Feedback in Silicon Compilers

Eby G. Friedman

Abstract

In order for silicon compilers to become a truly viable VLSI design option, application specific information must be extracted and used as feedback to permit optimization of chip designs. This paper describes three significant phases necessary to provide feedback in silicon compilers and techniques for their implementation. In order to exemplify the techniques and utility of providing feedback in silicon compilers, an example of feedback that uses layout-extracted interconnect impedances to parameterize buffer cells is described.

Introduction

The trend in VLSI design has been the aggressive use of CAD software to minimize the time and tedium of chip layout and to permit the VLSI designer to emphasize the functional and timing requirements of system integration. As the application specific design requirements become incorporated into the overall structured CAD strategy, the ability to create final mask artwork from a high-level behavioral description or "silicon compilation" will become tractable. The trend toward true silicon compilation has been accelerating dramatically and, in a partial sense, exists today. Major portions of currently designed VLSI circuits are automatically routed, using powerful hierarchical automated layout systems. The ability to generate final geometric structures based on a precise high-level description is currently available for specific problems [1]–[3]. Cell compiler systems for generating Programmable Logical Arrays (PLAs) [4]–[5], Read Only Memories (ROMs), and Finite State Machines (FSMs) [6]–[7] are becoming increasingly commonplace. Functional cells can be automatically configured using parameterized cell generators [8]. These cell compiler capabilities are a subset of the overall silicon compiler problem and emphasize the need to consider the on-chip relationships of these various functional modules.

The current requirement in the development of a true silicon compiler capability is more than the accurate and complete generation of final mask geometries from a high-level functional description. Instead, the emphasis in developing silicon compilers needs to be oriented to producing chips of competitive performance and density. The automatic creation of an optimal chip design should be based on appropriate speed/power/area trade-offs inherent to the specific application being integrated. In order to automatically produce a fully optimized chip design in a silicon compiler sense, application specific data must be extracted from the circuit data base and used as feedback in creating an optimal design. This extracted information must then be transformed into a compatible format, which will optimally constrain the functional cells. Thus, to permit generalizing the ability to implement final geometries from behavioral descriptions in an optimal way, each functional cell must be automatically designed using well-defined, constraining, circuit parameters. Therefore, to ensure an optimal high performance silicon compilation capability, various mechanisms must be provided that will automatically parameterize the functional cells based on application specific feedback.

This paper describes the general requirements for implementing feedback in silicon compilers. The example of extracting interconnect parasitic impedances from automated layout to parameterize buffer cells is described to exemplify the techniques and advantages of using feedback to automatically configure final cell geometries. The next section describes the generic requirements of providing feedback mechanisms in silicon compilers between preliminary extracted information and cell geometries. Then, the following three sections describe, in general terms, the three significant phases necessary to provide feedback in silicon compilers. An example of providing feedback in silicon compilation is described in the section titled "A Parameterized Buffer Cell System Based on Parasitic Extraction." Finally, the last section summarizes the overall value of providing both feedback and in iterative passing of cell parameters in silicon compilers to create an optimal chip design.

The Need for Feedback Mechanisms in Silicon Compilers

Silicon compilation is becoming a serious design choice for integrating system functionality. Present-day capability permits complete final mask artwork to be automatically created from specific high-level descriptions. In order for silicon compilers to have general commercial applicability, they must meet the following criteria:

(a) The final product must be accurate and complete.
(b) The final circuit must satisfy all I/O and internal timing requirements.
(c) The silicon compiler must be applicable to a wide variety of system architectures.
(d) The final chip must utilize die area efficiently and not dissipate unnecessary power.
(e) The silicon compiler must be able to push the performance limitations of a given technology.
(f) These qualities must all be provided in a high turnaround environment.

In order to improve the density, performance, and power dissipation of chips generated by silicon compilers, feedback mechanisms based on application specific information must be provided. A system whereby application-dependent information is ex-

8755-3996/85/0500-0015$1.00 © 1985 IEEE

tracted and then transformed into a format compatible with parameterizing final mask geometries is necessary to improve the overall quality of an integrated circuit generated by a silicon compiler.

In summary, as depicted in Fig. 1, three mechanisms must occur when providing feedback in silicon compilers. The first, extraction, is the derivation of significant data from preliminary constraints. The second, transformation, is the configuring of that information into a compatible format. The third, parameterization, is the insertion of these enhanced parameters into functional cells so as to constrain the synthesis process in an optimal way.

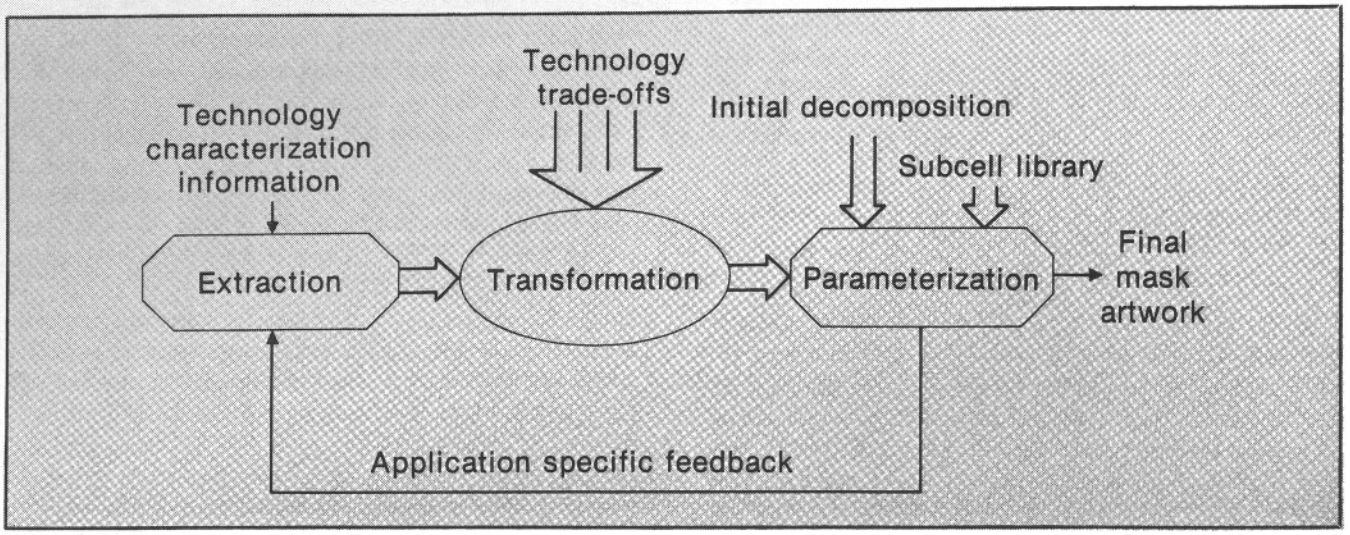


Fig. 1 A general overview of feedback in silicon compilation.

Extraction of Application Specific Information

As depicted in Fig. 1, in order to further improve the overall quality of the final chip design, application specific information must be extracted and used as feedback to iterate the entire synthesis process. As a chip design is completed, it is evaluated to determine whether it meets overall system specifications. If necessary, the silicon compiler design sequence can be reiterated utilizing the same feedback mechanisms based on the most updated optimal version.

When extracting application specific information, careful attention must be spent on the techniques that generate the information. The data must be carefully toleranced for the particular application area to which it is oriented and technology-dependent characterization data must be provided. Serious attention must be given to proper tolerancing of physical parameters since different applications have their own requirements (e.g., military systems demand much greater degrees of reliability and environmental variation than most commercial applications). In addition, as data are manipulated, many assumptions and physical approximations are made. Since this information is utilized as feedback to improve the entire system design, special effort must be made to ensure that the extracted information is accurate as well as properly toleranced.

The types of application specific information that need to be extracted should help to improve and constrain the overall integrated circuit. In order to improve the general utility of providing feedback in silicon compilers, the feedback mechanisms must consider all aspects of a final IC product such as system performance, die area, quiescent and transient power dissipation, as well as system architecture. Examples of useful information that could be extracted from a circuit in order to automatically reoptimize a given design are listed below:

(a) layout parasitics
(b) cell fan-out and/or fan-in
(c) relative spacing between functional elements
(d) Nbit organization
(e) signal flow

Transformation of Application Specific Information

Once the application specific information is properly extracted and toleranced, the data must be reconfigured into a format that can optimally parameterize the functional cell's organization, location, and/or final geometries. The particular transformation technique that is applicable to the feedback mechanism is entirely dependent upon the type of information that is being utilized as feedback. In addition, every feedback mechanism must be evaluated in terms of the particular semiconductor technology in which the final IC product is being implemented. In certain cases, the basic physics of the problem defines the given transformation technique. In other cases, the application entirely constrains the transformation mechanism. Listed below are various examples of different transformation techniques, each having its

own particular advantages and disadvantages:

(a) an algorithmic relationship
(b) a look-up table
(c) an expert system
(d) heuristics

Parameterization of the Functional Cell Geometries

Constraining parameters are used to optimally configure the organization, location, and/or physical geometries of the functional cells. In many cases, the functional cells are constrained by more than one set of parameters (e.g., one set for I/O locations, another for cell placement, and a third for performance optimization). For each particular feedback mechanism, a software program is required to manipulate the functional cells. In addition, the program requires a compatible subcell library to permit generation of the final mask artwork. The subcell library is referenced by the parameterization program, which generates the functional cell geometries.

Once the functional cells are properly configured by the parameterization set, an iterated optimized chip design exists. From this finalized chip description, new application specific information can be extracted for further reoptimization. Examples of different parameters that can be used to optimize performance, density, power dissipation, or provide more flexible system architectures are listed below:

(a) device geometries
(b) cell size
(c) I/O locations
(d) Nbit organization
(e) cell placement

Thus, for every feedback chain in a silicon compiler system, there must be a specific data extraction capability, a particular problem-related transformation mechanism, a parameterization software program, and its compatible subcell library. The number of iterations necessary is directly dependent upon the system specifications, the inherent traits of the particular process technology, and the time permitted to complete the chip design. As design iterations are completed, new sets of parameters are passed, which further improve a final chip design. Therefore, for a silicon compiler to be able to truly optimize area, performance, and power dissipation, and be applicable to general architectures, complete feedback mechanisms must be integrated into the silicon compiler system.

A Parameterized Buffer Cell System Based on Parasitic Extraction

An example of providing feedback to improve chip performance, area, and power dissipation is the automatic generation of buffer cells from layout-extracted interconnect impedances. The use of sophisticated, hierarchically controllable, automated layout systems has become increasingly common within the industry as evidenced by their common use in gate arrays and standard cells. These tools permit automated layout in an architecturally unconstrained manner and have exhibited significant applicability to the generalized requirements of silicon compilers. As large functional blocks are automatically laid out, significant quantities of previously undetermined interconnect capacitance are generated. As has become well known and accepted in the design of small geometry VLSI circuits, a significant portion of the RC delay of a given signal path is due to cell-to-cell interconnect [9]. Therefore, it is imperative that these capacitive and resistive loads be calculated accurately and efficiently and incorporated into the performance analyses of the design process. Silicon compilers must achieve the performance requirements necessary to permit these systems to become a truly viable VLSI design option. Toward this end, parasitic interconnect loads must be automatically extracted and utilized as feedback in the design of functional cells. A flow diagram of this feedback mechanism is depicted in Fig. 2.

An in-house VLSI design system [10] is used to illustrate this feedback mechanism. The design system performs physical synthesis in a variety of ways, ranging from the automated layout of standard cells to the automatic generation of Programmable Logic Arrays [5]. In an effort to accurately quantify the para-

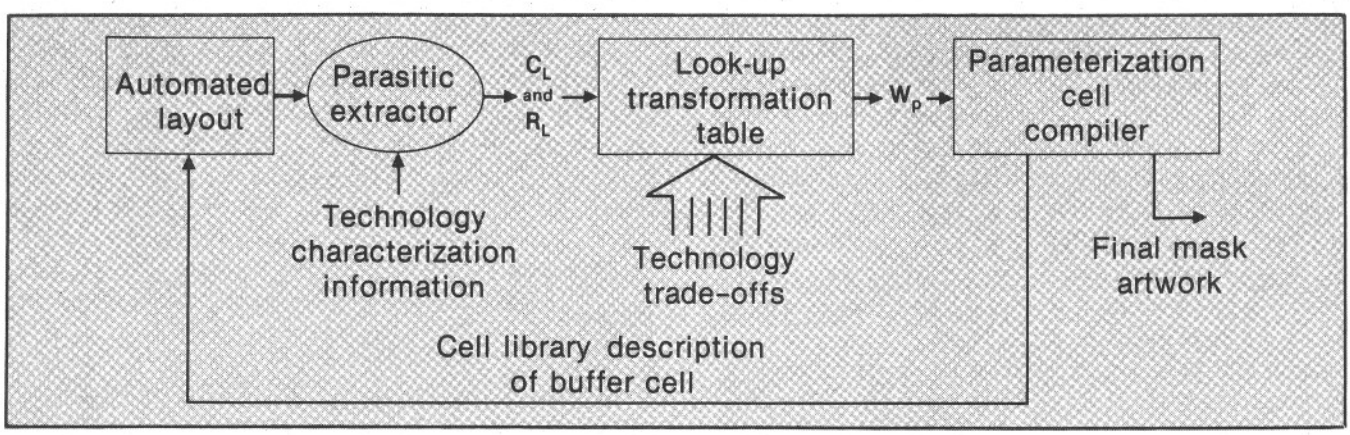


Fig. 2 Flow diagram of parameterized buffer cell system.

sitic interconnect impedances generated from the automated layout of standard cells, a specific extraction module, the Parasitic Extractor, has been written. This program automatically derives, in a precise non-statistical manner, the capacitive and resistive parasitic impedances inherent to automated, two-layer wiring.

The Parasitic Extractor uses the actual physical layout generated by an internally developed layout system, HAL (Hughes Automated Layout) [11]–[12], to automatically calculate the parasitic resistive and other portions of the physical synthesis process. Since this program is technology-independent, it has general applicability to a varied set of technologies. The program uses a special technology-dependent file built to characterize a given technology's parasitic impedances.

As depicted in Fig. 2, once the Parasitic Extractor is in place to generate information used to optimize the design of a parameterized buffer cell, it is then necessary to transform the loading information into a format compatible with the parameterized cell compiler.

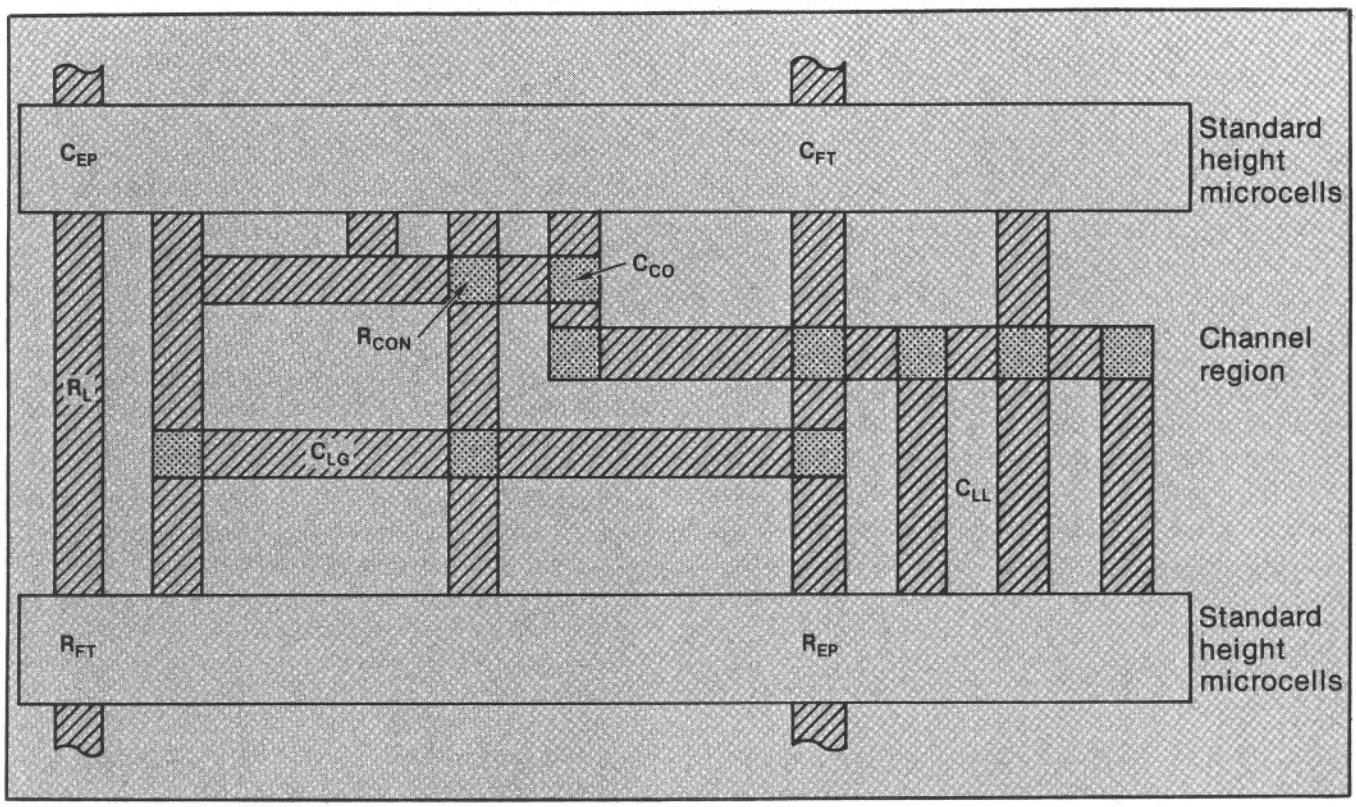


Fig. 3 Extracted parasitics for each net.

capacitive impedances associated with each HAL generated signal. As depicted in Fig. 3, all layout-dependent loads such as line-to-ground, line-to-line, crossover capacitances, feedthrough resistance and capacitance, equivalent pin resistance and capacitance, line resistance, and contact resistance are calculated. In addition, the program accurately derives the effect of a given net's distance to its next nearest neighbor on line-to-line capacitance. At specific intervals along the horizontal segments of the net, the number of empty neighbor tracks to either side of the net being analyzed are counted and a next nearest neighbor factor is used to modify the line-to-line capacitance over this interval.

Once all the resistive and capacitive parasitic components of a given set of nets have been generated, they are tabulated and presented to the design system for use in performance analysis and to constrain

The optimal P-channel device width of a single stage CMOS inverter is dependent upon two constraints: the signal loading and the performance requirements of the critical signal path. Since both these constraints are application-dependent, it is difficult to algorithmically characterize a general relationship between signal loading and the propagation delay through a single stage buffer. Thus, in order to maximize the generality of this design system, a look-up transformation table characterizing a variety of design trade-offs is necessary. Depending upon the precise requirements of a particular circuit application and its on-chip loading, an optimal P-channel device width can be chosen to parameterize the buffer cells driving the critical signal nets.

This look-up transformation table is characterized by different power/speed/area trade-offs as a function of signal net loading. A look-up transformation table

was selected as the most appropriate technique for the parasitic impedance feedback mechanism inherent to the parameterized buffer cell system. It must be emphasized, however, that for different feedback mechanisms, different transformation techniques may be preferable.

The resistive and capacitive parasitic interconnect impedances coupled with the fan-out and fan-in cell-dependent impedances define the required load that must be driven by an optimally sized buffer cell. Therefore, the on-chip performance, specified by system requirements, defines precisely the current necessary to drive the signal interconnect load in the required amount of time. Depending upon the signal loads being driven, a certain-sized buffer cell will minimize the overall path delay. The literature [13]–[14] describes in detail optimal inverter configurations and width ratios, which minimize propagation delay through a series of cascaded inverters. However, for a single stage and a given load, only the application specific requirements determine the precise device width necessary to minimize path delay. Therefore, the ability to automatically define a buffer cell as a function of its current drive has important applications to VLSI circuits where many buffers are needed to provide optimal current drive.

In a simple CMOS inverter, the geometric widths of the P- and N-channel devices can be used to define or parameterize the buffer current drive. Since the relative transconductances of the P- and N-channel devices are constant for a particular process technology, the P-channel and N-channel device widths are ratioed. Therefore, the current drive of a buffer cell can be optimized by the use of a single parameter, the P-channel geometric width.

Figure 4 depicts the layout of the required CMOS

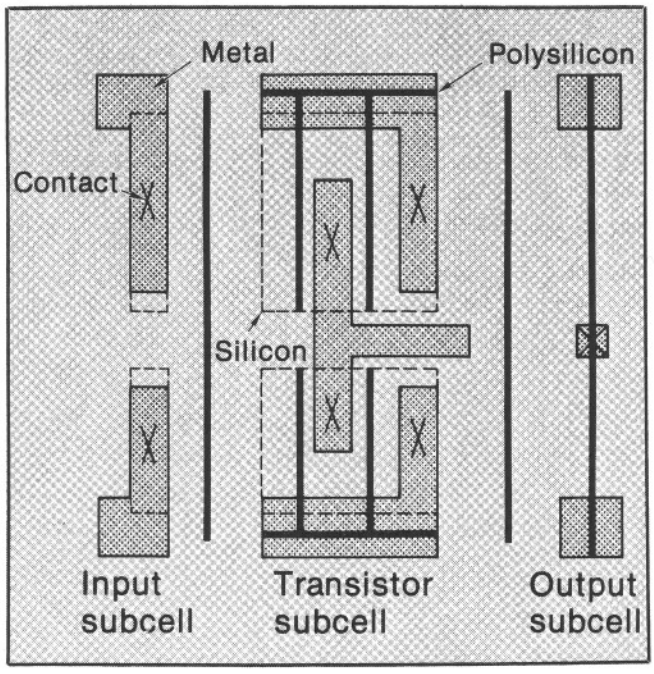


Fig. 4 Subcells of parameterized buffer cell system.

subcells of the parameterized buffer cell system. For any given P-channel width, the same input and output subcells are used. As shown, each transistor subcell contains two polysilicon stripes to maximize current density per area. The power and ground buses as well as the common P- and N-channel drains are shared by parallel transistors. The polysilicon gate has been placed on both the top and bottom of all the transistor cells for maximum flexibility in automated layout. The initial and final N-channel and P-channel polysilicon gates are connected by both the input cell and the output cell to minimize polysilicon resistance.

The cell height as well as the power and ground buses are of standard dimension so as to be compatible with the buffer cell's own cell library. Every parameterized buffer cell is made up of three significantly different subcells. Butted to the left and right portions of the buffer cell are the input and output subcells, respectively. A cascaded series of transistor subcells is placed between these two subcells. For a given set of technology-dependent design rules and a standard cell height, layout-dependent maximum and minimum transistor widths are defined. Within this range of P-channel transistor widths, additional transistor subcells have been designed. Depending upon the required P-channel width, the parameterization program connects the proper transistor subcells together and places them between the input and output subcells.

In addition to generating the final geometric artwork of the parameterized buffer cells, a cell library description of the buffer cell is generated, which defines the buffer cell size, pin locations, pin types, and pin equalities for use in HAL, as shown in Fig. 2. Therefore, depending upon the required width, a buffer cell can be quickly and optimally designed in mask defined geometries as well as for immediate use in automated layout.

Conclusions

As the requirements of VLSI-oriented silicon compilers become greater, it will be increasingly important to extract useful information from the design process and transform this information into an appropriate format that will optimize the synthesis process. Feedback in silicon compilers is composed of three significant phases: (1) extraction of accurate application specific information, (2) transformation of that data into a compatible format, and (3) parameterization of the functional cells based upon constraining application specific information. Depending upon the feedback mechanism, various transformation techniques are preferable when optimizing the physical implementation process. The appropriate transformation technique is dependent upon the particular extracted information and the resulting manner in which the functional cell utilizes the information. Examples of different extraction, transformation, and parameterization techniques as well as gen-

eral requirements for their implementation have been described.

Finally, an example of providing feedback in silicon compilers was discussed. Specifically, the feedback mechanism for extracting accurate layout-dependent interconnect parasitics and the use of this information to choose an optimal device width for a CMOS buffer cell has been described. The immediate benefits of utilizing extracted interconnect parasitics to automatically parameterize a buffer cell's circuit density and performance depict only a portion of the value of providing feedback in silicon compilers.

Acknowledgments

The author wishes to thank G. Persky, B. Valdez, and G. Kramer for their helpful discussions and encouragement and S. Powell, E. Iodice, and W. Marking for their technical assistance in the cited example as well as M. Carchidi and L. Friedman for their assistance in proofreading this paper.

References

[1] P. B. Denyer, D. Renshaw, and N. Bergmann, "A Silicon Compiler for VLSI Signal Processors," European Solid-State Circuits Conference (ESSCIRC), pp. 215–218, Sept. 1982.

[2] S. C. Johnson, "VLSI Circuit Design Reaches the Level of Architectural Description," *Electronics*, pp. 121–128, May 3, 1984.

[3] VLSI DESIGN Staff, "Silicon Compilers—Part 1: Drawing a Blank," VLSI DESIGN, pp. 54–58, Sept. 1984.

[4] B. W. Fung, T. J. Macnee, and G. Rugila, "A High-Density PLA Macro and Its Layout Generator," International Solid-State Conference (ISSCC), pp. 58–59, Feb. 1982.

[5] S. Powell, E. Iodice, and E. Friedman, "An Automated, Low Power, High Speed, Complementary PLA Design System for VLSI Applications," International Conference on Computer Design (ICCD), pp. 314–319, Oct. 1984.

[6] G. DeMicheli and A. Sangiovanni-Vincentelli. "Computer-Aided Synthesis of PLA Based Finite State Machines," International Conference on Computer-Aided Design (ICCAD), pp. 154–156, Sept. 1983.

[7] M. J. Meyer, P. Agrawal, and R. G. Pfister, "A VLSI FSM Design System," 21st Design Automation Conference, pp. 434–440, June 1984.

[8] D. Edgington, B. Walker, S. Nance, C. Starr, S. Dholakia, and M. Kliment, "CMOS Cell-Layout Compilers for Custom IC Design," Custom Integrated Circuits Conference (CICC), pp. 512–517, May 1984.

[9] A. K. Sinha, J. A. Cooper, Jr., and H. J. Levinstein, "Speed Limitations Due to Interconnect Time Constants in VLSI Integrated Circuits," *IEEE Electron Device Letters*, vol. EDL-3, No. 4, pp. 90–92, Apr. 1982.

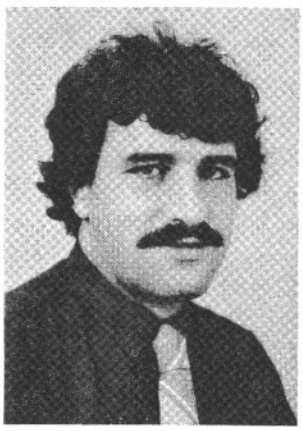
[10] E. Friedman, G. Yacoub, and S. Powell, "A Hierarchical VLSI Design System for Synthesizing CMOS/SOS Integrated Circuits," presented at SOS/SOI Technology Workshop '84, Hilton Head, South Carolina, Oct. 1984.

[11] G. Persky, C. Enger, and D. M. Selove, "The Hughes Automated Layout System—Automated LSI/VLSI Layout Based on Channel Routing," 18th Design Automation Conference, pp. 22–28, July 1981.

[12] M. Beaven, C. Enger, G. Persky, and J. Tsu, "An Automated Layout System for VLSI Design," Government Microcircuits Applications Conference (GOMAC), pp. 71–75, Nov. 1984.

[13] H. C. Lin and L. Linholm, "An Optimized Output Stage for MOS Integrated Circuits," *IEEE J. Solid-State Circuits*, vol. SC-10, no. 2, pp. 106–109, Apr. 1975.

[14] R. Jaeger, "Comments on 'An Optimized Output Stage for MOS Integrated Circuits'," *IEEE J. Solid-State Circuits* (Corresp.), vol. SC-10, no. 3, pp. 185–186, June 1975.



Eby G. Friedman

Eby G. Friedman was born in Jersey City, New Jersey, in 1957. He received the B.S. degree in electrical engineering from Lafayette College in 1979 and the M.S. degree in electrical engineering from the University of California at Irvine in 1981.

He was previously employed by Philips Gloeilampen Fabrieken in Eindhoven, The Netherlands, in 1978 and worked on the design of bipolar differential amplifiers. From 1979 to 1983, he was employed by Hughes Aircraft Company in Newport Beach, California, working in the areas of custom IC design, software compatible gate array design, one- and two-dimensional device modeling, circuit modeling, and double-level metal process development. He is currently a section head in the Industrial Electronics Group Technology Center with Hughes Aircraft Company in Carlsbad, California, responsible for custom and semicustom CMOS VLSI design and for the development of cell compiler-oriented CAD tools for IC synthesis. He is the author of several articles and presentations in the field of CMOS design techniques and CAD tools. Eby Friedman is a Member of the IEEE.