A CMOS/SOS VLSI design system*

E.G. Friedman, G. Yacoub and S. Powell
Hughes Aircraft Company, Technology Center, 6155 El Camino Real, Carlsbad, CA 92008, USA.

A VLSI design system for synthesizing CMOS/SOS integrated circuits is discussed in terms of the physical implementation techniques, simulation tools, and verification capabilities that it possesses. The integrated CAD tools utilized by the design system follow a hierarchical VLSI custom cell approach. In particular, the orientation of these CAD tools to a CMOS/SOS technology will be described.

1. Introduction

As the resolution of device and interconnect geometries decreases in size, the level of system integration increases dramatically. One significant requirement of Very Large Scale Integration (VLSI) is a cohesive design system that permits fast and efficient synthesis while maintaining accuracy and design flexibility. This system must provide extensive verification capabilities and an easily transportable database at several different levels of abstraction (e.g. functional, logic, circuit, and geometric). It must remain functionally efficient while providing quick design turnaround and it must permit easy integration of engineering changes. A hierarchical VLSI design system for synthesizing CMOS/SOS integrated circuits [1] is described that is believed to possess these traits. It utilizes a mixture of in-house generated and commercial CAD tools serving the IC design spectrum from logic capture to pattern generation tapes.

2. Overview of VLSI design system

The design system consists of a group of independent CAD tools connected by an assortment of translators transparent to the user. As shown in Fig. 1, the initial starting point of the design process is the capture of the logic design on a graphical data entry system. These logic schematics can include macrocells described solely by their Boolean or truth table functional equivalent. A commercial logic capture program, TEGATE of CALMA Company, graphically describes the chip logic functionally and connectivity in a digitized format. Once captured, connectivity and instantiation equations are generated. These netlist equations are described in TDL, TEGAS Design Language. The TDL format is compatible with in-house automated layout and PLA design systems as well as a commercial logic simulator and test vector generation system. This approach of using the same design equations for physical synthesis, simulation, and test vector generation maintains a continuity of database and establishes a correct-by-construction theme which permeates the entire design process.

*This paper is based on a presentation given at the SOS/SOI Technology Workshop, Hilton Head, South Carolina, in October 1984 by the authors.

2.1 TDL hierarchical expander

Often, these netlist design equations are not hierarchically captured for optimal layout efficiency. The criteria used to define the hierarchical module boundaries of the logic design of a circuit may not coincide with the optimal hierarchy of the physical design of the circuit. During logic design, the hierarchy for the TEGATE generated TDL netlists is defined for logical clarity and simulation ease, giving little emphasis to layout efficiency. Maintaining the logically defined hierarchical partitioning in the physical design often results in an inefficient use of silicon area. An in-house program to redefine the hierarchy of the TDL database has been created. This program, the TDL Hierarchical Expander, was deemed necessary to maximize the efficiency of the layout organization instead of physically implementing the logically sensible, layout inefficient TDL description. It provides three different modes of operation. One mode flattens the originally defined hierarchy, the second mode permits the combination of two hierarchically equivalent macrocells and/or the inclusion of primitive microcells within higher order macrocells, and the third mode creates a higher order macrocell from a set of primitives. Thus, the user can automatically traverse a circuit's hierarchy, and repartition the chip design equations into a format that improves layout efficiency without losing hierarchical control of the chip. Therefore, the various repartitioned modules are completely unconstrained and can be designed to provide aspect ratios and pin locations that permit efficient insertion into the overall chip floorplan. In addition, the functional block floorplan can be reconfigured to shorten critical paths for improved

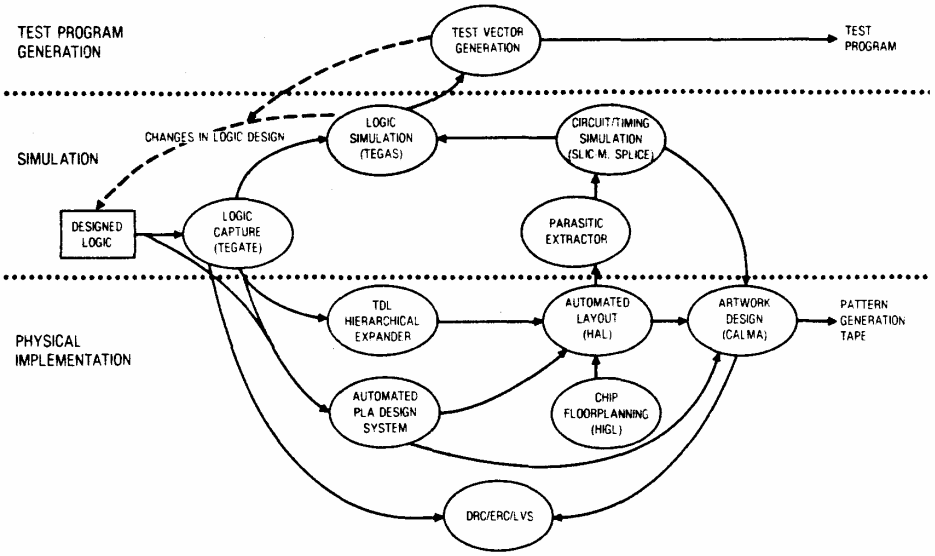


Fig. 1. Overview of the VLSI design system.

performance. The output of the TDL Hierarchical Expander program is a syntactically correct TDL description of the circuit which has the same hierarchy as the physical design. Logic simulations can be performed on the reconfigured TDL if desired.

2.2 Hughes automated layout (HAL) system

Once hierarchically reconfigured, the TDL equations are translated into a format compatible with our in-house automated layout system, HAL (Hughes Automated Layout) [2]-[4]. The netlists defined in the HAL system maintain the same hierarchy as the netlists defined in the TDL description. The HAL system consists of a one-dimensional placement (constructive initial placement with interactive capabilities) and routing program that routes standard height cells (Fig. 2), and a two-dimensional layout program capable of routing variably sized macrocells in both the X and Y directions by performing a global maze route followed by a detailed channel route (Fig. 3). In the one-dimensional routing mode [5], HAL is capable of performing automatic cell placement, cell placement optimization, layout compaction, and routing. In the two-dimensional routing mode, HAL is capable of performing variable height cell routing in both the X and Y directions. These variable height cells can either be standard cells, PLAs, or other hierarchically lower HAL generated macrocells. The flexibility of routing variable sized functional blocks two-dimensionally encourages and complements a hierarchical VLSI custom cell approach.

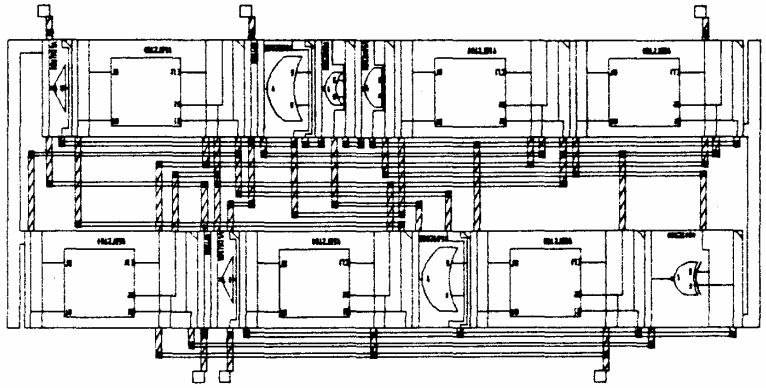


Fig. 2. HAL macrocell.

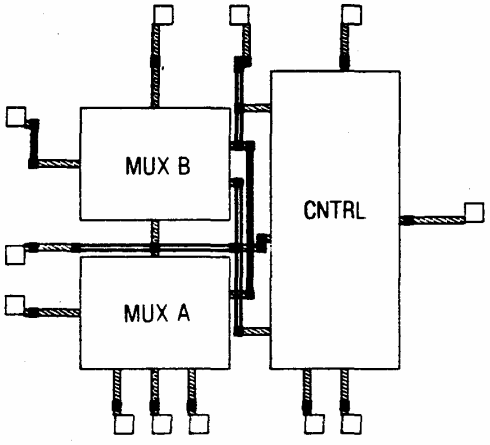


Fig. 3. HAL II macrocell.

In the one-dimensional and two-dimensional modes, HAL permits the designer the flexibility to meet a wide range of layout strategies. The initial HAL input is a file tailored by the designer to fulfill a specific circuit layout implementation. This source file consists of eight distinct subfiles. The first two describe the circuit layout architecture and design rules. The third subfile describes the HAL cell library. Cell library characteristics such as pin equalities, I/O designations, pin coordinates, and the cell perimeter are defined for every cell in the library. The fourth subfile instantiates (associates specific cell occurences with their proper library type) the application specific cells being utilised while the fifth subfile contains the HAL formatted connectivity equations. The sixth subfile defines the absolute locations of each macrocell or row of microcells while the seventh subfile permits the relative placement of any or all of the microcells within the previously defined rows. Finally, the eighth subfile provides the designer with the ability to control the routing of critical signal nets (e.g. clock lines). These eight aforementioned subfiles permit the insertion of different command options so as to customize the automated layout process.

Since HAL permits user interaction during placement to minimize track density and wirelength, an efficient layout of a high level macrocell can be quickly and easily generated. A graphics complement to the HAL system is HIGL (HAL Interactive Graphics on Lexidata terminal). It performs top-down chip floorplanning by providing high level information, which permits graphical insight into the optimization of aspect ratios, I/O locations and block locations at the functional block level. HIGL is also used to modify the placement and routing of cells within a functional block interactively, thereby improving design time.

2.3 PLA design system

Parallel to the automated standard cell and macrocell layout and performing a similar synthesis function is an in-house generated Programmable Logic Array (PLA) design system [6]. The PLA design system accepts as an input a logic description in the form of a truth table, a set of Boolean equations or mathematical expressions, or TDL from a graphically captured schematic. The PLA design system generates as an output the physical artwork of a minimised and verified PLA and provides a design capability fully compatible with the density and power constraints of VLSI.

The PLA design system is technology independent and supports many different PLA architectures. To meet the low power requirements of many VLSI applications, a very low power complementary PLA (CPLA) has been developed and incorporated into the PLA design system. The CPLA structures uses a CMOS NAND gate configuration for both the AND and OR plane so as to remove all DC current paths between power and ground. This static self-contained functional cell requires no timing signals and dissipates CMOS power levels. The close proximity of the P-channel transistors to the N-channel transistors inherent in SOS has proven instrumental in permitting the CPLA technique to be sufficiently dense for VLSI applications.

The PLA design system consists of four stages as depicted in Fig. 4. The first stage generates a complete exhaustive truth table from various input formats. Mathematical expressions are permitted as one form of input to the system. PLAs which implement addition, multiplication, trigonometric, and other mathematical functions can be generated directly from a mathematical equation. Using a logic simulation program, truth tables can be generated from simplified or unsimplified Boolean equations. Functions described by complex Boolean equations with nested parantheses, intermediate variables, and exclusive-OR operators can be generated in this manner. Additionally, if a logic schematic has been captured in a digitized format, a logic simulator can be used to generate a truth table automatically from the connectivity description. Using the technique, PLA artwork can be generated directly from a logic schematic.

The second stage of the PLA design system performs a logical minimization of the truth table generated by the first stage. Since the size of the PLA is directly dependent upon the number of product terms, it is essential that all redundant terms are removed. The output of the minimizer is in the same format as the truth table with 'don't care' states signified by a '-'. At this point in the design process, the minimized truth table is compared to the original truth table. The verification program accepts a minimised truth table as an input and recreates an exhaustive truth table as an output. This complete truth table is then automatically checked against the original truth table to verify proper minimization. The verification program also permits a manual logic minimization to be verified so that the first two stages of the PLA design system can be bypassed, permitting direct entry into the next stage of the design system. The final two stages of the PLA design system generate the PLA artwork from the fully verified and minimized truth table. Every '0', '1' and '-'

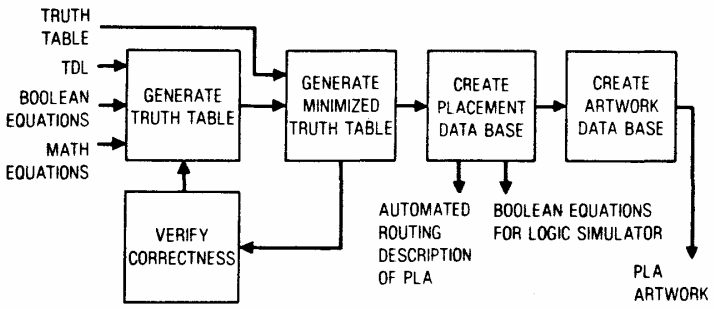


Fig. 4. PLA design system.

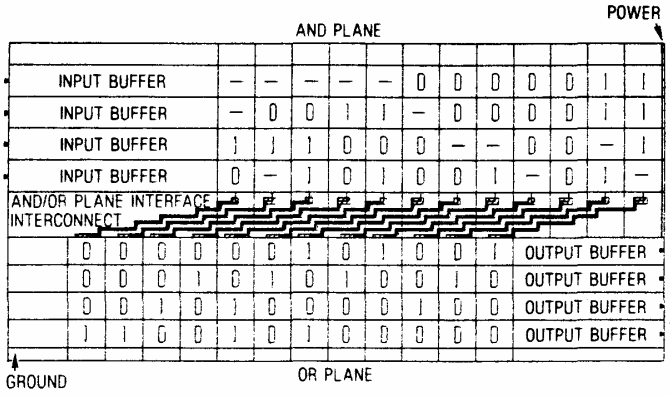


Fig. 5. PLA toplogy, with OR plane offset from AND plane.

state in the truth table references a previously created CPLA subcell. The third stage of the PLA design system reconfigures the truth table into its compatible PLA organization while the fourth stage references the appropriate subcells resident on the graphical layout system. The final product is a fully self-contained and minimized PLA.

The generated artwork includes the input and output buffers, the NAND arrays with appropriately sized transistors, the AND/OR plane connections, and the power and ground routing as depicted in Fig. 5. The design system automatically generates a set of Boolean equations that are used in the logic analyzer program, TEGAS (a product of Calma company), to permit the PLA to be simulated logically. In addition to the artwork, PLA boundary and pin location information is automatically generated in a form suitable for higher level routing. Processing speeds for the PLA range from 20 minutes to one hour of user time to generate PLA artwork from the various input options.

Thus, the PLA design system works in concert with the automated layout system to optimally synthesize logic into mask artwork. Highly random combinatorial functional blocks are generated from the PLA design system while sequentially oriented functions are implemented with HAL. Therefore, both density and automation are efficiently maintained, permitting the design to be optimal and error free.

2.4 Graphical layout system

All final artwork is deposited on a CALMA GDS II turnkey design system. Once on the CALMA system, the fully characterized CMOS/SOS microcells, macrocells, and PLA subcells are incorporated into the chip layout. If necessary, top level functional blocks are combined and interconnected and extraneous cells (e.g. chip number, company logo, etc.) are added to the chip layout. After final verification, pattern generation (PG) tapes are manufactured.

3. Design verification

Verification and simulation tools are utilised to analyze the accuracy of the circuit design. The identical TDL description used as an input to the TDL Hierarchical Expander and HAL is also compatible with the logic simulation program, TEGAS, the PLA design system, and the test vector generation system. The use of equivalent netlist design equations maintains the database continuity necessary for the successful design

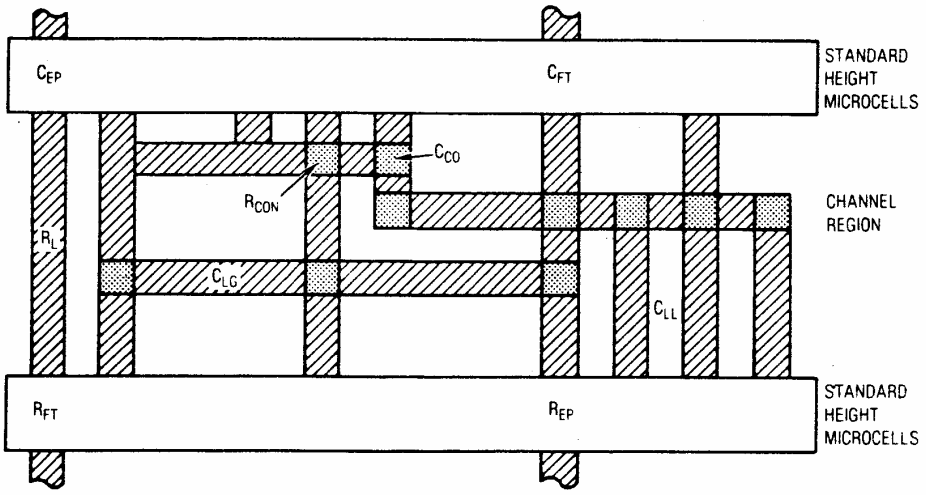


Fig. 6. Extracted parasitics for each net.

and test of a VLSI circuit. In addition, commercial software is used for geometric, electrical, and connectivity checking. This software is capable of making a direct comparison between mask level geometries and the original TDL. Thus, a correct-by-construction design theme is followed by a layout-to-logic verification check to ensure correctness.

3.1 Parasitic extraction of HAL generated interconnect

Figure 6 illustrates the main components of parasitic capacitances in CMOS/SOS interconnect structures. Since interconnect lines in a routing channel in a CMOS/SOS technology run on a sapphire insulator, the line-to-line capacitive component can be a significant portion of the total line capacitive loading due to the thickness of the sapphire. The line-to-line capacitance varies according to the proximity of neighbouring lines (referred to as the next nearest neighbour effect).

A CAD tool which automatically extracts the parasitic capacitive and resistive interconnect impedances associated with HAL-generated wiring has been developed. This program, the Parasitic Extractor program [7], uses the actual physical layout as an input and accurately accounts for capacitive coupling between multilevel crossovers and next nearest neighbour effects. The program extracts its parasitic information from the HAL database. Since the HAL system uses a horizontal and vertical fixed grid organization, each track within a channel is separated by one or more grids. The total line-to-line capacitance of a given net is dependent upon the distance to the nearest occupied grid on both sides of the signal net.

Specific CMOS/SOS technology information is provided to the program via a technology file to characterise accurately the parasitic impedances. The program extracts crossover capacitance, equivalent pin resistance and capacitance, feedthrough resistance and capacitance, contact resistance, line resistance, line-to-ground capacitance, and line-to-line capacitance. In order to accurately extract the particular SOS trait of relatively large line-to-line capacitance, the Parasitic Extractor program contains a next nearest neighbour algorithm. At specific intervals along the horizontal segments of the net, the number of free neighbour tracks to either side are counted and a next nearest neighbour factor modifies the capacitance of the segment along the net. Once all of these interconnect parasites are extracted, they are annotated into the circuit simulation files so as to permit accurate critical path analysis.

3.2 Timing and logic analysis

To analyse electrical delays and worst case paths, two different circuit simulation programs, SLIC-M and SPLICE, are used. SLIC-M, a product of Signetics Corporation, is used for detailed analysis of a given worst case path, while SPLICE, a product of the University of California at Berkeley, performs mixed mode simulation and electrical analysis of large circuits at a computationally efficient rate (1 to 2 orders of magnitude better than SPICE). TEGAS is used for gross timing analysis and general logic simulation. Together, these CAD simulation tools verify both the logical and electrical performance constraints necessary to completely satisfy system requirements.

3.3. Geometric, electrical, and connectivity checking

VSLI circuits contain large amounts of detailed information. In order to systematically verify the validity of this information absolutely and relatively, software tools are necessary to automatically perform geometric verification (i.e. Design Rule Checking (DRC)), electrical verification (i.e. Electrical Rule Checking (ERC)) and layout versus schematic (LVS) verification. A commercial software package from ECAD Inc. is used to perform these tasks.

The DRC and ERC software are capable of

selectively checking isolated windows of a hierarchical VLSI circuit. Multiple runs on selected layers and command checks eliminate the need for fracturing unnecessary layers when debugging errors. DRC and ERC must be completed before LVS checking is begun. The LVS software fractures the hierarchical mask layout into a transistor nodal equivalent while reducing the hierarchical TDL netlist into an equivalent SPICE transistor nodal description. It then performs a detailed comparison between nodal tables to check for differences between the TDL connectivity description and the final geometric artwork.

4. Testability concerns

Chip database continuity is also preserved during test vector generation. As exemplified in Fig. 1, test vectors are generated from the same set of TDL netlist equations that were used in HAL, the PLA design system, and in TEGAS, and are automatically translated into SENTRY compatible format for functional testing.

To improve the overall testability of the chip design, special Design for Testability (DFT) techniques are included in the original logic design. Options such as Set/Scan, LSSD, boundary latches, and BILBO techniques [8] all exemplify various density/exhaustive testability/testing time tradeoffs that must be made early in the design cycle. Once decided, the proper DFT technique can be implemented by using DFT-oriented CMOS/SOS microcells and the corresponding testable logic. Examples of specially designed CMOS/SOS microcells which contain different testability techniques include special flip flops with upfront multiplexers to permit a set/scan operational mode, and BILBO logic cells each of which consist of a flip flop, an AND gate, an OR gate, and an Exclusive-OR gate which permit three different modes of operation, namely data transfer, shift register, and signature analysis.

5. Conclusion

This paper describes a CMOS/SOS oriented, hierarchical design system which quickly and accurately generates VLSI circuits. Various in-house and commercial CAD tools have been integrated with special purpose translators to provide an overall VLSI design capability that is completely independent of manual assistance. Functional design information can be implemented into physical mask artwork using a variety of techniques ranging from standard cell and macrocell automated layout to programmable logic arrays. Special verification and simulation tools are used to satisfy the accuracy of the chip design. DFT techniques are embedded within the microcell library for improved testability. These CAD tools, integrated with properly optimized microcells, macrocells, and PLAs form a hierarchical VLSI design system that provides flexibility in synthesis while maintaining accurate verification and simulation capabilities.

6. Acknowledgements

The authors wish to thank the many people who assisted in developing and maintaining the VSLI design system described within this paper.

7. References

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

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

[3] Persky, G. and Tran, L. V., 'Topological Routing of Multi-Bit Data Buses', Proc. IEEE 21st Design Automation Conference, June 1984.

[4] Beaven, M., Enger, C., Persky, G. and Tsu, J., 'An Automated Layout System for VLSI Design', Government Microcircuit Applications Conference (GOMAC), pp. 71–75, November 1984.

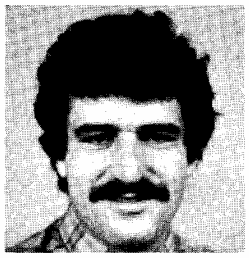
[5] Feller, A., Noto, R. and Smith, A., 'Standard Cell Approach for Generating Custom CMOS/SOS Devices Using a Fully Automatic Layout Program', IEEE Circuits and Systems, Vol. 3, No. 3, pp. 9–13, September 1981.

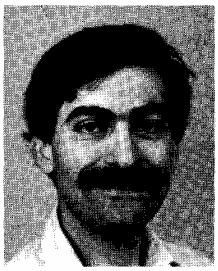
[6] Powell, S., Iodice, E. and Friedman, E., 'An Automated, Low Power, High Speed Complementary PLA Design System for VLSI Applications', Microelectronics Journal, Vol. 15, No. 4, pp. 47–54, July/August 1984.

[7] Friedman, E., Marking, W., Iodice, E. and Powell, S., 'Parametrised Buffer Cells Integrated into an Automated Layout System', Proc. IEEE Custom Integrated Circuits Conference (CICC), May 1985.

[8] Williams, T. and Parker, K., 'Design for Testability — A Survey', IEEE Trans. Comput., Vol. C. 31, pp. 2–15, January 1982.

E. G. Friedman received his 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 area 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-oritented CAD tools for IC synthesis. He has authored many articles and presentations in the fields of CMOS design techniques and CAD tools.





G. Yacoub received his B.S. degree in electrical engineering from the University of California at San Diego in 1980 and the M.S. degree in electrical engineering from the University of California at Irvine in 1981. Since 1981, he has been employed by Hughes Aircraft Company working in the areas of IC processing, two-dimensional device modeling, gate-array master-slice design, custom IC design, and structured VLSI design. He is currently a group head in the Industrial Electronics Group Technology Center with Hughes Aircraft Company in Carlsbad, California working in the areas of VLSI circuit design and VLSI CAD system research and development. He has co-authored several technical papers in the field of IC design.

S. Powell received his B.S. degree and M.S. degree in electrical engineering from Oregon State University in 1982 and 1983, respectively. He was previously employed by Hewlett Packard and Tektronix working in the areas of analog and digital circuit design, LSI testing, and CAD development. He is currently a group head at Hughes Aircraft Company in Carlsbad, California, where he is involved in the research of VLSI circuit design techniques and CAD tools. He has authored several papers in the field of digital VLSI circuit design.