that the design should satisfy as a formula in computation-tree logic (CTL) and (2) verifies a design specification against the CTL formula using a model checker.

The model checking approach suffers from several drawbacks. First, the designer has to specify man-

ually the properties that the specification is supposed to satisfy. This can become a tedious and error-prone task for complex properties and large designs. Second, not all useful properties can be specified in CTL. More expressive logics have been devised, but the complexity of model checking increases for such logics. Third, model checking entails the traversal of the entire state space of the design specification, which can be huge. The last mentioned problem is called state explosion.

We have developed strategies to efficiently traverse the state space of a given design specification (or implementation) using a mixed depth-first/breadth-first strategy (c.f. "Sequential Logic Verification" on page 276) as opposed to previous traversal algorithms.<sup>11</sup> This traversal method is more complex to implement than the traversal methods proposed in the past, and model checking under this approach requires substantial modifications to retain efficiency. We will develop a new model checking algorithm based on our traversal algorithm. We believe that this new algorithm will be applicable to larger circuits than previous approaches have been.

### *Protocol Verification by Symbolic Analysis*

One problem with most current approaches to design verification is that they do not scale well with problem size. For instance, model checking using currently available traversal algorithms for an 8-bit circuit will typically take significantly more time than for a 4-bit version of the same circuit, even though the property to be checked for is independent of bit-width.

Symbolic analysis can alleviate the scalability problem. For instance, checking a protocol corresponding to interdependent processes and interacting sets of timers for correctness involves checking to ensure that deadlock does not occur. More simply stated, control does not remain indefinitely in a particular loop. One way we can analyze the loop is to traverse the entire state space represented by the loop, but this approach has the same drawbacks mentioned above. A symbolic analysis can result in a significantly faster judgment regarding the nature of the loop. The

<sup>10</sup> E.M. Clarke and O. Grumberg, "Research on Automatic Verification of Finite State Concurrent Systems," *Ann. Rev. Comput. Sci.* 2: 269-290 (1987).

<sup>11</sup> A. Ghosh, S. Devadas, and A.R. Newton, "Verification of Interacting Sequential Circuits," paper presented at the 27th Design Automation Conference, Orlando, Florida, June 1990, pp. 213-219.

existence or non-existence of exit conditions can be determined easily.

We are attempting to classify protocols based on their amenability to symbolic analysis. We have found that the state space corresponding to loops in a common class of protocols can be simply specified using a linear system of equations. This implies that the reachability analysis of such protocols can be accomplished in polynomial time. A traversal method that depends on the bit-width could take exponential time for these circuits. We are currently trying to generalize our method to handle a larger class of protocols.

### 6.3.3 Implementation Verification

#### *Introduction*

During the design process, increasingly sophisticated optimization tools that modify a design to improve its area, performance or power dissipation characteristics are applied simultaneously or in sequence at various levels of design abstractions. Beginning from a behavioral specification, we typically move through register-transfer level representations, symbolic finite state machine representations, logic-level representations and transistor-level representations all the way to mask-level layout.

In order to ensure a working IC, each step of the design process above has to be checked for correctness. A viewpoint might be that we can check the transformations applied by the CAD tools for correctness once, and all the designs produced by the CAD system will be "correct by construction." Verifying the correctness of transformations in a synthesis system is absolutely necessary, but our experience has found that it is equally essential to provide an independent verification mechanism. For example, in logic synthesis systems all logic optimization steps are typically verified, formally or informally, to be correct, but experience has shown that it is necessary to supply *post facto* verification procedures to independently verify the results of the application of logic transformations on a particular example. Similarly, while behavioral transformations may have been proved correct, the correctness of their realization in a particular software system is not presently amenable to proof, and, as a result, the final product of their application in synthesizing a circuit also requires independent verification.

#### *Timing Verification*

It is important to determine efficiently and accurately the critical path delay of a VLSI circuit so that the clock period can be chosen to be greater than the critical delay. Overestimating the critical path can lead to an unnecessarily slow design, while underestimating the critical path has dangerous reliability implications.

Determining the critical path of a combinational circuit is complicated because a combinational circuit may have false paths, i.e., paths that are never sensitized during normal system operation. Therefore, simply picking the longest path in the circuit as the critical path can result in gross overestimations of critical delay.

Viability has been proposed by McGeer and Brayton as a robust and correct criterion for determining the critical path delay of a combinational circuit. Currently, algorithms that determine the longest viable path in a circuit require large amounts of CPU time on medium-large circuits. We are investigating more efficient means of determining the longest viable path in a given network. In particular, we see a connection between determining the longest viable path and generating a test for a particular multiple stuck-at fault in the circuit.

#### *Circuit-Level Verification*

Logic net-lists can be implemented in a variety of circuit-level methodologies. Static or dynamic circuits in CMOS or nMOS may be used. In the case of dynamic circuits, a wide range of clocking methodologies serve as design alternatives. Different methodologies impose differing sets of restrictions on the topology and interconnection of the circuit blocks. Violations of these circuit design rules can go undetected during simulation. It is important that one checks for schematic-level correctness under the chosen design methodology.

Recent work by Bamji and Allen<sup>12</sup> formalized the process of verifying that a circuit-level schematic obeys a particular design methodology. Circuit correctness was tied to a rigorous set of context-free grammar composition rules. These rules define how a small set of module symbols may be combined for circuits adhering to the design methodology. Schematic net-lists are represented by graphs, and composition rules are defined as graph

<sup>12</sup> C. Bamji and J. Allen, "GRASP: A Grammar-Based Schematic Parser," in *Proceedings of the 26th Design Automation Conference*, June 1990, pp. 448-453.

transformations similar to grammatical productions. Starting with a circuit net-list, a hierarchical parse tree that can demonstrate the well-formedness of the circuit is constructed. Since the procedure is hierarchical and incremental, it operates one or two orders of magnitude faster than previous approaches.

A drawback to the above approach is that it can handle only circuit design methodologies which can be specified under context-free grammar rules. While context-free grammars can be used to specify a variety of circuit design methodologies (e.g., a 2-phase CMOS clocking methodology), complex special-case rules typically cannot be specified. We are currently investigating the use of higher-level grammars under a similar mechanism which will allow for the verification of more complex circuit design methodologies.

### **Sequential Logic Verification**

It is necessary to verify the equivalence of combinational or sequential circuits described at the logic level to check that a logic design step or a logic optimization system has not introduced errors into the design.

Recently, there has been considerable progress in the area of sequential logic verification. Two broad classes of approaches can be identified — approaches that use reduced, ordered Binary Decision Diagrams as their base representation to perform a breadth-first traversal of the State Transition Graph (STG) of a FSM<sup>13</sup> and approaches that use cube representations to perform a depth-first traversal of the STG.<sup>14</sup> The former approach performs significantly better than the latter on datapath-like FSMs whose STGs are highly connected, while the latter outperforms the former on counter-like FSMs where there are long chains of states.

We have developed a depth-first traversal technique for sequential machines that enables the

traversal of counter FSMs in  $O(n)$  steps, where  $n$  is the number of bits in the counter. Our depth-first geometric chaining is based on it traversing geometrically increasing (or decreasing) chains of states and edges in the STG in any given set of steps. Other STG traversal approaches require  $2^n$  steps, since only one state can be traversed in each step. We believe that a synergistic combination of depth-first geometric chaining and previously proposed breadth-first approaches offers an efficient traversal technique for a broad class of circuits that include both datapath and control portions. We will implement and investigate the applicability of this mixed depth-first/breadth-first strategy on real-life circuits.

### **Behavioral Verification**

As described above, equivalence checking between two finite-state automata or two combinational logic circuits is precisely defined and supported by a body of theoretical work. The major issue in the use of these algorithms is CPU efficiency. In contrast, verifying that a logic-level description correctly implements a behavioral specification is considerably less developed. One major hindrance toward a precise notion of behavioral verification has been that parallel, serial or pipelined implementations of the same behavioral description can be implemented in finite-state automata with different input/output behaviors.<sup>15</sup>

In recent work,<sup>16</sup> we used nondeterminism to model the degree of freedom that is afforded by parallelism in a behavioral description that also contains complex control. Given some assumptions, we showed how the set of finite automata derivable from a behavioral description under all possible schedules of operations can be represented compactly as an input-programmed automaton (p-Automaton). The above method allows for extending the use of finite-state automata equivalence-checking algorithms to the problem of behavioral verification.

---

<sup>13</sup> O. Coudert, C. Berthet, and J.C. Madre, "Verification of Sequential Machines Using Boolean Functional Vectors," In IMEC-IFIP International Workshop on Applied Formal Methods for Correct VLSI Design, November 1989, pp. 111-128.

<sup>14</sup> S. Devadas, H-K. T. Ma, and A.R. Newton, "On the Verification of Sequential Machines at Differing Levels of Abstraction," in *IEEE Trans. Comput.-Aided Des.* 7: 713 (June 1988); A. Ghosh, S. Devadas, and A.R. Newton, "Verification of Interacting Sequential Circuits," paper presented at the 27th Design Automation Conference, Orlando, Florida, June 1990, pp. 213-219.

<sup>15</sup> M.C. McFarland and A.C. Parker, "An Abstract Model of Behavior for Hardware Descriptions," in *IEEE Trans. Comput.* C-32: 621 (1983).

<sup>16</sup> S. Devadas and K. Keutzer, "An Automata-Theoretic Approach to Behavioral Equivalence," in *Proceedings of the International Conference on Computer-Aided Design*, November 1990.

We are in the process of extending the above approach to handle a richer set of behavioral transformations including pipelining and complex don't care conditions specified at a behavioral level. We believe that the use of p-Automata represents a general method to verify behavior against implementation.

### 6.3.4 Manufacture Test and VLSI Synthesis for Testability

#### *Introduction*

Once a chip has been fabricated, it is essential to check that no manufacturing defects have been introduced during the fabrication process. Manufacturing defects are modeled at different levels of design abstraction, and these fault models are used as a base for generating test vectors for a given design.

Fault models associated with defects that alter logical functionality are called logical fault models. For example, the most commonly used and the simplest logical fault model is the single stuck-at fault model, where a single wire in the IC is tied to a constant logical "1" or "0." Enhanced logical fault models like the multiple stuck-at fault model and the bridging fault model are more comprehensive, i.e., test vectors generated under these fault models typically detect a larger fraction of possible manufacturing defects. Deriving the test vectors under these fault models is considerably more difficult. For instance, given  $k$  wires in a circuit, there are only  $2k$  single stuck-at faults, but  $3^k - 1$  multiple stuck-at faults (multifaults), an astronomical number even for small circuits. However, for some applications reliability considerations are paramount, and it may be necessary to obtain as comprehensive a test set as possible.

Just checking the logical functionality of a fabricated circuit is not enough, its performance has to be verified as well. Manufacturing defects can degrade performance without altering logical functionality. Temporal or dynamic fault models like the gate delay fault model, transistor stuck-open fault model, and the path delay fault model have been proposed to model manufacturing defects that alter the performance characteristics of

a circuit. As with the more comprehensive logical fault models, test generation under these dynamic fault models is considerably more difficult than under the simplistic single stuck-at fault model.

Most circuits being designed today are not testable for dynamic faults, and testing for multifaults and bridging faults requires exorbitant amounts of CPU time. Also, post-design testability measures cannot, in general, increase the testability of the circuit under these fault models. Our solution to this problem is to completely integrate testability considerations into an automatic design process, i.e., synthesize the circuit to be highly or fully testable under any given fault model.

VLSI circuits are typically sequential, and the problem of generating tests for sequential circuits is more difficult than that for combinational circuits, under any given fault model. While techniques such as scan design can transform a sequential test problem into a simpler one of combinational test generation, these techniques have not found widespread use due to reasons stemming from the design constraints, and the area and performance penalties associated with scan methods. Our research in testing also targets test generation under stuck-at and delay fault models for non-scan VLSI circuits, and synthesis for sequential testability.

#### ***Combinational Logic Synthesis for Testability Under Enhanced Fault Models***

Combinational logic synthesis and optimization is a well-understood problem. The traditional cost functions used in synthesis have been layout area and circuit performance. Researchers have only recently begun to investigate the effect of logic transformations on the single and multiple stuck-at fault testability of a circuit<sup>17</sup> We are working on deriving algebraic and Boolean transformations for logic circuits to improve testability (and remove redundancy) with minimal or no impact on area and performance. Methods for implicit/test generation, i.e., obtaining test vectors to detect stuck-at faults as a by-product of the optimization steps, will be investigated. These methods are potentially much more efficient than explicit test generation algorithms, especially for multiple stuck-at faults.

<sup>17</sup> G.D. Hachtel, R.M. Jacoby, K. Keutzer, and C.R. Morrison, "On the Relationship Between Area Optimization and Multifault Testability of Multilevel Logic," paper published in the International Conference on Computer-Aided Design, November 1989, 422-425, (extended version submitted to *IEEE Trans. Comput.-Aided Des.*); S. Devadas and K. Keutzer, "Necessary and Sufficient Conditions for Robust Delay-Fault Testability of Logic Circuits," in the *Proceedings of the Sixth MIT Conference on Advanced Research on VLSI* (Cambridge: MIT Press, 1990).

Physical defects in ICs can degrade performance without affecting logic functionality. It is important to ensure that a given IC satisfies its timing specifications — This requires performing a delay fault test. This is especially important for aggressive design methodologies that push processing technology to the utmost. We have done some recent work on the synthesis of delay-fault testable combinational circuits. Certain classes of large circuits can be synthesized for complete robust path-delay-fault testability (which is the most comprehensive fault model devised thus far) with small area and performance penalties.<sup>18</sup> The primary drawbacks of current techniques are that non-flattenable random logic circuits, those that cannot be reduced to sum-of-products form, cannot be synthesized efficiently. We are currently improving upon the available synthesis techniques through the use of Binary Decision Diagrams as our base representation. Multiplexor-based networks, derived from replacing the nodes of Binary Decision Diagrams by 2-input multiplexors, have very interesting testability properties. Exploiting these properties can lead to an efficient synthesis-for-test strategy.

We are also investigating dynamic compaction techniques for test sets during synthesis. Compaction techniques tailored toward multiple stuck-at faults and delay faults are being developed. For instance, we believe that complete multiple stuck-at-fault test sets can be generated for random logic blocks that are only two to four times the size of the single-fault test set. Nonrobust delay test methodologies are not as comprehensive as robust delay test methodologies but offer scope for smaller test vector sets.

## ***Sequential Logic Synthesis for Testability Under Enhanced Fault Models***

VLSI circuits are typically sequential. A popular method of reducing the complexity of sequential test generation has involved the use of scan design methodologies (e.g., LSSD). Sequential test generation<sup>19</sup> and sequential logic synthesis for testability techniques that ensure non-scan single-stuck-at fault testability<sup>20</sup> have been, and will continue to be, a subject of investigation at MIT and elsewhere.

We are now looking toward the synthesis of sequential circuits for delay-fault testability. It is well known that arbitrary vector pairs cannot be applied to a sequential circuit under a standard-scan design methodology. This means that even if the combinational logic of a sequential circuit is made fully robustly delay-fault testable, it may not be fully testable under a standard-scan design implementation. An enhanced scan flip-flop that can store two bits of state is required. This "triple-latch" flip-flop increases the area of the circuit.

We have made preliminary investigations into the synthesis of non-scan sequential circuits for gate-delay-fault testability<sup>21</sup>. These initial approaches were limited to circuits of small size. Several avenues are being explored in current research. First, we are developing techniques that ensure fully robustly gate-delay-fault testable chips under a standard-scan design methodology. Second, we are exploring the use of partial enhanced-scan and standard-scan design methods to improve the robust delay-fault testability of sequential circuits. Third, we are investigating the use of partial standard-scan and non-scan design. Our goal is to eventually obtain non-scan circuits that are completely robustly gate-delay-fault testable, with negligible area/performance penalties.

---

<sup>18</sup> S. Devadas and K. Keutzer, "Design of Integrated Circuits Fully Testable for Delay Faults and Multifaults," In *Proceedings of the International Test Conference*, October 1990, pp. 284-293; S. Devadas and K. Keutzer, "Synthesis and Optimization Procedures for Delay-Fault Testable Combinational Logic," paper presented at the 27th Design Automation Conference, Orlando, Florida, June 1990.

<sup>19</sup> A. Ghosh, S. Devadas, and A. R. Newton, "Test Generation and Verification of Highly Sequential Circuits," In *Proceedings of the International Conference on Computer-Aided Design*, November 1989, pp. 362-365.

<sup>20</sup> S. Devadas and K. Keutzer, "A Unified Approach to the Synthesis of Fully Testable Sequential Machines," *IEEE Trans. Comput.-Aided Des.*, 10(1): 39-50 (1991); S. Devadas and H-K. T. Ma, "Easily Testable PLA-based Finite State Machines," *IEEE Trans. on Comput.-Aided Des.*, 9(5): 614-611 (1990).

<sup>21</sup> S. Devadas, "Delay Test Generation for Synchronous Sequential Circuits," in *Proceedings of the International Test Conference* (Washington, D.C.: August 1989), pp. 144-152.

**Publications**

- Ashar, P., S. Devadas, and A.R. Newton. "Testability-Driven Decomposition of Large Finite State Machines." *Proceedings of International Conference on Computer Design: VLSI in Computers and Processors*, Boston, Massachusetts, September 1990.
- Ashar, P., S. Devadas, and A.R. Newton. "Optimum and Heuristic Algorithms for A Formulation of Finite State Machine Decomposition." *IEEE Trans. Comput.-Aided Des.* 10(3): 296-310 (1991).
- Ashar, P., A. Ghosh, S. Devadas, and K. Keutzer. "Implicit State Transition Graphs: Applications to Sequential Logic Synthesis and Test." *Proceedings of the International Conference on Computer-Aided Design*, Santa Clara, California, November 1990.
- Ashar, P., S. Devadas, and A. Ghosh. "Boolean Satisfiability and Equivalence Checking Using General Binary Decision Diagrams." *International Conference on Computer Design: VLSI in Computers and Processors*, Cambridge, Massachusetts, October 1991.
- Ashar, P., S. Devadas, and K. Keutzer. "Testability Properties of Multilevel Logic Networks Derived From Binary Decision Diagrams." *Proceedings of the Conference on Advanced Research in VLSI*, Santa Cruz, California, March 1991.
- Ashar, P., S. Devadas, and A.R. Newton. "Irredundant Interacting Sequential Machines Via Optimal Logic Synthesis." *IEEE Trans. Comput.-Aided Des.* 10(3): 311-325 (1991).
- Bryan, M.J., S. Devadas, and K. Keutzer. "Testability-Preserving Circuit Transformations." *Proceedings of the International Conference on Computer-Aided Design*, Santa Clara, California, November 1990.
- Bryan, M.J., S. Devadas, and K. Keutzer. "Analysis and Design of Regular Structures for Robust Dynamic Fault Testability." *IEEE International Symposium on Circuits and Systems*, Singapore, June 1991.
- Bryan, M.J., S. Devadas, and K. Keutzer. "Analysis and Design of Regular Structures for Robust Dynamic Fault Testability." *Inter. J. Comput.-Aided VLSI Des.* Forthcoming.
- Cheng, K-T., S. Devadas, and K. Keutzer. "Robust Delay-Fault Test Generation and Synthesis for Testability Under a Standard Scan Methodology." 28th Design Automation Conference, San Francisco, California, June 1991.
- Devadas, S. "Delay Test Generation for Synchronous Sequential Circuits." *Inter. J. Comput.-Aided VLSI Des.* Forthcoming.
- Devadas, S. "Optimization of Interacting Finite State Machines Using Sequential Don't Cares." *IEEE Trans. Comput.-Aided Des.* Forthcoming.
- Devadas, S. "Optimal Layout Via Boolean Satisfiability." *Inter. J. Comput.-Aided VLSI Des.* 2(2): 251-262 (1990).
- Devadas, S. "Optimization of Interacting Finite State Machines Using Sequential Don't Cares." *IEEE Trans. Comput.-Aided Des.* Forthcoming.
- Devadas, S., and H-K. T. Ma. "Easily Testable PLA-based Finite State Machines." *IEEE Trans. Comput.-Aided Des.* 9(5): 614-611 (1990).
- Devadas, S., and K. Keutzer. "A Unified Approach to the Synthesis of Fully Testable Sequential Machines." *IEEE Trans. Comput.-Aided Des.* 10(1): 39-50 (1991).
- Devadas, S., and K. Keutzer. "Synthesis of Robust Delay-Fault Testable Circuits: Practice." *IEEE Trans. Comput.-Aided Des.* Forthcoming.
- Devadas, S., and K. Keutzer. "Synthesis of Robust Delay-Fault Testable Circuits: Theory." *IEEE Trans. Comput.-Aided Des.* Forthcoming.
- Devadas, S., and K. Keutzer. "An Automata-Theoretic Approach to Behavioral Equivalence." *Proceedings of the International Conference on Computer-Aided Design*, Santa Clara, California, November 1990.
- Devadas, S., K. Keutzer, and A. Ghosh. "Recent Progress in VLSI Synthesis for Testability." *Proceedings of the VLSI Test Symposium*, Atlantic City, April 1991.
- Devadas, S., and K. Keutzer. "Design of Integrated Circuits Fully Testable for Delay Faults and Multifaults." *Proceedings of the International Test Conference*, Washington, D.C., September 1990.
- Devadas, S., and A.R. Newton. "Exact Algorithms for Output Encoding, State Assignment and Four-Level Boolean Minimization." *IEEE Trans. Comput.-Aided Des.* 10(1): 13-27 (1991).
- Devadas, S., and K. Keutzer. "An Automata-Theoretic Approach to Behavioral

Equivalence." *Intern. J. Comput.-Aided VLSI Des.* Forthcoming.

Devadas, S., K. Keutzer, and J.K. White. "Estimation of Power Dissipation in CMOS Combinational Circuits Using Boolean Function Manipulation." *IEEE Trans. Comput.-Aided Des.* Forthcoming.

Devadas, S., K. Keutzer, and S. Malik. "A Synthesis-Based Approach to Test Generation and Compaction for Multifaults." 28th Design Automation Conference, San Francisco, California, June 1991.

Devadas, S., K. Keutzer, and A.S. Krishnakumar. "Design Verification and Reachability Analysis Using Algebraic Manipulation." International Conference on Computer Design: VLSI in Computers and Processors, Cambridge, Massachusetts, October 1991.

Ghosh, A., S. Devadas, and A.R. Newton. "Heuristic Minimization of Boolean Relations Using Testing Techniques." *Proceedings of International Conference on Computer Design: VLSI in Computers and Processors*, Boston, Massachusetts, September 1990.

Ghosh, A., and S. Devadas. "Implicit Depth-First Transversal of Sequential Machines." IEEE International Symposium on Circuits and Systems, Singapore, June 1991.

Ghosh, A., S. Devadas, and A.R. Newton. "Test Generation and Verification of Highly Sequential Circuits." *IEEE Trans. Comput.-Aided Des.* Forthcoming.

Ghosh, A., S. Devadas, and A.R. Newton. "Synthesis for Sequential Logic Testability using Register-Transfer Level Descriptions." *Proceedings of the International Test Conference*, Washington, D.C., September 1990.

## 6.4 Mixed Circuit/Device Simulation

### Sponsors

IBM Corporation  
U.S. Navy - Office of Naval Research  
Contract N00014-87-K-0825

### Project Staff

Mark W. Reichelt, Professor Jacob White, Professor Jonathan Allen

For critical applications, the four-terminal lumped models for MOS devices used in programs like SPICE are not sufficiently accurate. Also, it is difficult to relate circuit performance to process changes using lumped models. Sufficiently accurate transient simulations can be performed if, instead of using a lumped model for each transistor, some of the transistor terminal currents and charges are computed by numerically solving the drift-diffusion based partial differential equation approximation for electron transport in the device. However, simulating a circuit with even a few of the transistors treated by solving the drift-diffusion equations is very computationally expensive because the accurate solution of the transport equations of an MOS device requires a two-dimensional mesh with more than a thousand points.

One approach to accelerating this kind of mixed device and circuit simulation is to use waveform relaxation to perform the transient simulation, not only at the circuit level, but also inside the devices being simulated with a drift-diffusion description. In the present investigation, the WR algorithm is being applied to the sparsely-connected system of algebraic and ordinary differential equations in time generated by standard spatial discretization of the drift-diffusion equations that describe MOS devices. Several theoretical results about the uniformity of WR convergence for the semiconductor problem have been proved,<sup>22</sup> and we have also completed experiments using waveform relaxation to perform transient two-dimensional simulation of MOS devices. Speed and accuracy comparisons between standard direct methods and red/black overrelaxed waveform-relaxation-Newton indicate that for the experiments examined, calculated terminal currents match well between the methods, and our overrelaxed waveform-relaxation-Newton

---

<sup>22</sup> M. Reichelt, J. White, J. Allen and F. Odeh, "Waveform Relaxation Applied to Transient Device Simulation," *Proceedings of the International Symposium on Circuits and Systems*, Espoo, Finland, June 1988, pp. 1647-1650.



method was five to eleven times faster.<sup>23</sup> Currently, efforts are underway to develop the WORDS program (Waveform OverRelaxation Device Simulator) into a more robust code, with adaptive SOR parameterization; Poisson-only, or one or two carrier simulation options; and improved physical modeling. All physical models were verified by comparison and agreement with the industry-standard device simulator PISCES. Finally, a parallel version of WORDS was written for the iPSC/2 parallel architecture, demonstrating a speedup factor of 15.5 on 16 processors.

## 6.5 Simulation Algorithms for Clocked Analog Circuits

### Sponsors

AT&T  
Analog Devices  
Digital Equipment Corporation  
IBM Corporation  
National Science Foundation  
Grant MIP 88-58764  
U.S. Navy - Office of Naval Research  
Contract N00014-87-K-0825

### Project Staff

Steven B. Leeb, Luis M. Silveira, Professor Jacob White

A challenging problem in the area of analog circuits is the simulation of clocked analog circuits like switching filters, switching power supplies, and phase-locked loops. These circuits are computationally expensive to simulate using conventional techniques because these kinds of circuits are all clocked at a frequency whose period is orders of magnitude smaller than the time interval of interest to the designer. To construct such a long time solution, a program like SPICE or ASTAP must calculate the behavior of the circuit for many high frequency clock cycles. The basic approach to simulating these circuits more efficiently is to exploit only the property that the behavior of such a circuit in a given high frequency clock cycle is similar, but not identical to,

the behavior in the preceding and following cycles. Therefore, by accurately computing the solution over a few selected cycles, an accurate long time solution can be constructed.

Simulating clocked analog systems is an old problem, but this novel approach has led to a very efficient algorithm for the distortion analysis of switched-capacitor filters. The idea is based on simulating selected cycles of the high-frequency clock accurately with a standard discretization method and pasting together the selected cycles by computing the low frequency behavior with a truncated Fourier series. If carefully constructed, the nonlinear system that must be solved for the Fourier coefficients is almost linear and can be solved rapidly with Newton's method.<sup>24</sup> Transient behavior, important for switching power supply designers, has also been accelerated using similar techniques.<sup>25</sup> In particular, the "envelope" of the high-frequency clock can be followed by accurately computing the circuit behavior over occasional cycles.

## 6.6 Parallel Simulation Algorithms for Analog Array Signal Processors

### Sponsors

National Science Foundation  
Grant MIP 88-58764  
U.S. Navy - Office of Naval Research  
Contract N00014-87-K-0825

### Project Staff

Andrew Lumsdaine, Luis M. Silveira, Professor John L. Wyatt, Jr., Professor Jacob White

The "vision circuits" form a class of circuits which, for the most part, cannot be simulated with traditional programs. These circuits are necessarily very large and must be simulated exactly at an analog level (i.e., one cannot perform simulations at a switch or gate level as is commonly done with very large digital circuits). Typical analog circuit simulators can not handle vision circuits simply

<sup>23</sup> M. Reichelt, J. White, and J. Allen, "Waveform Relaxation for Transient Simulation of Two-Dimensional MOS Devices," *Proceedings of the International Conference on Computer-Aided Design*, Santa Clara, California, October 1989, pp. 412-415.

<sup>24</sup> K. Kundert, J. White, and A. Sangiovanni-Vincentelli, "A Mixed Frequency-Time Approach for Distortion Analysis of Switching Filter Circuits," *IEEE J. Solid-State Circ.* 24(2): 443-451 (1989).

<sup>25</sup> J. White and S. Leeb, "An Envelope-Following Approach to Switching Power Converter Simulation," *IEEE Trans. Power Electron.*

because of their immense size, since the computation time for these simulators grows super-linearly with the size of the circuit. However, because these circuits are somewhat similar to certain discretized partial differential equations, one can exploit their special characteristics to obtain efficient simulation techniques.

Over the last past year, we have completed the development of CMVSIM, a program for circuit-level simulation of grid-based analog signal processing arrays which uses a massively parallel processor. CMVSIM uses: (1) the trapezoidal rule to discretize the differential equations that describe the analog array behavior, (2) Newton's method to solve the nonlinear equations generated at each time-step, and (3) a block conjugate-gradient squared algorithm to solve the linear equations generated by Newton's method. Excellent parallel performance is achieved through the use of a novel, but very natural, mapping of the circuit data onto the massively parallel architecture. The mapping takes advantage of the underlying computer architecture and the structure of the analog array problem. Experimental results demonstrate that a full-size Connection Machine can provide a 1400 times speedup over a SUN-4/280 workstation.<sup>26</sup>

## 6.7 Numerical Simulation of Short Channel MOS Devices

### Sponsors

Analog Devices  
U.S. Navy - Office of Naval Research  
Contract N00014-87-K-0825.

### Project Staff

Jennifer A. Lloyd, Joel Phillips, Khalid Rahmat, Professor Dimitri A. Antoniadis, Professor Jacob White

The model which is used in conventional device simulation programs is based on the drift-diffusion model of electron transport and does not accurately predict the field distribution near the drain in small geometry devices. This prediction is of particular importance for predicting oxide breakdown due to penetration by "hot" electrons. There are two approaches for more accurately computing the electric fields in MOS devices: one is based on adding an energy equation to the drift-diffusion

model and the second is based on particle or Monte-Carlo simulations.

In the first approach, an energy balance equation is solved along with the drift-diffusion equations so that the electron temperatures are computed accurately. This combined system is numerically less tame than the standard approach and must be solved carefully. The most serious problem was that with the physically correct mobility model, which utilizes the electron temperature, our simulator did not converge if the mesh spacing was not fine enough; this problem was especially acute in the region under the drain junction. The problem was inherent in the method used to discretize the energy balance equation. With a new discretization strategy, this problem was solved, and the simulator now converges even when a very coarse mesh is used.

Besides including the electron temperature dependence, the mobility model used in the simulator has been made more physically realistic by including the effects of the vertical gate field. To tune some of the empirical parameters in the simulator, extensive comparisons with data from fabricated devices, including some with very short channels (0.16  $\mu\text{m}$ ), were done. The simulator predicts the current to about 10-12 percent accuracy. The largest errors occur in deep saturation. This also leads to inaccuracies in the conductance in the deep saturation region.

Also, a simple method for estimating the substrate current has been implemented. The scheme assumes that the substrate current is proportional to the number of electrons above a threshold energy. With a fixed choice of this energy threshold, good agreement has been achieved with measured data except for low gate biases and high drain voltages.

In the area of Monte Carlo device simulation, we are focusing on transient calculations with self-consistent electric fields. Specifically, we are trying to apply the recently developed implicit particle methods. To apply these implicit particle methods to semiconductors, we are decomposing the field calculation into three parts: one due to charged particles, a second due to dopant ions, and a third due to boundaries. This allows the calculation of the electric field acting on every charged particle in the system to be performed rapidly and accurately. In particular, this allows the use of the fast multipole algorithm for the

<sup>26</sup> L.M. Silveira, A. Lumsdaine, and J. White, "Massively Parallel Simulation of Grid-based Analog Circuits," *Proceedings of the International Conference on Computer-Aided Design*, Santa Clara, California, October 1990; A. Lumsdaine, L. Silveira, and J. White, "CVSim Users Guide," in preparation.

particle-particle interactions. Currently, we have rewritten a Silicon Monte Carlo code from the National Center for Computational Electronics to use ensemble Monte Carlo methods and are now including the electric field calculations.<sup>27</sup>

## 6.8 Efficient 3-D Capacitance Extraction Algorithms

### Sponsors

IBM Corporation  
National Science Foundation  
Grant MIP 88-58764  
U.S. Navy - Office of Naval Research  
Contract N00014-87-K-0825.

### Project Staff

Keith S. Nabors, Songmin Kim, Professor Jacob White

We have developed a fast algorithm for computing the capacitance of a complicated 3-D geometry of ideal conductors in a uniform dielectric and have implemented the algorithm in the program FASTCAP. This method is an acceleration of the boundary-element technique for solving the integral equation associated with the multiconductor capacitance extraction problem. Boundary-element methods become slow when a large number of elements are used because they lead to dense matrix problems which are typically solved with some form of Gaussian elimination. This implies that the computation grows as  $n^3$ , where  $n$  is the number of panels or tiles needed to accurately discretize the conductor surface charges. Our new algorithm, which is a generalized conjugate residual iterative algorithm with a multipole approximation to compute the iterates, reduces the complexity so that accurate multiconductor capacitance calculations grow nearly as  $nm$  where  $m$  is the number of conductors.<sup>28</sup>

Our most recent work has been to develop an adaptive multipole algorithm, and establish a link between the FASTCAP program and the MIT Micro-Electrical-Mechanical CAD (MEMCAD) system, by using the MEMCAD system to draft and compute the capacitance of a realistic geometry. Our future work in this subject will be to

include dielectric interfaces and then to turn our attention to developing a multipole algorithm to accelerate the calculation of inductances.

## 6.9 Parallel Numerical Algorithms

### Sponsor

U.S. Navy - Office of Naval Research  
Contract N00014-87-K-0825.

### Project Staff

Andrew Lumsdaine, Mark W. Reichelt, Luis M. Silveira, Ricardo Telichevesky, Professor Jacob White

We are trying to develop parallel algorithms for circuit and device simulation that are effective on either massively parallel machines like the Connection Machine or on hypercube machines like the Intel hypercube. In the sections above we described our work on parallel device simulation using waveform relaxation and on parallel simulation algorithms for analog arrays. In addition, we are also trying to understand some fundamental problems of the interaction between architecture and certain numerical algorithms.

For example, the direct solution of circuit simulation matrices is particularly difficult to parallelize, in part because methods like parallel nested dissection are ineffective due to the difficulty of finding good separators. For that reason, general sparse matrix techniques are being studied. In particular, the interaction between sparse matrix data structures, computer memory structure, and multiprocessor communication is being investigated (with Professor W. Dally). The most interesting results from simulations so far is that communication throughput, and not latency, is more crucial to final performance.

Explicit integration methods avoid matrix solution and are, therefore, also interesting algorithms to use on parallel processors. For this reason, we investigated some of properties of the recently developed explicit exponentially-fit integration algorithms. We considered the multivariate test problem  $\dot{x} = -Ax$  where  $A \in \mathcal{R}^{n \times n}$  and is assumed to be connectedly diagonally-dominant with posi-

<sup>27</sup> J. Lloyd, J. Phillips, and J. White, "A Boundary-Element/Multipole Algorithm for Self-Consistent Field Calculations in Monte-Carlo Simulation," *Proceedings of Workshop on Numerical Modeling of Processes and Devices for Integrated Circuits*, NUPAD III, Honolulu, Hawaii, June 1990, pp. 81-82.

<sup>28</sup> K. Nabors and J. White, "FastCap: A Multipole-Accelerated 3-D Capacitance Extraction Program," *IEEE Trans. Comput.-Aided Des.*, forthcoming; K. Nabors, S. Kim, and J. White, "FastCap Users Guide," in preparation.

tive diagonals, because this models the equations resulting from the way MOS circuits are treated in timing simulation programs. We showed that for these problems the CINNAMON exponentially-fit algorithm is A-stable and gave an example where the algorithm in XPSim is unstable. A semi-implicit version of the XPSim algorithm was derived and shown to be A-stable. Examination of examples demonstrate that neither the stabilized XPSim algorithm nor the CINNAMON algorithm produces satisfactory results for very large timesteps. This is also the case for the semi-implicit algorithms as used in the MOTIS timing simulator. The effect of ordering on the accuracy and stability of the integration methods was also examined, and it was shown that ordering always enhances accuracy, although not significantly for large timesteps, and that the XPSim algorithm can be made more stable with a carefully chosen ordering.<sup>29</sup>

## 6.10 Integrated Circuit Reliability

### Sponsors

Digital Equipment Corporation  
U.S. Navy - Office of Naval Research  
Contract N00014-87-K-0825

### Project Staff

Kevin Lam, Professor Srinivas Devadas, Professor Jacob White

The high transistor density now possible with CMOS integrated circuits has made peak power dissipation and peak current density important design considerations. However, peak quantities in a logic circuit are usually a function of the input vector or vector sequence applied. This makes accurate estimation of peak quantities extremely difficult, since the number of input sequences that have to be simulated in order to find the sequence that produces the peak is *exponential* in the number of inputs to the circuit. By using simplified models of power and current dissipation, peak quantities, like power or current density, can be related to maximizing gate output activity and weighted to account for differing load capacitances or transistor sizes. Transformations can then be derived that convert a logic description of a circuit into a multiple-output Boolean function of the input vector or vector sequence, where each output of the Boolean function is associated with a logic gate output transition. It then follows that to find the input or input sequence that maximizes the quantity of interest, a *weighted max-satisfiability* problem must be solved. For the problem of estimating peak power dissipation, algorithms for constructing the Boolean function for dynamic CMOS circuits, as well as for static CMOS, which take into account dissipation due to glitching, have been derived and exact and approximate algorithms for solving the associated weighted max-satisfiability problem have been developed.<sup>30</sup>

---

<sup>29</sup> H. Neto, L. Silveira, J. White, and L. Vidigal, "On Exponential Fitting for Circuit Simulation," *Proceedings of the International Symposium on Circuits and Systems*, New Orleans, 1990, pp. 514-518.

<sup>30</sup> S. Devadas, K. Keutzer, and J. White, "Estimation of Power Dissipation in CMOS Combinational Circuits," *Proceedings of the Custom Integrated Circuits Conference*, Boston, 1990, pp. 19.7.1-19.7.6.