VLSI Design
1993, Vol. 1, No. 1, pp. 1–7
Reprints available directly from the publisher
Photocopying permitted by license only

Overlapped Subarray Segmentation: an Efficient Test Method for Cellular Arrays

EARL E. SWARTZLANDER, JR. and MIROSLAW MALEK
Department of Electrical and Computer Engineering, University of Texas, Austin, Texas, USA

This paper presents a new test approach that is suitable for repetitive structures such as cellular arrays. As such, it is directly applicable to most arithmetic circuits which are generally quite regular. It is based on exhaustive testing of overlapping segments of the array. This approach detects bridging faults in addition to stuck-at faults. Such bridging faults are a significant problem in arithmetic circuits. The ability to detect arbitrary faults involving any cells within a designer selected distance (i.e., the diameter of the subset that is exhaustively tested) is unique to this testing approach. The high coverage of the proposed technique makes it attractive for testing current VLSI and future WSI arrays.

Key Words: *Cellular array; Bridging faults; Fault models; Subarray segmentation; VLSI/WSI Array testing*

With the ever growing complexity of computer systems, test design and implementation has become a major stumbling block in the quest to develop and produce complex circuits at a reasonable cost. Current test generation algorithms [1]–[4] are confined to heuristics and require a prohibitive amount of design effort and test time for current VLSI circuits, even for very limited stuck logic level fault models (where faults are assumed to cause a signal line to be stuck at either a logic ONE or a logic ZERO). Testing is currently recognized as a very difficult problem for VLSI, and is expected to become even more of a problem in the future for WSI and ULSI due to the trend toward increased logic complexity and limited package pin counts.

It has been known for some time [5]–[13] that integrated circuits may display bridging faults where one cell incorrectly affects the logic values in a different (but generally physically adjacent) cell. Early work on bridging faults [5] makes the assumption that bridging faults occur where two or more leads in a logic network are connected accidentally and that wired logic (i.e., either a wired OR a wired AND depending on the circuit technology) is performed at the connection. In CMOS bridging faults may not create an unwanted wired OR or wired AND gate, but may produce a signal that causes both the nMOS and pMOS networks in a gate to conduct simultaneously. The resulting signal level may be interpreted by succeeding circuits as a logic ZERO, a logic ONE, or both. Such behavior may be detected by monitoring the circuit current while testing [10], [11].

It was reported by Shen et al. [12] that in the design of an array multiplier, "the most prominent fault type is the bridging faults (30 percent)." Array multiplier production experience confirms the existence of bridging faults between adjacent cells in both bipolar [13] and CMOS implementations.

Although the problem of detecting bridging faults might seem to be similar to the problem of detecting pattern sensitive faults in memories [14], there is greater similarity between bridging faults in logic arrays and coupling faults in memories [15].

This paper follows conventions established in McCluskey [16] by assuming the cells to be combinational and further assuming that they remain combinational when faulty so that exhaustive testing of a group of cells will detect any bridging faults between cells in that group. Stuck open and stuck closed faults will be detected if they produce combinational errors. The basic approach is to partition the logic array into subarrays that can be tested exhaustively. Unlike the approach in McCluskey and Bozorgui-Nesbat [17], we employ overlapping sub-

arrays so that bridging faults at the edges of subarrays are detected.

Most conventional testing methods which are designed to detect stuck logic levels fail to detect bridging faults. This paper introduces a testing technique called *overlapped subarray testing,* which employs exhaustive testing of overlapping subarrays of physically adjacent cells. This approach is capable of detecting most static fault types including bridging faults. Formulas giving the number of subarrays that must be tested for arbitrary array and segment sizes are derived. Overlapped subarray testing is directly applicable to "bit-slice" structures that are frequently employed in realizing arithmetic functions. An example of applying this approach to the testing of a fully parallel array multiplier is given.

After introducing our notation, definitions, and the model, the proposed testing method is described, and the formula for the test length as a function of array and subarray sizes and subarray test length is derived. Next, some practical issues are discussed and the effectiveness of the proposed technique is demonstrated by applying it to an array multiplier.

DEFINITIONS

The following basic concepts underlie the overlapped subarray segmentation approach:

A *cell* is a combinational logic block that is replicated in one or more dimensions with regular interconnection to create an *array*.

An *intercell bridging fault* is a fault where one cell affects another cell to produce an erroneous output.

Bridging span, Span (x, y) is the largest (x, y) distance between two cells which may interact causing a bridging fault.

Segmentation is the process of dividing an array into subarrays. A segment $P_i(x, y)$ of an array is a rectangular subarray of x by y contiguous cells. Two segments, $P_i(x, y)$ and $P_j(x, y)$, overlap if they have one or more cells in common.

A *test vector* is a pattern of inputs applied to a circuit to determine the presence or absence of faults.

A *cell test* is an application of a b-bit test vector to the b inputs of a cell and comparison of the cell outputs with the correct outputs.

An *exhaustive cell test* is the application of all 2^b distinct cell tests to a cell.

A *segment test* is the application of an input test vector to a segment of size x cells by y cells and comparison of the cells' outputs with the correct outputs.

An *exhaustive segment test* is the application of all distinct input test vectors to a segment of size x by y and comparison with the correct outputs. For a segment of n_x by n_y cells with b inputs, the segment test will include no more than 2^{bxy} tests.

The *test coverage* is the fraction of a given class of faults that is detected by specific test. It is assumed that an exhaustive test achieves 100% test coverage of the combinational faults.

NOTATION

N is the total number of cells.

n_x, n_y is the number of cells in the array along X and Y coordinates, respectively.

(x, y) is the segment size.

$P_i(x, y)$ is the i-th segment of size x cells by y cells.

$O(P_i, P_j)$ is the region common to segments $P_i(x, y)$ and $P_j(x, y)$.

$p(x, y)$ is the number of segments of size x cells by y cells.

$O_x(P_i, P_j)$ is the number of elements in one row of $O(P_i, P_j)$.

$O_y(P_i, P_j)$ is the number of elements in one column of $O(P_i, P_j)$.

$t(x, y)$ is the number of test vectors required to test an x cell by y cell segment. When there are m inputs for the x cell by y cell segment $t(x, y) = 2^m$.

$t[x, y, o_x, o_y]$ is the number of test vectors required to test the entire array for x cell by y cell segments with incremental overlap o_x, o_y.

Figure 1 illustrates an example of overlapping segments. Here $P_i(2, 3)$ and $P_j(2, 3)$ overlap in $O(P_i, P_j)$, a one cell by three cell region. Thus, $O_x(P_i, P_j)$ is one and $O_y(P_i, P_j)$ is three. For this example $p(x, y)$ is six.

THE FAULT MODEL

The general fault model, introduced here, assumes that any fault within a segment $P_i(x, y)$ can be detected by applying a segment test. This is done by applying an exhaustive segment test that applies all 2^n input patterns on the n inputs to the segment. This test achieves 100% coverage of the segment for stuck faults and bridging faults (note that a current monitor

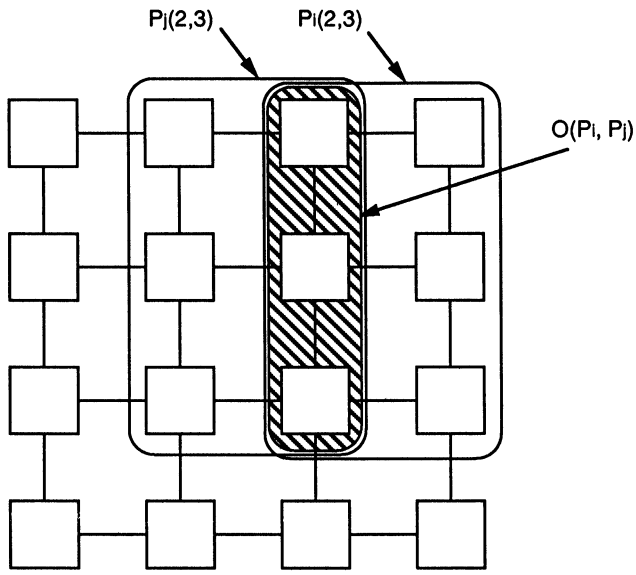

FIGURE 1 Illustration of Overlapping Segments of a Mesh.

capability may be needed to detect bridging faults in CMOS).

In general, it is desirable to apply an exhaustive test to the smallest segment for which bridging faults have been observed. Often bridging occurs between adjacent cells or between adjacent layers of metallization [18] and a 2 by 2 cell segment will be adequate. Overlapping segments of size (x, y) may replace segments of larger size without sacrifice in test coverage as long as all bridging faults have spans of less than x, y.

OVERLAPPED SUBARRAY TESTING OF 2-DIMENSIONAL ARRAYS

For a two-dimensional cellular array, the overlap process can be viewed as successive tiling with incremental displacement in a raster scan format which is analogous to beam motion for a video display. The incremental x-overlap $o_x = O_x(P_i, P_{i+1})$ is the overlap between adjacent segments within a row along the X axis. Similarly, the incremental y-overlap o_y is the overlap between adjacent segments within a column along the Y axis. This section provides a procedure to optimize the array test length $t[x, y, o_x, o_y]$ with respect to a given Span(x, y).

Span(x, y) is a crucial design parameter that may be assumed a priori or derived through experience. For logic arrays bridging has been observed primarily between adjacent cells. Certainly technology, feature size and layout geometry can provide valuable guidance for an initial Span(x, y) which may be refined based on observation of faulty circuits.

The total number of tests required for a given circuit (system) is a function of the length of the segment test and the number of segments. The length of the segment test increases as the segment becomes larger. The number of segments decreases as the segment size increases and increases as the amount of overlap increases.

The number of distinct segments of width x within a row is $p(x)$,

$$p(x) = \left\lceil \frac{n_x - o_x}{x - o_x} \right\rceil$$

Similarly the number of distinct segments of height y within a column is $p(y)$,

$$p(y) = \left\lceil \frac{n_y - o_y}{y - o_y} \right\rceil$$

Then the total number of segments $p(x, y)$ for the two-dimensional array is

$$p(x, y) = p(x)p(y) = \left\lceil \frac{n_x - o_x}{x - o_x} \right\rceil \left\lceil \frac{n_y - o_y}{y - o_y} \right\rceil$$

Since the number of test vectors required to test a segment is $t(x, y)$, the total number of test vectors for a comprehensive test is equal to

$$t[x, y, o_x, o_y] = p(x, y)t(x, y) = \left\lceil \frac{n_x - o_x}{x - o_x} \right\rceil \left\lceil \frac{n_y - o_y}{y - o_y} \right\rceil t(x, y) \quad (1)$$

Since $t(x, y)$, the number of tests for an x cell by y cell (with b inputs per cell) segment, is less than or equal to 2^{bxy} (for an exhaustive test), the maximum number of tests is bounded by

$$t_{\max}(x, y) = p(x, y)\, 2^{bxy} \quad (2)$$

Equations (1) and (2) allow us to construct a test which requires fewer test vectors than naive exhaustive test of the entire array. Of course, testing using segments of size (x, y) cannot detect bridging faults with Span(a, b) with $a > x$ or $b > y$. Since exhaustive test of each x by y segment is assumed to give full coverage of that segment, overlapping subarrays with $(o_x, o_y) = (x - 1, y - 1)$ will give full coverage of the array with respect to Span(a, b) with $a \leq x$ and $b \leq y$.

In some situations it may be possible to reduce the number of required test vectors by minimizing the overlap of adjacent (both horizontally and vertically) segments. If for example a four by four cell segment is used, displacing adjacent segments by two cells will reduce the test vector count by a factor of four. This approach of a large segment size (x, y) and minimal overlap will not detect all faults with Span(x, y).

EXAMPLES

An example is presented in this Section to illustrate the application of overlapped subarray segmentation.

Examination of a parallel multiplier illustrates the application of these proposed concepts to a two dimensional logic array. The structure of an array multiplier for a pair of N-bit positive numbers is shown by the block diagram of Figure 2 and the photomicrograph of Figure 3. It consists of N^2–$2N$ full adders, N half-adders, and $2N$–1 AND gate cells arranged in a regular rectangular array. Commercial array multipliers negate some inputs to the cells along the left and bottom edges of the array to implement two's complement arithmetic [19], but retain the basic structure consisting of a rectangular cellular array.

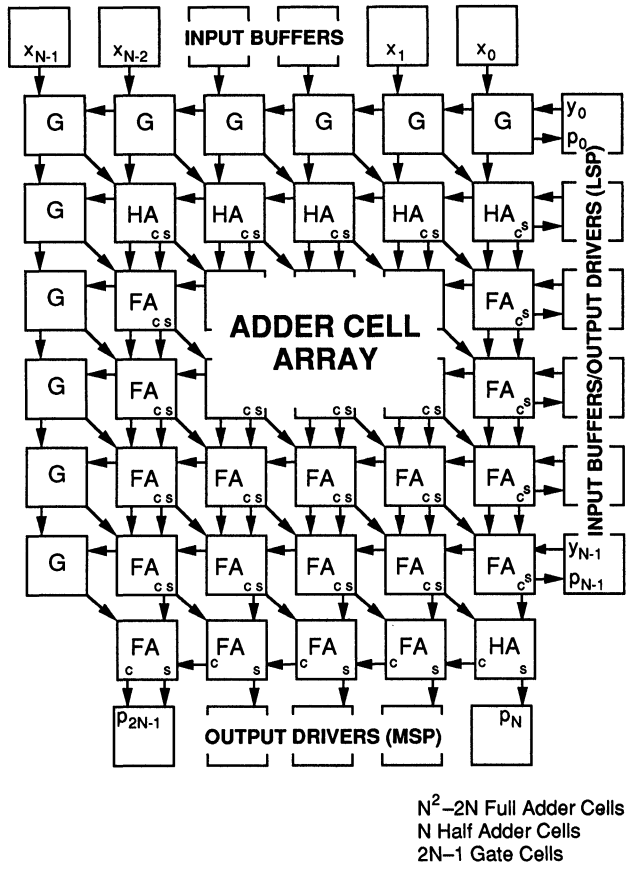

FIGURE 2 Block Diagram of Parallel Multiplier.

Exhaustively testing an N by N array multiplier requires 2^{2N+1} test vectors since there are $2N + 1$ inputs (N bits each for the two operands and one bit for the round control). Although exhaustive testing has been used for multipliers as large as 16 by 16 (where 8.6×10^9 test vectors are required), it is not feasible for larger multipliers.

An alternative approach is to exhaustively test each of the $N^2 + N$–1 cells. Because the gate cells have two inputs, four test vectors are required to exhaustively test them. Similarly, the half adder cells have three inputs and require eight test vectors and the full adder cells have four inputs and require 16 test vectors. For a 16 by 16 multiplier, 3836 test patterns are required. Actual experience indicates that production multipliers exhibit bridging faults that are not detected by testing the individual cells [7].

Bridging faults have been observed that affect adjacent (i.e., right-left or up-down) cell pairs. Accordingly, they may be detected by exhaustively testing all overlapping one by two, two by one, or two by two subsets of the array. A 16 by 16 multiplier has N^2 overlapping one cell by two cell subarrays, N^2–2 overlapping two cell by one cell subarrays, and N^2–N overlapping two cell by two cell subarrays. Because one gate input is common across rows of cells and the second gate input is common across the columns, even the two by two subarrays of gated full adders have only nine inputs. If it is assumed for simplification that all of the N^2–N overlapping two cell by two cell subarrays are comprised of gated full adders, a total of 114,688 test vectors are required. In general the number of test vectors scales as $O(N^2)$, so this approach remains reasonable even for large values of N.

The overlapping segment testing may be extended to larger segments. For example, three cell by three cell could be used, but because the number of inputs to the nine cell subarray grows (it is 14 when all the cells are gated full adders) the number of test vectors grows quite rapidly. For the commercial multipliers, bridging faults that span beyond adjacent cells have not been observed, so the vastly increased test complexity to accommodate the three cell by three cell subarray does not seem to be warranted.

Six test options for a 16 by 16 multiplier are compared in Table I. In preparing this Table, each cell is assumed to have four inputs even though the cells along the top and left edge have fewer inputs. The

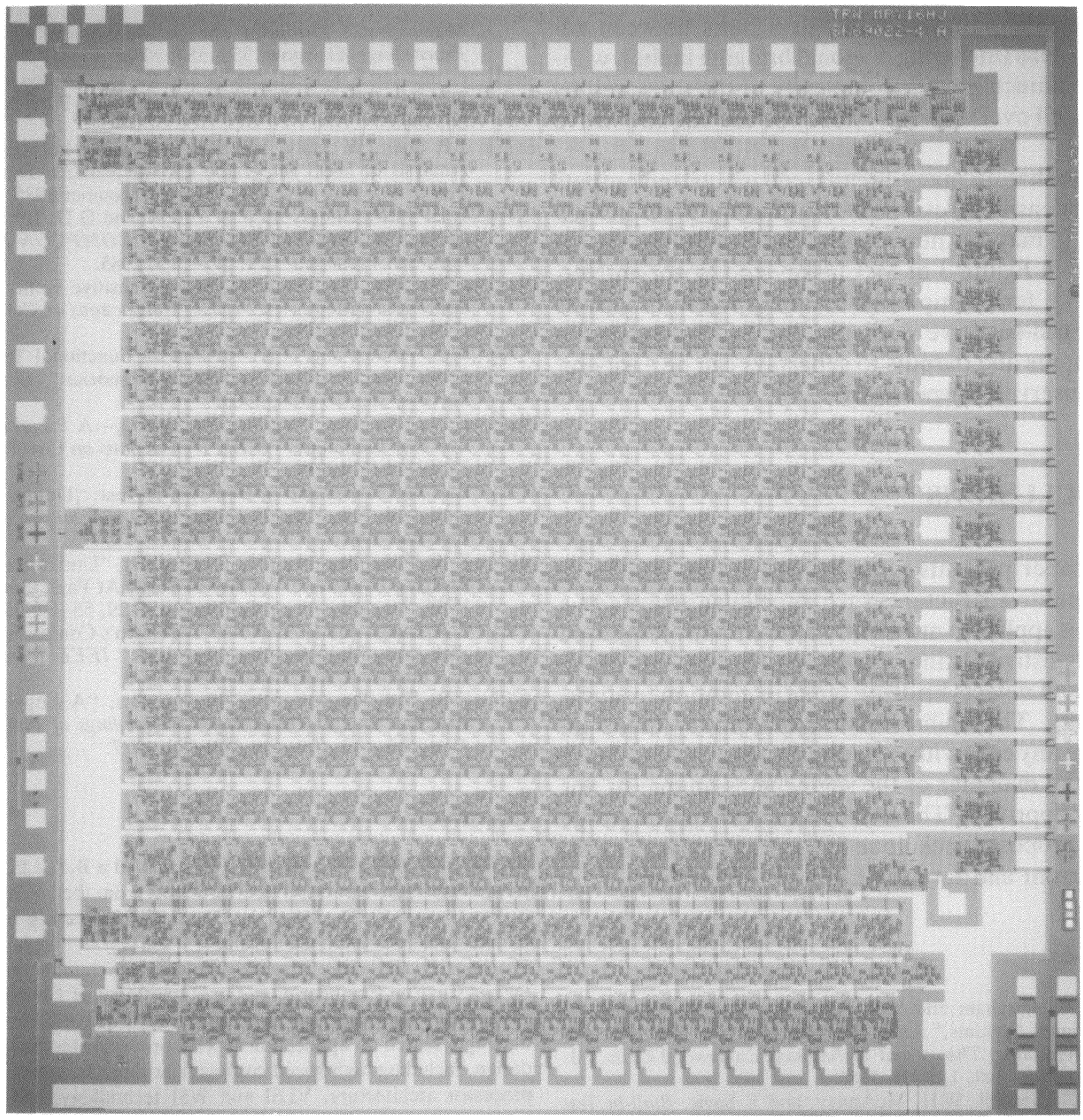

FIGURE 3 Photomicrograph of Parallel Multiplier.

one cell by one cell segment size requires only a couple of thousand test vectors, but cannot detect inter cell bridging faults. The one cell by two cell and two cell by one cell segment size require on the order of ten thousand test vectors each and will detect horizontal or vertical bridging faults between adjacent cells, respectively. The two cell by two cell segment size requires about a hundred thousand test

TABLE I
16 by 16 Multiplier Test Comparison

Segment Size	Number of Segments	Inputs/Segment*	Test Vectors/Segment*	Total Test Vectors*
1 by 1	271	4	16	4,336
1 by 2	256	6	64	16,384
2 by 1	254	7	128	32,512
2 by 2	240	9	512	122,880
3 by 3	210	14	16,384	3,440,640
16 by 16	1	33	8,589,834,592	8,589,834,592

*For all except the 16 by 16 case, the count of inputs and test vectors assume that all cells are gated full adders.

vectors and will detect bridging faults between adjacent cells (including diagonal bridging faults), without the much greater complexity of the three cell by three cell overlapping subarray or the full exhaustive tests.

Note that for many circuits, cell inputs and outputs are not easily accessible at the package pins. In such cases it may be necessary to add multiplexers (as shown in Figure 3 of McCluskey and Bozorgui-Nesbat [17], for example) or scanning registers such as level-scan-sensitive design (LSSD) [20] so that the test inputs and outputs to subarrays can be easily set and observed, respectively.

CONCLUSIONS

This paper presents a test approach that is suitable for repetitive structures such as cellular arrays. As a result it applies to most arithmetic circuits. It is based on exhaustive testing of overlapping segments of the array and detects bridging faults in addition to stuck-at faults. The ability to detect arbitrary faults involving any cells within a designer selectable distance (i.e., the diameter of the segment) is unique to this testing approach. The universality and high coverage of the proposed technique make it attractive for testing current and future VLSI and WSI arrays.

References

[1] J.A. Abraham and V.K. Agrawal, "Test Generation for Digital Systems," in D.K. Pradhan (ed.), *Fault-Tolerant Computing: Theory and Techniques*. Englewood Cliffs, NJ: Prentice-Hall, 1, 1–94, 1986.

[2] P.H. Bardell, W.H. McAnney, and J. Savir, *Built-in Test for VLSI: Pseudorandom Techniques*. New York: John Wiley and Sons, 1987.

[3] H. Fujiwara, *Logic Testing and Design for Testability*. Cambridge, MA: The MIT Press, 1985.

[4] T.W. Williams, *VLSI Testing*. Amsterdam: North-Holland, 1986.

[5] K.C.Y. Mei, "Bridge Faults and Stuck-At Faults in Two-Level Logic," *IEEE Transactions on Computers,* C-27, 452–455, 1978.

[6] A. Iosupovicz, "Optimal Detection of Bridging and Stuck-At Faults," *IEEE Transactions on Computers,* C-23, 720–727, 1974.

[7] M. Karpovsky and S.Y.H. Su, "Detection and Location of Input and Feedback Bridging Faults Among Input and Output Lines," *IEEE Transactions on Computers,* C-29, 523–527, 1980.

[8] M. Karpovsky, "Universal Tests for Detection of Input/Output Stuck-At and Bridging Faults," *IEEE Transactions on Computers,* C-32, 1194–1198, 1983.

[9] T. Yamada and T. Nanya, "Stuck-At Fault Tests in the Presence of Undetectable Bridging Faults," *IEEE Transactions on Computers,* C-33, 758–761, 1984.

[10] N. Jha and Q. Tong, "Detection of Multiple Input Bridging and Stuck-On Faults in CMOS Logic Circuits Using Current Monitoring," *European Design Automation Conference Proceedings,* Glasgow, 350–354, 1990.

[11] N. Jha and S. Kundu, *Testing and Reliable Design of CMOS Circuits*. Boston, MA: Kluwer Academic Publishers, 21–24, 1990.

[12] J.P. Shen, W. Maly, and F.J. Ferguson, "Inductive Fault Analysis of MOS Integrated Circuits," *IEEE Design and Test of Computers,* 2(12), 13–26, December 1985.

[13] E.E. Swartzlander, Jr., J.A. Eldon, and D.D. Hsu, "VLSI Testing: A Decade of Experience," *COMPCON Proceedings,* San Francisco, CA, 392–395, 1985.

[14] J.P. Hayes, "Detection of Pattern-Sensitive Faults in Random-Access Memories," *IEEE Transactions on Computers,* C-24, 150–157, 1975.

[15] M.S. Abadir and H.K. Reghbati, "Functional Testing of Semiconductor Random Access Memories," *Computing Surveys,* 15, 175–198, 1983.

[16] E.J. McCluskey, "Verification Testing—A Pseudoexhaustive Test Technique," *IEEE Transactions on Computers,* C-33, 541–546, 1984.

[17] E.J. McCluskey and S. Bozorgui-Nesbat, "Design for Autonomous Test," *IEEE Transactions on Computers,* C-30, 866–874, 1981.

[18] K.K. Kodandapani and D.K. Pradhan, "Undetectability of Bridging Faults and Validity of Stuck-At Fault Test Sets," *IEEE Transactions on Computers,* C-29, 55–59, 1980.

[19] C.R. Baugh and B.A. Wooley, "A Two's Complement Parallel Array Multiplication Algorithm," *IEEE Transactions on Computers,* C-22, 1045–1047, 1973.

[20] E.B. Eichelberger and T.W. Williams, "A Logic Design Structure for LSI Testability," *Proceedings of 14th Design Automation Conference,* 462–468, 1977.

Biographies

EARL E. SWARTZLANDER, JR. received a B.S.E.E. degree from Purdue University, M.S.E.E. degree from the University of Colorado, and Ph.D. degree in Electrical Engineering from the University of Southern California.

Professor Swartzlander presently holds the Schlumberger Centennial Chair in Engineering in the Department of Electrical and Computer Engineering at the University of Texas. He is conducting research on application specific processor technology and design, including high-speed computer arithmetic, systolic signal processor architecture, VLSI and WSI technology, and system prototyping. Previously at TRW, he managed the Independent Research and Development program for the TRW Defense Systems Group and the Digital Processing Laboratory in the TRW Electronic Systems Group. He developed the architectural and functional design of VLSI components which are in production; managed the development of the first semi custom integrated circuit with over 100,000 transistors; and managed high speed signal processor development projects including a 40 MSPS floating point FFT processor.

Dr. Swartzlander is a member of the Board of Governors for the IEEE Computer Society, Editor-in-Chief of the *IEEE Transactions on Computers,* hardware area editor of *ACM Computer Reviews,* and founding Editor-in-Chief of the *Journal of VLSI Signal Processing.* He was an Editor of the *IEEE Transactions on Parallel and Distributed Systems,* and was an Associate Editor of the *IEEE Journal of Solid-State Circuits.*

Professor Swartzlander has written the book *VLSI Signal Processing Systems* (Kluwer, 1986) and edited five books including two collections of reprints on Computer Arithmetic (IEEE Computer Society Press, 1990). He has written or co-written over 100 papers in the fields of computer arithmetic, signal processing, and VLSI implementation.

MIROSLAW MALEK received an M.Sc. degree in Electrical Engineering and Ph.D. degree in Computer Science from the Technical University of Wroclaw, Poland.

Dr. Malek is the Bettie Margaret Smith Professor in Engineering at the University of Texas at Austin. He was a visiting scholar at the Department of Systems Design at the University of Waterloo, Ontario, Canada. He works in the areas of parallel architectures, networks, testing and fault tolerance and has published over 75 papers on a variety of topics germane to these areas. He co-authored a book titled *Parallel Computing: Theory and Comparisons*. He has also participated in two parallel architecture projects: the Texas Reconfigurable Array Computer (TRAC) and IBM's Research Parallel Processor Prototype (RP3).

Professor Malek has organized, chaired and been a program committee member of numerous IEEE and ACM international conferences and workshops. Among others, he was program and general chairman of the Real-Time Systems Symposium. He serves on the editorial boards of *Journal of Parallel and Distributed Computing* and *Journal of Real-Time Systems*. He is currently on leave at the Office of Naval Research in London.