Use of Storage Elements as Primitives for Modeling Faults in Synchronous Sequential Circuits

W. K. Al-Assadi, Y. K. Malaiya†, and A. P. Jayasumana
Electrical Engineering Department
† Computer Science Department
Colorado State University
Fort Collins, CO 80523

Abstract

While fault models for various implementations of combinational logic have been examined in detail, only limited attention has been paid to storage elements. In many designs, storage elements (SEs) represent a significant fraction of the electrical nodes. Most testable designs are geared towards testing of combinational logic, and generally depend on an increase in number of SEs. The stuck-at fault model assumes that an output (input) of a SE can be stuck-at-0 or 1. This minimal fault model may not model some common fault behaviors for SEs. These include feed-through *faults which cause the cell to exhibit* data or clock-feed-through *behavior, and the inability to latch H or L signals. In a master-slave cell, or when two-phase clocking is used, many of these faults may appear as output stuck-at faults. The minimal fault model may be an efficient fault model when only modest coverage may be required. The* enhanced fault model *proposed here represents the effects of faults in popular SEs more effectively.*

1. Introduction

Testing of sequential circuits has long been known to be a very difficult problem. A common approach is to convert the problem of testing synchronous sequential circuits into the simpler problem of testing combinational circuits. This is accomplished by using design for testability approaches, which provide direct access to inputs and outputs of combinational blocks [1, 2]. If one can assume that most faults within a SE can be modeled as stuck-at-0/1 faults on the outputs, then these faults do not need to be explicitly considered. This is because such faults are equivalent to the stuck-at faults in the combinational logic surrounding the SEs. This paper critically examines the validity of such an assumption.

Some recent papers address the detection of several physical failures in CMOS latch cell. Reddy and Reddy [3] found that several stuck-opens in CMOS latches are not detectable, though they may degrade timing performance. They have proposed testable implementations for the transmission-gate latch and the symmetric D-latch. Liu and McClusky [4] have suggested a testable implementation for the transmission-gate latch cell in which any stuck-open faults can be detected. Anglada and Rubio [5] have also analyzed the symmetrical D-latch for stuck-open faults. Lee and Breuer [6] have investigated several bridging faults in CMOS scan registers implemented by using a CMOS transmission-gate latch and proposed a universal test sequence that also detects all irredundant stuck-at and stuck-open faults. They also found that current monitoring is necessary for the detection of all irredundant bridging faults. Here a more complete set of possible faults is considered that may alter the functional behavior of the cell.

This paper examines the major transistor-level faults for three elementary SEs which represent different implementations of the D-latch. The behavior of each cell under the above faults is analyzed to evaluate possible functional fault models. The elementary SEs are basic building blocks in complex cells like master-slave registers and static memory chips, and thus the results would have wide applicability. Here the results for elementary SEs have been used for characterizing master-slave flipflops. As an example, faults in a gray code counter are examined.

2. Stuck-at Model for Storage Elements

Often the SEs are handled in one of two common ways. The first approach is to assume that the internal faults cause the SE output or input to be stuck-at-0/1. In many situations when faults are only considered in the combinational logic, this implicity assumes that any internal faults in a SE can be either be shown to be equivalent to a stuck-at-output, or a stuck-at-input,

0-8186-3180-5/92 $3.00

which in turn would appear as a stuck-at-output of the combinational logic surrounding the SE. This is termed a *minimal fault model*. This model presumes that a stuck-at-0/1 fault at the output (input) of a SE is equivalent to the stuck-at-0/1 fault of the combinational logic input fed (output feeding) by the SE. The second approach does not regard a SE as a primitive, but regards it as a network of transistor or gate level components. This approach can be computationally prohibitive for large circuits.

We examine below the effectiveness of the minimal fault model in representing physical failures. The results reveal the need for a more accurate fault model to better represent the physical failures at the transistor level of an elementary SE so that a fault model for complex SEs can be inferred from the elementary SE fault model. This can reduce the test generation and fault simulation efforts significantly.

To examine a SE cell, in general an input sequence is required rather than a single input vector. Let T=$\{t_1,....,t_n\}$ be the set of all possible input combinations. For an elementary synchronous SE with input D and a control signal CLK, t_i is a 2-tuple corresponding to (D, CLK) and n=4. Let $R(s, t_i)$ be the response of the cell to the input vector t_i applied to the cell when the cell is at state s. The behavior of each cell under all possible transistor faults is examined for all input combinations and previous states. Multivalued logic is used to give better representation for voltage levels that are not exactly logic 1 (hard 1) or logic 0 (hard 0) [7]. These two levels, termed 'soft 1' and 'soft 0', are recognized by the following gate as full logic levels but cannot drive a pass transistor fully on. Here *High* level (H) corresponds to both 'hard 1' and 'soft 1', and *low* level corresponds to both 'hard 0' and 'soft 0'. A fault that causes the SE output to be L(H) for all $t_i \in T$ regardless the state of the SE can be modeled as stuck-at-0/1. Under some faults the output of the faulty cell cannot have a high to low (low to high) transitions. These can be described by the notation H$\not\rightarrow$L (L$\not\rightarrow$H) defined below [8].

Definition 1: Consider a faulty SE cell in the state H. If there exists no input vector t_i ($t_i \in T$) such that $R(H, t_i) = L$ then the behavior is said to be H$\not\rightarrow$L. L$\not\rightarrow$H is defined in a similar manner.

Under these faults we may not be able to find input vectors that will force the cell into a certain state. A H$\not\rightarrow$L (L$\not\rightarrow$H) will generally appear as stuck-at-1 (stuck-at-0), however an initial 0(1) is possible due to power on conditions.

In this work an *enhanced fault model* is proposed, which provides a higher explicit fault coverage for the SEs under consideration. It includes *feed-through* faults. The effect of these faults is remarkably different from stuck-at-0/1 faults. Such faults cause the cell to become either *data-feed-through* or *clock-feed-through*. Such conditions can lead to timing problems or coupling between combinational blocks which are normally separated by the SEs [9]. The formal definition for these two behaviors is given below [8].

Definition 2: A faulty SE cell is said to have a *feed-through* fault if it becomes either *data-feed-through* or *clock-feed-through*.

(i) A faulty SE cell is said to be *data-feed-through* when its behavior becomes combinational such that $R(s, t_i) = f(y)$ for each $t_i \in T$, where y is the data part of t_i.

For example, for a D-latch, y is a single element vector containing the input D only.

(ii) A faulty synchronous SE cell is said to be *clock-feed-through* if $R(s, t_i) = CLK$ or $\overline{CLK}$ where CLK is the control signal.

3. Detailed Examinations of the elementary SEs

In this section, the three cells are examined. Each cell is examined for all possible transistor faults. Results obtained analytically have been verified by using SPICE. A good functional fault model is sought such that the functional behavior of faulty SEs can be adequately described. Both the *minimal* and the *enhanced* fault models are examined for their effectiveness in representing the functional faults.

In the presence of stuck-open faults, a SE cell could turn from static to dynamic under some input vectors. This means that the logic value of the output of the cell is maintained due to the high impedance state caused by the charge stored on the capacitance associated with the output node. This state may not last for a long time due to the leakage of the charge stored leaving the output node at an indeterminate state. However, if the clock is toggled often (at rates of normal frequency), such a state cannot be detected. Examining a SE cell for stuck-on and bridging faults reveals the importance of using multivalued logic and the effect of transistor sizing. In this work, we have assumed a (W/L) ratio of pMOS transistor to nMOS transistor to be 1.5. The results given are still applicable for different ratios as long as '0' dominates if outputs of two similar gates are shorted. The outcome of a bridging fault between two nodes depends on the strength of the nodes. The source nodes V_{dd} and V_{ss} (ground) are considered the strongest nodes in the circuit, hence in general, the source node signal dominates if it is bridged with any node in the circuit. The outcome of briding faults between logical inputs and internal nodes of the circuit are in general unpredictable [10]. Bridging faults between internal

nodes of the same well are included. We refer to such a fault as (x,y) corresponding to a short between nodes x and y. Bridging faults between internal nodes of different wells are not included because the probability of having such faults is very small. Analysis assumes that the bridging fault corresponds to a *hard short*. In general, stuck-on and bridging faults change the resistance of the path between V_{dd} and V_{ss}. This suggests that monitoring the leakage current, which is many orders higher than the fault free leakage current, can detect many such faults. Figure 1 shows the naming convention used for bridging faults.

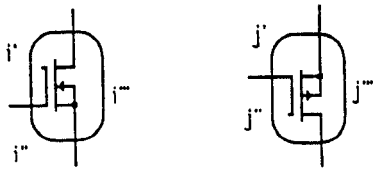


Figure 1: Naming convention used for bridging faults

The clock is applied such that the cell is devoid of clock turn-on and turn-off hazards and glitches, i.e. the data is stable when clock changes. The latch is race free if the data is stable when clock is active [3]. Hence input data (D) is allowed to change only when the control signal (CLK) is low. Under some faults, a synchronous SE exhibits behavior change in the latch phase, while still functioning properly in the transparent phase. These changes cause the cell to be unable or sometimes unable to latch 1(0). The definitions for such cases are given below [8].

Definition 3: Let the state of a faulty synchronous elementary SE during the transition from the transparent phase to the latch phase be $Q = 1(0)$. If the state of the cell becomes 0(1) during the latch phase irrespective of data input, then the cell is said to be *unable to latch* 1(0). The notation *UL-1 (UL-0)* is used to describe this.

Definition 4: Let the state of a faulty synchronous elementary SE during the transition from the transparent phase to the latch phase be $Q = 1(0)$. If for a specific data input (D) during the latch phase, Q changes to 0(1), the cell is said to be *sometimes unable to latch* 1(0). This is described by using the notation *SUL-1 (SUL-0)*.

Figure 2 shows the cell termed *Clocked D-latch*. This cell is commonly used in the implementation of scan-path registers such as LSSD and scan-path flip-flop in the TITUS system[2]. Table 1 shows the behavior of the cell under all possible faults. Some bridging faults cause the cell to be transparent (i.e. $Q1 = D$) and therefore the cell becomes *data-feed-through*. Complex behavior that can be represented as a logical function of the control signal CLK is observed under some faults. Bridging faults between internal nodes are also included. Such faults are; $f_1 = (a,b)$, $f_2 = (b,c)$, $f_3 = (c,d)$, $f_4 = (a,c)$, $f_5 = (a,d)$, and $f_6 = (b,d)$. The table shows that the minimal model will cover 44 out of 96 faults considered (i.e. 45.8%), while the enhanced model will cover 6 more , i.e. 50 faults (i.e. 52%).

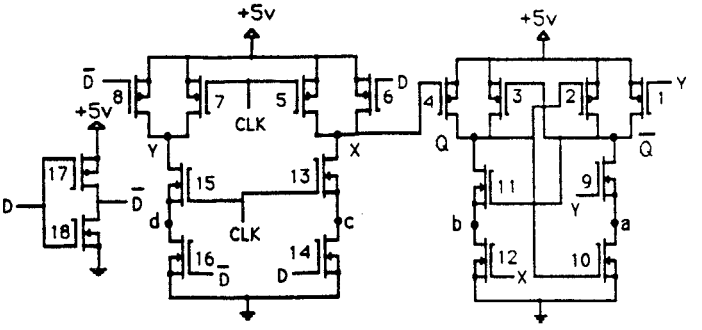


Figure 2: The clocked D-latch

Output (Q)	Faults	Model
1	Stuck-on: 3, 4 Bridging: $2'$, $3'''$, $4'''$, $12''$	s-at-1
H↛L	Stuck-open: 1, 11, 12, 15, 16, 17 Stuck-on: 7, 8, 18 Bridging: $1'$, $4''$, $6'$, $6''$, $7'''$ $8'''$, $9'''$, $11''$, $12'$, $15''$, $16'$, $16''$ $17'$, $17''$, $18'$, $18'''$	s-at-1
0	Bridging: $10''$	s-at-0
L↛H	Stuck-open: 4, 13, 14 Stuck-on: 5, 6 Bridging: $4'$, $5'''$, $6'''$, $12'''$, $13''$, $14''$, $18''$	s-at-0
D	Bridging: $5'$, $5''$, $7'$, $7''$, $13'$, $15'$	f-t*
SUL-1	Bridging: $15'''$	- - - -
UL-1	Stuck-open: 9, 10; Stuck-on: 1, 2 Bridging: $1''$, $1'''$, $2'''$, $3'$, $8''$, $9'$, $9''$, $10'$, $11'''$	- - - -
SUL-0	Bridging: $13'''$	- - - -
UL-0	Bridging: $10'''$, f_1	- - - -
indeterminate	Bridging: $2''$, $3''$, $11'$	- - - -
Fault free	Stuck-open: 2, 3, 5, 6, 7, 8, 18 Stuck-on: 9, 10, 11, 12, 13, 14, 15, 16, 17 Bridging: $8'$, $16'''$, $17'''$, f_2, f_3, f_4, f_5, f_6	- - - -
Complex	Bridging: $14'$, $14'''$	- - - -

*=The cell becomes data-feed-through and is modeled in the enhanced model

Table 1: Behavior of the clocked-D latch cell under all possible faults

The second cell shown in Figure 3 is another structure of the D-latch, termed *CMOS transmission-gate latch*. This structure is used in some scan-path register designs [3, 6]. The two-clock phases are assumed to be individually buffered, i.e. bridging of one clock signal does not affect the complimentary clock signal.

Bridging faults within the transmission gate shown in the figure have been analyzed assuming '0' dominance and are given in table 2.

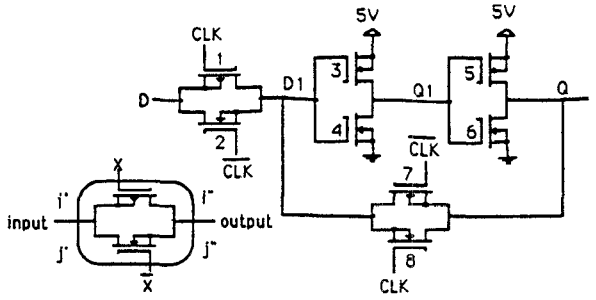


Figure 3: The CMOS transmission-gate latch

Table 4 shows that a bridging fault between CLK or $\overline{CLK}$ and node $D1$ cause the control signal to be transparent. Such a fault causes the cell to become *clock-feed-through* and is modeled as *feed-through* in the *enhanced* model.

Inputs		Output behavior				
A	X	normal	i'	j'	i''	j''
0	0	0	0	0	0	1
0	1	z	z	z	1	0
1	0	1	0	1	0	1
1	1	z	z	z	1	0
Model		FF	s-a-0	FF	X	$\overline{X}$

z=high impedance state

Table 2: Bridging faults Behavior of transmission gate

In this cell, stuck-open faults are difficult to detect. Stuck-open faults in transistors 1 and 2 are undetectable at node Q and the cell remains static. These two faults can only be detected by observing the delay in logic transition (delays in the faulty and fault free cell) at node $D1$. Stuck-open faults in transistors 5 and 6 cause charge sharing between nodes Q and $D1$ leaving node Q at an indeterminate level. The results in Table 3 show that the minimal fault model includes 17 out of 40 (i.e. 42.5%) faults considered, while the enhanced model includes 23 faults (i.e. 57.5%). It is important to notice that the above results could be different if the observations are made at node $Q1$. However, a higher fault coverage could be achieved if more than one observation point is used.

The third cell shown in Figure 4 is termed the *Symmetric D-latch*. Table 4 shows the results under all possible transistor faults. In the table, two possible bridging faults between internal nodes are also included. These faults are $f_1 = (a, b)$, and $f_2 = (c, d)$ in Figure 5. The table shows that the minimal fault model covers 24 faults out of 52 (i.e. 46%) faults considered, while the enhanced model covers 28 faults (i.e. 53.8%).

Output (Q)	Faults	Model
H↛L	Stuck-open: 3;Bridging: $2'$	s-at-1
0	Stuck-on: 6;Bridging: $3'''$, $4''$ $5'$, $6'''$	s-at-0
L↛H	Stuck-open: 4;Bridging: $1'$ $7'''$, $8'''$	s-at-0
Fault free	Stuck-open: 1, 2, 7, 8 Stuck-on: 1, 7, 8	- - - -
D	Bridging: $1'''$, $2'''$	f-t*
CLK	Bridging: $1''$, $8''$	f-t†
indeterminate	Stuck-open: 5, 6 Bridging: $3''$, $4'$, $5''$, $6'$	- - - -
$\overline{CLK}$	Bridging: $2''$, $7''$	f-t†
SUL-1	Stuck-on: 2	- - - -
UL-1	Bridging: $7'$	- - - -
Complex	Stuck-on: 3, 5;Bridging: $8'$	- - - -

*=The cell becomes data-feed-through and is modeled in the enhanced model
†=the cell becomes clock-data-through and is modeled in the enhanced model

Table 3: Behavior of the CMOS transmission-gate cell under all possible faults

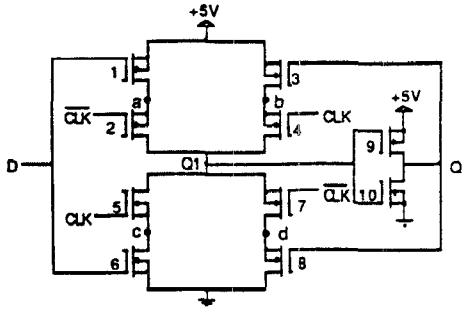


Figure 4: The Symmetric D-latch

Output (Q)	Faults	Model
1	Bridging: $3'$, $9'''$, $10''$	s-at-1
H↛L	Stuck-open: 1, 2, 10 Stuck-on: 6, 7 Bridging: $1'$, $1''$, $2'$, $5'''$, $6'''$, f_2	s-at-1
0	Stuck-on: 10 Bridging: $8''$, $9'$, $10'''$	s-at-0
L↛H	Stuck-open: 5, 6, 9 Bridging: $1'''$, $4'''$, $5''$, $6''$	s-at-0
Fault free	Stuck-open: 3, 4, 7, 8 Stuck-on: 1, 2, 3, 4;Bridging: f_1	- - - -
CLK	Bridging: $2''$, $7'$	f-t†
$\overline{CLK}$	Bridging: $5'$, $4''$	f-t†
indeterminate	Bridging: $9''$, $10'$	- - - -
SUL-0	Stuck-on: 5	- - - -
UL-0	Stuck-on: 8;Bridging: $3''$, $4'$, $8'''$	- - - -
SUL-1	Bridging: $2'''$, $7''$	- - - -
UL-1	Bridging: $3'''$	- - - -
Complex	Bridging: $6'$	- - - -

†=The cell becomes clock-feed-through and is modeled in the enhanced model

Table 4: Behavior of the Symmetric-D cell under all possible faults

4. Fault model of Complex SE cells

In this section, we examine the fault model for the master-slave structure composed of two synchronous SEs. The structure, shown in Figure 5, is controlled by a single clock signal. The proposed fault model for the complex SEs is inferred from the fault model of a single SE presented in Section 3, which gives accurate representation of the physical failures at the transistor level. Therefore accurate fault models can be obtained at higher level complex SEs without significant loss of information about the structure of the circuit.

Table 5 summarizes the mapping between the fault models of an elementary SE cell used as a master latch and that of the corresponding master-slave cell.

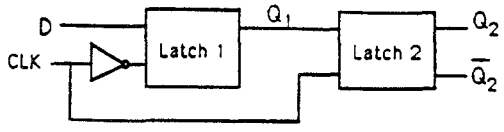


Figure 5: The Master-Slave cell

The results show that the *minimal fault model* should be augmented by considering *data-feed-through* faults if high coverage is desired. Results are the same if two non-overlapping clocks are used except for the SUL faults which might not be sensed due to the timing considerations between the two clocks. For this structure, any fault in the slave latch has the same effect as that of the corresponding faulty elementary SE.

Higher level fault models taking advantage of the hierarchical design of complex VLSI systems using higher level primitives can reduce the complexity of test generation and fault simulation by reducing the number of elements to be considered.

Faulty master latch	Behavior of Master-Slave cell	Model
Data-feed-through	Data-feed-through	f-t
Clock-feed-through	s-at-Q	s-at-0/1
s-at-0/1	s-at-0/1	s-at-0/1
$UL-1$	s-at-0	s-at-0
$UL-0$	s-at-1	s-at-1
$SUL-0(1)$ for input vector t_i	proper behavior except for input vector t_i, output≡1(0)	- - --
Fault free	Fault free	- - --

Table 5: Mapping of elementary SE fault model and a master-slave cell fault model

Example: Let us consider the Gray up-down counter finite state machine shown in Figure 6 [11]. This is a double latch design using two non-overlapping clock signals. When $F = 1$, the circuit displays on $Z1$, $Z2$ the modulo 4 Gray code representation of the number of positive clocks received. When $F = 0$, the circuit counts backward (modulo 4). The effect of four different faults is illustrated in Figure 7, which gives the faulty state tables/diagrams. In Figure 7(a), the state diagram shows that since latch1 is *UL-0*, transitions into states (00) and (01) are not possible. The behavior shown in Figure 7(b) is similar except that the latch is *sometimes* unable to latch a 0. Figure 7(c) illustrates the behavior when latch3 is data-feed-through. The state diagram shows that the fault results in *race-ahead*, i.e a state may be reached one clock period too early. Figure 7(d) shows the effect of latch3 clock-feed-through, which appears as stuck-at-0 at output $Z2$.

5. CONCLUSION

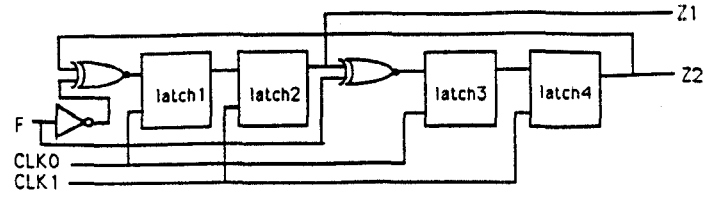
Faults in four elementary storage cells have been examined. The results obtained analytically have been verified by SPICE. The *minimal model* may not show a high explicit fault coverage for elementary SEs. For more complex cells such as master-slave cells, flip-flops, and scan-path registers, the minimal model may still be effective when considering the elementary SE as a primitive, provided high coverage is not sought.

An enhanced fault model is proposed for elementary SEs which includes faults that cause the cell to become either data-feed-through or clock-feed-through. This model provides a higher explicit coverage than the minimal fault model. When a SE becomes data-feed-through, the fault may or may not affect the normal operation depending on the timing involved. We will examine the related timing issues for test generation and simulation in the future.

A question that might arise is whether one should seek the convenience of using simple functional fault models or should high coverage be sought. The analysis has shown that, in general, both can not be achieved at the same time, since it is not always possible to find a simple and elegant fault model which will represent effects of all possible physical defects.

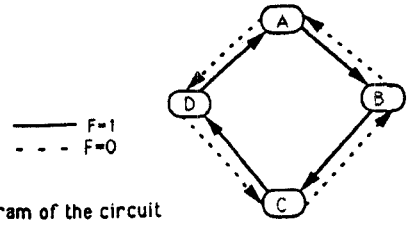
Acknowledgment

This work was supported by a SDIO/IST funded project monitored by ONR.



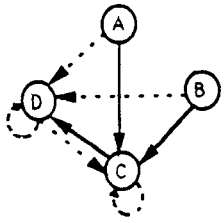
(a) The circuit diagram

Z1 Z2	F=1	F=0
00	01	10
01	11	00
11	10	01
10	00	11



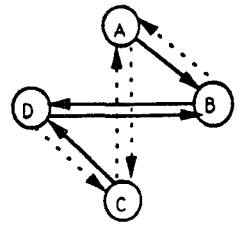
(b) The state diagram of the circuit

Figure 6: The Gray up-down counter



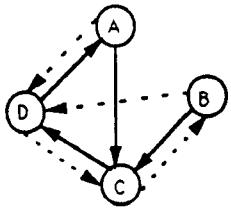
	Z1 Z2	F=1	F=0
A	00	11	10
B	01	11	10
C	11	10	11
D	10	10	11

(a) Latch1 UL-0



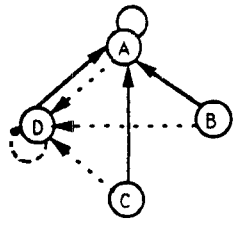
	Z1 Z2	F=1	F=0
A	00	01	11
B	01	10	00
C	11	10	00
D	10	01	11

(c) Latch3 Data-feed-through



	Z1 Z2	F=1	F=0
A	00	11	10
B	01	11	10
C	11	10	01
D	10	00	11

(b) Latch1 SUL-0



	Z1 Z2	F=1	F=0
A	00	00	10
B	01	00	10
C	11	00	10
D	10	00	10

(d) Latch3 Clock-feed-through

Figure 7: Behavior of the counter under several faults

References

[1] Y. K. Malaiya, B. Gupta, A. P. Jayasumana, R. Rajsuman, S. Menon, and S. Yang, " Functional Fault Modeling for Elementary Static Storage Elements," *Technical Report, 1989.* (Presented at 12*th* DFT Workshop).

[2] V. D. Agrawal, S. K. Jain, and D. M. Singer, "Automation in Design for Testability," *IEEE 1984 Custom Integrated Circuits Conf.* pp. 1 59-163.

[3] M. K. Reddy and S. M. Reddy, "Detecting FET Stuck-Open Faults in CMOS Latches and Flipflops," *IEEE Design & Test of Computers,* pp. 17-26, October 1986.

[4] D. L. Liu and E. J. McCluskey, "A CMOS Cell Library Design for Testability," *VLSI Systems Design,* pp. 58-65, May 4, 1987.

[5] R. Anglada and R. Rubio, "Functional Fault Models for Sequential Circuits," *Research Report DEE-3,* Electronic Engineering Department, Polytechnical University of Catalunya, Barcelona 1987.

[6] K. J. Lee and M. A. Breuer, "A Universal Test Sequence or CMOS Scan Registers," *IEEE 1990 Custom Integrated Circuits Conf.* pp. 28.5.1-28.5.4.

[7] P. Banerjee and J. A. Abraham, "A Multivalued Algebra for Modeling Physical Failures in MOS VLSI Circuits," *IEEE Transaction on Computer-Aided Design,* vol. CAD-4, no. 3, pp. 312-321, July 1985.

[8] W. K. Al-Assadi, Y. K. Malaiya, and A. P. Jayasumana, " Use of Storage Elements as Primitives for Modeling Faults in Sequential Circuits," Tech. Report CS-92-129, Computer Science Dept., Colorado State University, August 1992.

[9] W. K. Al-Assadi, Y. K. Malaiya, and A. P. Jayasumana, "Detection of Feed-Through Faults in CMOS Storage Elements," Accepted in the Fourth NASA Symposium on VLSI Design, October 29-30, 1992.

[10] Y. K. Malaiya, A. P. Jayasumana, and R. Rajsuman, " A Detailed Examinations of Bridging Faults," *Proc. IEEE Conf. on Computer Design,* pp. 78-81, 1986.

[11] E. J. McCluskey, *Logic Design Principles,* Prentice-Hall, 1986.