A Novel Clocking Technique for VLSI Circuit Testability

M. RAY MERCER, MEMBER, IEEE, AND VISHWANI D. AGRAWAL, SENIOR MEMBER, IEEE

Abstract—**Scan-testable digital designs have a special "scan" operating mode to set and read the states of flip-flops in the circuit. All previous scan-testable design implementations require at least one additional input pin to specify either "scan" or "normal" operating mode, and this mode specification signal must be routed to every flip-flop. A new clocking structure is described which eliminates these requirements for certain designs with static flip-flops that are controlled by two independent signals (master clock and slave clock). This is possible because, in "normal" circuit operation, the master and slave clocks are never simultaneously active. The new clocking structure uses the "all clocks active" condition to specify the scan mode. Implementation of the concept is discussed in detail for two-clock circuits. Single-clock circuits can be modified to use this scheme, and the results for this class of design are also presented.**

Introduction

THE use of scan design for testability in digital circuits is well known [1]–[3]. Apart from the normal operation of the circuit, such designs have an additional *scan* mode in which the flip-flops of the circuit are reconnected to form one or more shift registers. This mode allows setting and observation of the flip-flop states for testing. One essential requirement for all existing designs is an added input signal which switches the circuit between a scan mode and a normal mode. Depending on the layout style, this signal may add significantly to routing area on a large LSI chip since it must be supplied to each flip-flop in the circuit; the switch signal also requires one additional input pin.

In the design concept presented here, the scan switch signal is superimposed on the clock leads which must always be routed to all flip-flops for normal circuit operation. The addition of this function to clock signals is possible because of the signal redundancy that is present in the circuit clocks; in normal operation, the "all clocks active" state is never used.

This concept can easily be applied to a multiclock circuit, and would result in the saving of one input pin and the routing which would otherwise have to connect this pin to all flip-flops. However, some logic is added to each flip-flop to decode the clock signals, thus increasing the area. On balance, in LSI and VLSI circuits, long routing paths are typically considerably more expensive compared to the addition of a few local logic gates (requiring no long interconnections).

In a single-clock circuit with static flip-flops, the clock signal does not have any redundancy (both 0 and 1 states are used in normal operation), and the technique may not be directly applicable. Many single-clock circuits with static flip-flops, however, employ a clock-generating circuit that produces two nonoverlapping signals which are routed to all master–slave flip-flops. In such cases, the scan mode signal (which would require an added input pin) can be superimposed on the two signals produced by the clock generator since these signals do contain redundancy.

The main contribution of this work is a new design idea. The workability of this idea in the standard cell design environment has been demonstrated by actually designing the special flip-flop cell required for such design.

The Basic Concept

Scan design makes a digital circuit testable by allowing a scan mode in which all flip-flop data inputs are disconnected from the rest of the circuit and are reconnected to form one or more shift registers. For a design with static memory elements, this can be accomplished by adding multiplexing circuits at the flip-flop data inputs. In the following discussion, the combination of multiplexer, master latch, and slave latch will be referred to as a "scan register flip-flop." The switching of mode between the normal operation and the scan operation in the past has been affected by either a mode-select signal [1] or a scan clock signal [2]. In either case, this new signal must be routed to each flip-flop. If the master and slave clock inputs are independent, then those two signals are also routed to each flip-flop. Thus, three control signals are provided to each scan register flip-flop. These can specify up to eight different operations, but there are only four operations that have to be specified, i.e.,

1) load the master latch with normal circuit data,
2) load the master latch with scan data,
3) load the slave latch with data from the master latch,
4) maintain the current state of both latches.

Four operations can always be specified by two binary signals. Thus, it is sufficient to use just two input pins and route them to all flip-flops. By a proper manipulation of the signals on these pins, any one of the four operations listed above can be selected. Of course, a small decoding circuit will be required at each flip-flop.

Manuscript received January 10, 1983; revised August 2, 1983.
M. R. Mercer was with Bell Laboratories, Murray Hill, NJ 07974. He is now with the Department of Electrical Engineering, University of Texas, Austin, TX 78712.
V. D. Agrawal is with Bell Laboratories, Murray Hill, NJ 07974.

0018-9200/84/0400-0207$01.00 ©1984 IEEE

In the designs which use a single-clock signal to control both the master latch and the slave latch of a static flip-flop, the clock edge determines when one latch becomes active and its companion latch becomes inactive. This clocking scheme has the disadvantage that any glitch on the clock signal can produce spurious flip-flop activity. In contrast, designs which use independent clock signals for the master and the slave latches can be operated in a *level-sensitive* mode [2]. This clocking scheme has the advantage that only clock signal levels (and not signal edges) control the flip-flops. Such designs are less sensitive to clock noise, ac device characteristics, and timing problems.

The designs presented below use two-clock level-sensitive operation in the normal mode and a single-clock operation in the scan mode. This is a good compromise between the two clocking schemes since timing problems would be expected in normal mode, while the scan operation is a simple one and should have no timing problems.

Example 1—Scan Register Flip-Flop with Multiplexer

Fig. 1 shows a scan register flip-flop formed by adding a multiplexer switch at the data input of a master–slave flip-flop [1]. Normally, three control signals, mode-select signal SW, master clock MCK, and the slave clock SCK, must be routed to every scan register flip-flop in the circuit. For the purpose of this discussion, let us assume that SW = 0 feeds normal data to the flip-flop and SW = 1 feeds the scan data; also assume that the latches are active (can change state) when their clock signals are at logic one. Of the eight possible operations specified by the three control signals, two (operations 4 and 8 in Fig. 1) are not normally allowed since they will simultaneously activate both the master latch and slave latch. Operations 1, 2, and 3 are used in the normal mode, and 5, 6, and 7 are used in the scan mode. Noting that whenever the master latch is inactive (i.e., MCK = 0), the value of the signal SW can be DON'T CARE, we can collapse all the control operations into four essential operations. These are shown in the second table in Fig. 1.

The four essential operations can be specified by two signals M and S, as illustrated in Fig. 2. Here just one gate is required to decode the SW, MCK, and SCK inputs for the scan register flip-flop. Suggested waveforms for M and S are illustrated in Fig. 3. Note that the normal operation is level-sensitive as obtained by applying nonoverlapping clock signals to M and S. It is important to properly control the routing delays of M and S signals to preserve their nonoverlapping nature. An overlap of these signals (i.e., M and S being high simultaneously) can cause incorrect operation by storing scan data into the slave latch. Nonoverlapping clocks are also employed for a race-free operation of the circuit and should not be considered as an additional requirement for testability.

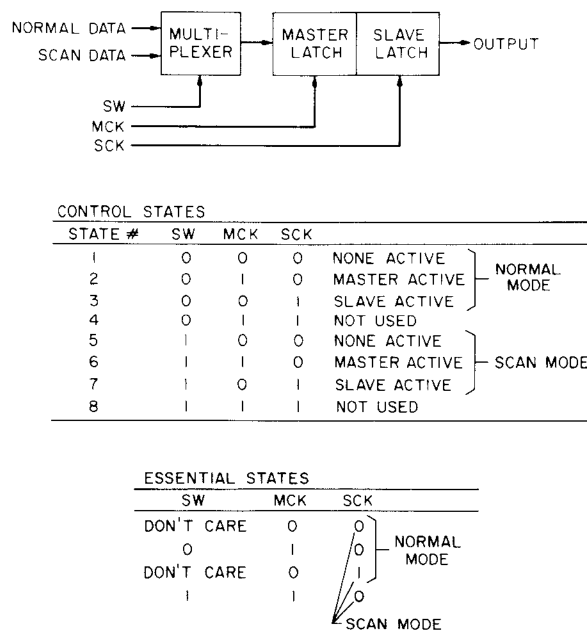


Fig. 1. Scan register flip-flop with multiplexer. SW = 0 selects normal data and SW = 1 selects scan data.

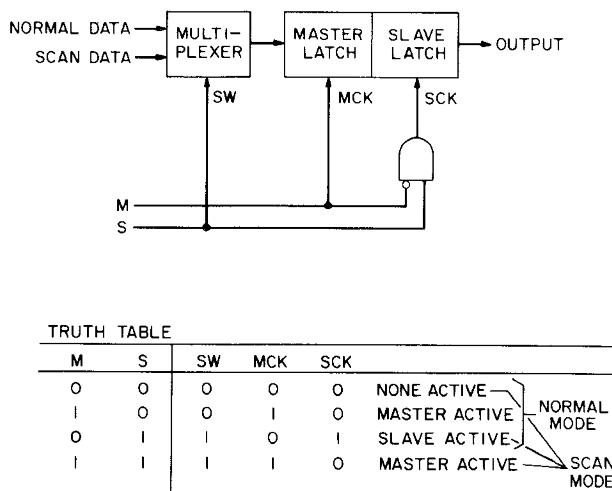


Fig. 2. Multiplexer type of scan register flip-flop with two control signals and the decoding gate.

In the scan mode, S is continuously held high, and a square wave is applied to M; the flip-flop operation is now controlled by a single clock. For this mode of operation, there is an essential hazard, but careful design of the delays internal to the flip-flops will avoid any problem due to this hazard. It may be desirable to have the slave clock OFF (SCK = 0) before the master latch becomes active (MCK = 1). This can be accomplished by holding the S signal "low" (as shown by the dotted waveform in Fig. 3) every time an upward transition of M takes place. As a result, momentarily, normal data are fed to the master latch. However, the situation is corrected before MCK turns low, thus latching the correct scan data into the master latch.

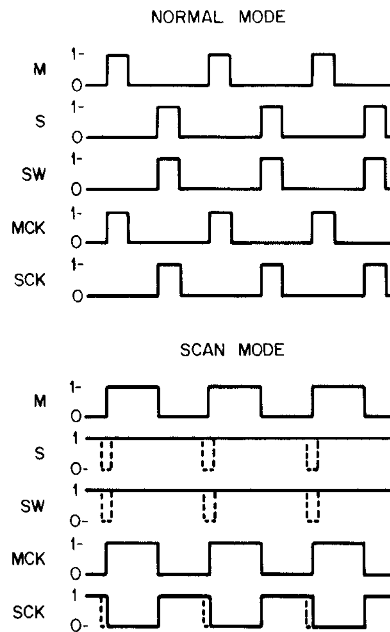


Fig. 3. Waveforms for the flip-flop of Fig. 2. The waveform shown in the dotted line can be used to ensure that the slave latch is turned off before the master latch goes on.

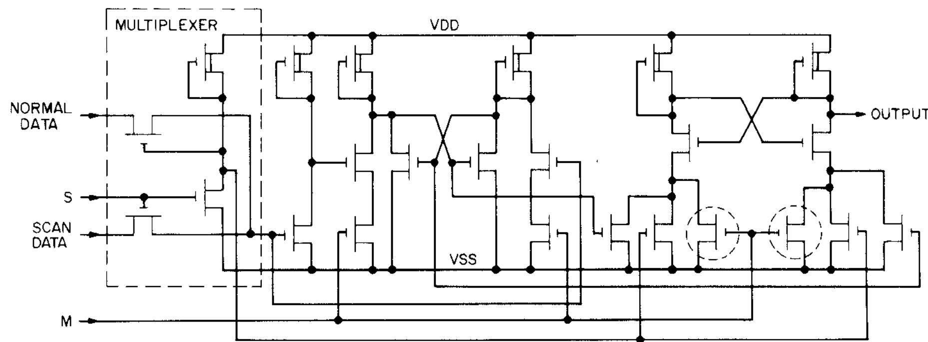


Fig. 4. An implementation of a multiplexer type of scan register flip-flop. Multiplexer (four transistors) is the overhead of the scan design. The two transistors which are encircled represent the additional overhead of the proposed design that eliminates the mode-select signal.

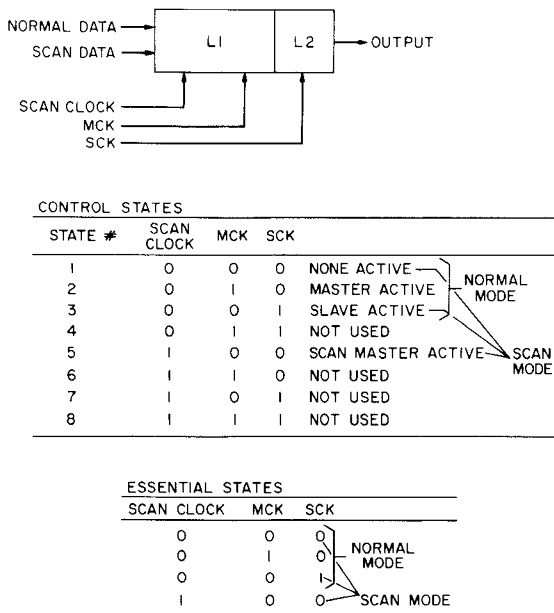
This approach allows the scan clocking also to be level-sensitive.

AN NMOS IMPLEMENTATION

Fig. 4 shows a scan register flip-flop implemented with 24 transistors. The master and slave latches are constructed using 18 transistors, while the multiplexer switch uses 4 transistors. $S=1$ feeds scan data to the master latch, and $S=0$ feeds normal data to the master latch. $M=1$ activates the master latch, and the slave latch is activated by a simultaneous $M=0$, $S=1$. Only two transistors (shown encircled in Fig. 4) have been added for decoding the slave clock signal from M and S. The waveforms for this flip-flop are the same as shown in Fig. 3. The flip-flop cell was actually laid out and its operation was verified through simulation.

EXAMPLE 2—SCAN REGISTER FLIP-FLOP WITH SCAN CLOCK

Another type of scan register flip-flop [2] is shown schematically in Fig. 5. Here the slave latch $L2$ is clocked by the signal SCK. The master latch $L1$ is modified to accommodate two clock signals, MCK and the "scan clock." MCK latches the normal data into $L1$, while the scan clock latches the scan data into $L1$. An implementation of such a circuit is given in [2]. As shown in Fig. 5, out of the eight possible operations specified by the three clock lines, only the first three are used in the normal mode. Scan mode uses the first, third, and fifth operations. Thus, there are again four essential operations. As shown in Fig. 6, these four operations can be specified to $L1$ and $L2$ through two lines M and S by a three-gate decoding circuit. Fig. 7 shows the waveforms. In the scan mode, the



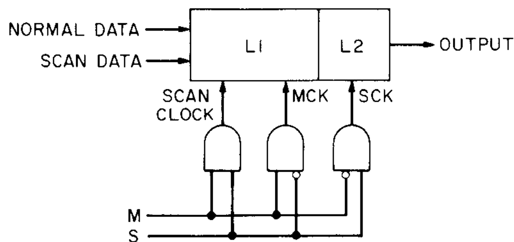
CONTROL STATES

STATE #	SCAN CLOCK	MCK	SCK		
1	0	0	0	NONE ACTIVE	NORMAL MODE
2	0	1	0	MASTER ACTIVE	NORMAL MODE
3	0	0	1	SLAVE ACTIVE	NORMAL MODE
4	0	1	1	NOT USED	
5	1	0	0	SCAN MASTER ACTIVE	SCAN MODE
6	1	1	0	NOT USED	
7	1	0	1	NOT USED	
8	1	1	1	NOT USED	

ESSENTIAL STATES

SCAN CLOCK	MCK	SCK	
0	0	0	NORMAL MODE
0	1	0	NORMAL MODE
0	0	1	NORMAL MODE
1	0	0	SCAN MODE

Fig. 5. Scan register flip-flop with scan clock. For implementation of $L1$ and $L2$, see [2].



TRUTH TABLE

M	S	SCAN CLOCK	MCK	SCK		
0	0	0	0	0	NONE ACTIVE	NORMAL MODE
1	0	0	1	0	MASTER ACTIVE	NORMAL MODE
0	1	0	0	1	SLAVE ACTIVE	NORMAL MODE
1	1	1	0	0	SCAN MASTER ACTIVE	SCAN MODE

Fig. 6. Scan register flip-flop of Fig. 5 modified to eliminate the scan clock.

signal S is held "high," while M is a square wave. Note that the scan mode uses a single-clock operation, while the normal mode operation is level-sensitive with two clocks.

Some Useful Remarks

It should be noted that in both the scan register flip-flops presented above, the two signals M and S in the normal mode have the same definition as two nonoverlapping system clocks. Thus, the operation is level-sensitive. In the scan mode, the flip-flops essentially operate with a single clock; however, the waveforms on M and S are still quite simple.

The ideas presented above can be applied to any multiple (two or more) clock circuit. For a single-clock circuit, however, one must use an extra input pin for the SW signal. The method presented here may be applicable to those single-clock circuits that require a level-sensitive operation in the normal mode. In such cases, the two nonoverlapping clock signals can be produced by a clock-generating circuit as shown in Fig. 8. These two signals are then routed to all the flip-flops in the circuit. In our modification, a simple circuit, as shown in Fig. 9, can be used to produce the two signals M and S. Notice that for the normal mode, SW $=0$, $M=$ MCK, and $S=$ SCK. When SW $=1$, $M=$ CLOCK, and $S=1$, and since CLOCK is a square-wave signal, M and S will perform the scan operation as shown in Fig. 3. To avoid spurious clock pulses, SW should always be changed when CLOCK $=0$ (and SCK $=1$).

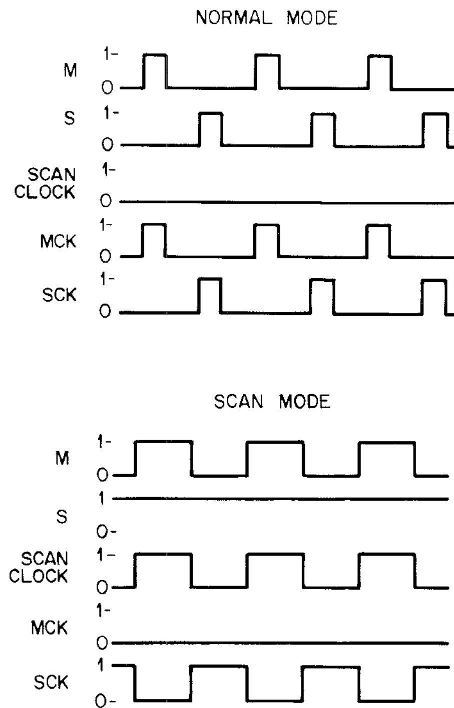


Fig. 7. Waveform for the flip-flop of Fig. 6.

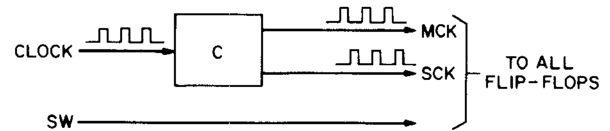


Fig. 8. Clock generator for level-sensitive operation in a single clock circuit. In scan design, mode-select (SW) is a separate pin. SW, MCK, and SCK are routed to all flip-flops in the circuit.

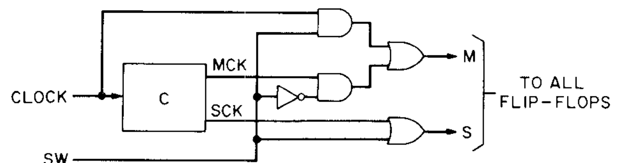


Fig. 9. A simple scheme that generates two signals M and S to be routed to all flip-flops in the circuit. The flip-flops of the type shown in either Figs. 2 or 6 should be used.

Testing the Scan Register

Two tests have traditionally been run to verify the scan operation.

A *flush test* is used in two-clock circuits [4]. This test simultaneously activates both the master clock and the scan clock in scan mode. Inputs of one and zero are successively applied at the scan register input; these input values propagate through the scan chain and are checked at the scan chain output. The flush test is not possible for the clocking scheme proposed above because master and slave latches cannot be simultaneously activated. This, however, is not a serious limitation since the single-clock scan design of [3] already manages without a flush test.

A *shift test* [4] is used in both two-clock and single-clock circuits. This test shifts patterns of zero and one through the shift register in the scan mode. The shift test is possible for the clocking scheme proposed above, and it was verified by actual fault simulation that all possible single stuck-at faults in the scan path are detected by the shift test. Therefore, no significant scan register testing capability is sacrificed in the new clocking scheme.

Estimated Overhead Savings

As suggested earlier, the proposed clocking scheme reduces the required routing area at the expense of some decoding logic in each flip-flop. In the standard cell layout [5] of the chip, the area is divided between the cell rows and the routing channels. Since a chip design using this concept has not been completed, the routing overhead given below is an estimate. The flip-flop cell was, however, laid out and the cell area is based on the actual layout.

The routing area which is saved corresponds to one control signal which would normally be routed to every flip-flop on the chip. In the case of standard cell layout [5], this saving depends on the average number of routing tracks per cell row. In actual designs, this number can vary typically from 5 to 20 routing tracks per standard cell row, depending on the particular design realized. One routing track can feed two standard cell rows (the row above and the row below the routing track). Since the average number of tracks per routing channel serving two cell rows is between 10 and 40, the corresponding overhead of one additional track per routing channel is between 10 and 2.5 percent. It is thus reasonable to expect that this approach will reduce routing area by 2.5–10 percent—depending on the average connectivity of the design.

The estimate of the decoding logic overhead is based on actual layout of the flip-flop cell. Note that the NMOS realization of Fig. 4 required 22 transistors and 2 additional transistors for the decoding. The layout of the standard cell for the circuit of Fig. 4 required 16 grids. A conventional 22-transistor scan register flip-flop cell implementation also used the same number of grids. This is because much of the area internal to a standard flip-flop cell is used for the internal connections of the required transistors. Thus, assuming no decoding area overhead, the total chip area savings may be as high as 10 percent, depending on the particular design.

Conclusion

It is well recognized that in VLSI circuits, long routing paths are more expensive in terms of chip area than a few devices which are locally connected. An innovation in the scan design is presented here which completely eliminates the mode-select (or scan clock) signal. The function of this signal is accomplished by making use of the signal redundancy on the clock leads. The definition of the clock signals in the normal mode remains unchanged. An example of a scan register flip-flop is given to illustrate that the decoding circuit (which is added to each flip-flop) can be implemented with just two additional transistors. Although a chip design using the presented concept has not been completed, the scan register flip-flop cell, which is the crucial component of this design, has been laid out and its hazard-free operation checked through simulation. The cell layout also justifies the overhead claim.

Acknowledgment

The authors thank A. K. Bose, M. K. Maul, and H. N. Nham for their comments on the manuscript.

References

[1] M. J. Y. Williams and J. B. Angell, "Enhancing testability of LSI circuits via test points and additional logic," *IEEE Trans. Comput.*, vol. C-22, pp. 46–60, Jan. 1973.
[2] E. B. Eichelberger and T. W. Williams, "A logic design structure for LSI testability," *J. Design Automat. Fault-Tolerant Comput.*, vol. 2, pp. 165–178, May 1978.
[3] S. Funatsu, N. Wakatsuki, and A. Yamada, "Designing digital circuits with easily testable consideration," in *Dig. Papers, 1978 Semiconductor Test Conf.*, Cherry Hill, NJ, 1978, pp. 98–102.
[4] P. S. Bottorff, R. E. France, N. H. Garges, and E. J. Orosz, "Test generation for large logic networks," in *Proc. 14th Design Automat. Conf.*, New Orleans, LA, June 1977, pp. 479–485.
[5] G. Persky, D. N. Deutsch, and D. G. Schweikert, "LTX—A minicomputer-based system for automated LSI layout," *J. Design Automat. Fault Tolerant Comput.*, vol. 1, pp. 217–255, May 1977.

M. Ray Mercer (S'65–M'69) received the B.S. degree from Texas Tech University, Lubbock, in 1968, the M.S. degree from Stanford University, Palo Alto, CA, in 1971, and the Ph.D. degree from the University of Texas, Austin, in 1980, all in electrical engineering.

He worked on real-time control and analysis systems at GTE Sylvania, Mountain View, CA, from 1968 to 1973. At Hewlett-Packard Laboratories, Palo Alto, CA, he was involved with the automation of manufacturing and testing processes for exotic solid-state devices in 1973–1977. He taught and did research at the University of Texas, San Antonio, from 1977 to 1980, and worked on the computer-aided design and testing of digital circuits at Bell Laboratories, Murray Hill, NJ, from 1980 to 1983. Currently, he is an Assistant Professor of Electrical Engineering at the University of Texas, Austin.

Dr. Mercer's paper, coauthored with V. D. Agrawal, won the Best Paper Award at the 1982 IEEE International Test Conference. He is a member of Tau Beta Pi and Eta Kappa Nu.

Vishwani D. Agrawal (S'68-M'70-SM'80) received the B.Sc. degree from Allahabad University, Allahabad, India, in 1960, the B.E. (Honors) degree from Roorkee University, Roorkee, India, in 1964, the M.E. degree from the Indian Institute of Science, Bangalore, in 1966, and the Ph.D. degree from the University of Illinois, Urbana, in 1971.

His experience includes teaching (1966-1967) in the Department of Electrical Engineering, Indian Institute of Technology, New Delhi; research (1967-1970) at the Antenna Laboratory, University of Illinois, Urbana; digital test technique development for Illiac IV Project (1970-1971) at Automation Technology, Inc., Champaign, IL; electromagnetic pulse (EMP) experimentation (1971-1972) at E.G. and G., Inc., Albuquerque, NM; teaching and research (1972-1975) at the School of Radar Studies, Indian Institute of Technology, New Delhi; and spacecraft antenna design (1975-1978) at TRW Defense and Space Systems Group, Redondo Beach, CA. Since 1978, he has been working on computer-aided design and testing of LSI circuits at the Bell Laboratories, Murray Hill, NJ, where he is now the Supervisor of the Test Aids Group.

Dr. Agrawal was co-recipient of the Best Applications Paper Award from the IEEE Antennas and Propogation Society in 1979, and his paper, coauthored with M. R. Mercer, was awarded the Best Paper Award at the 1982 IEEE International Test Conference. He is a Fellow of the Institution of Electronics and Telecommunication Engineers (India).

Functional Testing of EPROM's

MAREK PAWŁOWSKI, ANDRZEJ WOŹNIAK, AND JAN ZABRODZKI

Abstract—**The aim of this paper is to develop a testing scheme for EPROM memories. The start point is the assumed general model of EPROM memory logic structure. For this model, an adequate fault model is developed. The class of faults taken into consideration includes faults in input-output buffers, faults in address decoding circuitry, and faults in memory cell arrays. The proposed testing scheme enables detection of all faults included in the assumed fault model. This scheme takes into account technological and economic aspects as well. The method proposed is illustrated by detailed solutions for the 2716 EPROM memory.**

I. Introduction

Theoretically, EPROM memories can be regarded as a subclass of RAM memories—in both, one can write and read selected cells. Such a treatment cannot be extended to the problem of EPROM memories testing. Detailed analysis of EPROM's functional, electrical, and dynamic parameters leads to the conclusion that numerous methods developed for RAM memories testing cannot be directly adopted for the needs of EPROM's testing. Nevertheless, there are only a few papers devoted to the problem of EPROM's testing.

The main UVEPROM characteristics distinguishing them from RAM memories, from the point of view of testing, are as follows.

1) The memory content is nonvolatile.

2) The newly manufactured memory, or memory after erasing, contains all 1's or all 0's, depending on the type of memory.

3) During the word programming phase, one can change the memory cell state only in one direction ($1 \rightarrow 0$ or $0 \rightarrow 1$, depending on the type of memory).

4) Long programming time (tens of milliseconds for one word).

5) Long erasing time measured in minutes.

6) One cannot erase individual words—the erasing procedure affects the whole memory.

II. Model for UVEPROM Memory

When analyzing different types of UVEPROM memories (2708, 2716, $\cdots$), one can conclude that the general model of the internal logic structure for such memories can be represented as in the block diagram shown in Fig. 1. The memory cells are ordered into the p matrices with $m \times n$ dimension each. The individual words (of length p) are selected by the row decoders (1 of m) and p column decoders (1 of n). All inputs of the decoder are buffered by address buffers.

The selected word is transferred to the output lines O through the output buffers enabled by the signal $\overline{OE}$. Data written into memory are programmed into the selected cells using input buffers enabled by the PGM signal.

The memory array is built of FAMOS technology EPROM cells (Fig. 2). Each cell is a single stacked-gate transistor implemented using two layers of polycrystalline silicon. The cell consists of a bottom floating gate and a top select gate. The top gate is connected to the row decoder, while the floating gate is used for charge storage. The cell is programmed by injection of high-energy elec-

Manuscript received May 10, 1983; revised October 31, 1983.
The authors are with the Institute of Computer Science, Warsaw Technical University, 00-665 Warsaw, Poland.

0018-9200/84/0400-0212$01.00 ©1984 IEEE