Report Number: WUCS-87-32

1987-12-01

Classical Fault Analysis of MOS VLSI Circuits

Brian L. Shing and Mark A. Franklin

Due to the large cost involved in generating effective input vectors to test MOS circuits, finding ways to reduce this test vector generation cost is of considerable interest. In this paper, empirical results show the fault coverage obtained form MOS transistor-level fault simulation using randomly generated test inputs can be approximated by the fault coverage obtained using the test vectors generated from classical stuck-at-zero and stuck-at-one fault simulation on logic-gate-level circuits. Applying this results, an approach is presented to reduce the cost of test vector generation for MOS circuits.
... Read complete abstract on page 2.

Classical Fault Analysis of MOS VLSI Circuits

Brian L. Shing and Mark A. Franklin

Complete Abstract:

Due to the large cost involved in generating effective input vectors to test MOS circuits, finding ways to reduce this test vector generation cost is of considerable interest. In this paper, empirical results show the fault coverage obtained form MOS transistor-level fault simulation using randomly generated test inputs can be approximated by the fault coverage obtained using the test vectors generated from classical stuck-at-zero and stuck-at-one fault simulation on logic-gate-level circuits. Applying this results, an approach is presented to reduce the cost of test vector generation for MOS circuits.

CLASSICAL FAULT ANALYSIS OF MOS VLSI CIRCUITS

Brian L. Shing and Mark A. Franklin

WUCS-87-32

December 1987

Computer and Communications Research Center
Department of Computer Science
Washington University
Campus Box 1045
One Brookings Drive
Saint Louis, MO 63130-4899

For Submission

to

1988 IEEE International Conference on Computer Design:
VLSI in Computers & Processors

Classical Fault Analysis for MOS VLSI Circuits

by

Brian L. Shing and Mark A. Franklin

Computer and Communications Research Center
Campus Box 1115
Washington University
St. Louis, Missouri 63130
(314) 889-6106

Abstract:

Due to the large cost involved in generating effective input vectors to test MOS circuits, finding ways to reduce this test vector generation cost is of considerable interest. In this paper, empirical results show that fault coverage obtained from MOS transistor-level fault simulation using randomly generated test inputs can be approximated by the fault coverage obtained using the test vectors generated from classical stuck–at–zero and stuck–at–one fault simulation on logic–gate–level circuits. Applying this result, an approach is presented to reduce the cost of test vector generation for MOS circuits.

Keywords:

fault simulation, stuck–at fault, stuck–on fault, stuck–open fault, open connection fault, shorting fault

Classical Fault Analysis for MOS VLSI Circuits

Brian L. Shing and Mark A. Franklin

Computer and Communications Research Center
Campus Box 1115
Washington University
St. Louis, Missouri 63130

1. Introduction

Developing test vectors and testing fabricated chips are critical but costly steps in the overall production of very large scale integrated (VLSI) circuits. Typical examples of fabrication failures are the misalignment of mask layers, oxide defects and implant defects. These failures often appear as short and open connections[1, 2], which may generate undesirable circuit connections or destroy desired circuit connections. Due to the complexity and size of VLSI circuits, enormous amounts of computation time are required to generate test vectors which can accurately detect circuit fabrication errors. In fault simulation, a circuit model with one or more fabrication errors (faults) is simulated to determine if an input vector can detect the presence of error(s) in the circuit. Since the time to obtain an entire test vector set (set of test vectors which detects all possible fabrication errors) using fault simulation is approximately proportional to the cube of the circuit size[3], and VLSI circuits contain typically between 50,000 and 500,000 transistors, fault simulation is typically very expensive. This paper is concerned with finding ways to reduce test vector generation costs while maintaining reasonable fault coverage[4].

Traditionally, test vector sets have been generated by performing classical fault simulation. In classical fault simulation, a fault is represented as a stuck–at–one (SA1) or stuck–at–zero (SA0) signal at the input or output of a circuit component[5]. Due to the nature of metal oxide semiconductor (MOS) technology, many MOS faults such as transistor stuck–on and stuck–open faults[6, 7, 8], open and short electrical connections are not modeled well by the classical fault approach[9, 10]. Transistor stuck–on and stuck–open faults, for example, are MOS failures

which cause the faulty transistor to be always conducting or not conducting at all. For CMOS circuits, the stuck–open faults may change a combinational circuit to a sequential circuit, whereas stuck–on faults may generate additional wired–or connections. On the other hand, open connection faults unintentionally break circuit connections and, shorting connection faults generate undesirable circuit connections. These two types of faults may alter the functionality of the original circuit design by physically changing the circuit layout or cause the circuit to fail completely. One approach to generating test vectors in the MOS environment, therefore, is to model components with sufficient detail so that MOS faults can be accurately represented. This unfortunately, dramatically increases the computational complexity of the simulation and the number of fault types which must be considered.

In generating test vectors for MOS circuits, a circuit model which decreases the complexity of the original circuit and simplifies the simulation process without sacrificing much accuracy, would be helpful in reducing the cost of VLSI MOS fault simulation. The technique considered in this paper is to model MOS transistor-level circuits with logic–level gates (e.g., NAND, NOR) and three-terminal switches (pass transistors), and to perform classical fault analysis (stuck–at–zero and stuck–at–one) at the inputs and outputs of the model components. By working on a logic-gate-level abstraction of the MOS circuit and using a simplified fault analysis model to select test vectors, we trade the accuracy of test vectors for the expensive computation time required by MOS fault simulation. To see if the test vectors generated by classical fault analysis are adequate for MOS circuits, we need to examine their effectiveness in detecting MOS faults. This is the fundamental issue addressed in this paper.

In the simplified approach taken in this paper, classical faults are injected one at a time into a logic-gate-level model of the circuit. Random test vectors are then applied to the model inputs. The number of output vectors which are different from those of a fault–free circuit and the input vectors which cause the difference are then recorded. The fault coverage is computed by dividing this output vector count by the number of potential faults. Augmenting the classical faults by the

different types of physical MOS faults, the same simulation procedure is repeated with the same full set of randomly generated test vectors applied to the more accurate (but more costly) MOS transistor-level fault models. Then the MOS fault coverages generated by the entire random vector set and the set that detected only classical faults are compared. If the differences in the MOS fault coverages between the two different vector sets are small, we can conclude that classical fault analysis on logic-gate-level circuits is a reasonable approach to generating test vectors for MOS circuits.

In this paper, only the potential transistor (stuck–on and stuck–open) and connection (open line and shorting between adjacent neighbor signal pair) faults which are reflected in the transistor level design of a MOS circuit are considered for the MOS fault simulations. There are several reasons for considering this restricted set of possible MOS faults. First, since a MOS design can be laid out in many different ways (and thus may contain different fabrication errors), we select the potential fault set based on a transistor-level description of the circuit. Second, transistor stuck–on and stuck–open faults correspond to MOS fabrication errors at the diffusion layer, whereas adjacent signal pair shorting and open connection faults model the metal layer or polysilicon layer fabrication errors. (Two signals are adjacent if they are connected to two different terminals of a circuit component). Since Galiay[2] has said that about 75 percent of fabrication errors in VLSI circuits are shorts and opens at the levels of metallization and diffusion; many of these potential MOS fabrication errors are included in the fault set we considered for this paper. Finally, the large number of potential layout dependent shorting faults (2^n for a circuits with n signals) in a MOS circuit makes the simulation of all these faults impractical. By including only the adjacent neighbor signal pair shorting faults in the transistor level MOS circuit design, we cover the more probable shorting faults.

Applying the above classical and MOS fault simulation experiments to four benchmark circuits yield the following results:

1. Empirical results confirm general literature statements to the effect that classical fault

coverages of logic-gate-level circuit models do not closely correspond to MOS fault coverages of transistor-level circuit models.

2. A strong correlation exists between the detection of classical and MOS faults by a given test vector. Furthermore, the simulation data of the four circuits show that the MOS fault coverage obtained from MOS transistor-level fault simulation using randomly generated test inputs can be approximated by the MOS fault coverage obtained using test vectors generated from classical fault simulation on logic-gate-level circuits.

3. Using the result of 2., an approach can be developed which yields a two to four times speedup in the MOS fault simulation process when relatively low fault coverages are acceptable. For high fault coverage of around 90%, this approach has a speedup of roughly 20%.

In the remainder of the paper, the above results are presented more fully. Classical and physical MOS faults and their corresponding fault models are discussed. The classical and MOS fault simulation experiment results are analyzed and an approach to speeding up the test vector generation process for MOS circuits is presented.

2. Classical and MOS Faults

One way to determine whether a circuit operates properly as designed, is by applying all possible combinations of inputs to the circuit and comparing the resultant outputs with the corresponding faultless circuit outputs. Any difference in outputs indicates the presence of a *fault*, otherwise the tested circuit is 100 percent *fault free*. Before we can proceed to describe the fault simulation experiments, definitions of classical fault and physical MOS faults that we consider are needed.

Classical fault modeling is based on the assumption that all faults in a logic circuit can be modeled by a stuck–at–one or stuck–at–zero at the input or output of a circuit component[5, 11]. In classical fault modeling, a circuit component is considered to be either a standard device (NAND, NOR, switch etc.) or a primary input/output. Furthermore, primary inputs and primary outputs are taken to be circuit components with one output or one input respectively.

There are many different types of physical failures in MOS circuit. The majority of the faults are shorts and opens at the levels of metallization and diffusion[2]. Table 1 lists the major

MOS fault types, their corresponding physical failures and their effects. For the fault simulation experiments performed in this thesis, we considered only the potential MOS faults induced from the transistor level description of a MOS circuit.

The transistor stuck–open fault corresponds to the transistor–not–conducting fault. The normal consequence of this fault is an open connection on the circuit path which stops the flow of intended electrical signals. In a CMOS circuit, a stuck–open fault may cause a combinational circuit to act as a sequential circuit[7, 12], therefore a sequence of two test vectors is needed to detect a stuck–open fault[6]. The other type of transistor fault considered is the stuck–on fault[7].

Table 1. Common MOS Faults and Effects

MOS Faults		
fault types	physical failures	actual effects
transistor stuck-open	Open connection in diffusion layer between drain and source of a transistor.	Transistor will not conduct and may generate high impedance state to convert a combinational circuit to a sequential circuit.
transistor stuck-on	Short connection in diffusion layer between drain and source of a transistor.	Transistor will always conduct and may generate undesired wire-or connection or power to ground path.
open connection	Broken electrical connection at a signal line (usually at metal or polysilicon layer).	Destroy circuit connection and alter circuit function.
short connection	Bridged electrical connection of two or more signal lines (usually at metal or polysilicon layer).	Create new circuit connection and alter circuit function.

A stuck–on fault represents the condition of the transistor-always-conducting. Typically, stuck–on faults will generate an unintentional wired–or circuit connection which may alter the functionality of the circuit.

An open fault (as opposed to a transistor stuck–open fault) is generally regarded as a broken connection in a circuit network. In the MOS fabrication process, many sources such as defects in the patterning mask may contribute to the presence of open faults. An open fault may force the broken connection to be at a high impedance state. Thus, testing open connection faults is similiar to testing transistor stuck–open faults and requires a sequence of two test vectors. Like open faults, shorting faults are commonly generated during the MOS fabrication process. In general, shorting faults create additional circuit connections by bridging two or more signals in a circuit. Shorting signals together effectively creates wired–or connections of the signals that are shorted. In this paper, only pair shorting between two adjacent signals is considered. Two signals are adjacent if they are connected to the input(s) or output(s) of the same transistor. For example, the signals that connect to the gate and drain of a transistor are adjacent to each other.

Table 2 shows the definitions of symbols to be used throughout this paper. Using these definitions, the problem can be defined as:

GIVEN:

1) Circuit models, C_{GATE} and C_{MOS}.
2) Fault sets, F_{CL} and F_{MOS}.
3) Input vector set, V.

FIND:

1) Test vector set, V_{CL}.
2) Fault coverages, $FC_{MOS}(V)$ and $FC_{MOS}(V_{CL})$.
3) $\Delta FC = FC_{MOS}(V) - FC_{MOS}(V_{CL})$.

PROBLEM:

Based on the value of ΔFC, decide whether V_{CL} is an effective test vector set for detecting F_{MOS} in C_{MOS}.

In order to answer the question posed by the above problem, a set of fault simulation experiments

Table 2. Symbol Definitions

Symbol	Definition
C	A circuit
C_{GATE}	The logic-gate-level circuit model of C
C_{MOS}	The MOS transistor-level circuit model of C
F	Fault set in C
F_{CL}	A set of potential classical faults in C_{GATE}
F_{MOS}	A set of potential MOS faults in C_{MOS}
V	A set of randomly generated input vectors
V_{CL}	The subset of V which detect classical faults in F_{CL}
V_{MOS}	The subset of V which detect MOS faults in F_{MOS}
$FC(v)$	Fault coverage of vector set v
$FC_{CL}(v)$	Classical Fault coverage of vector set v
$FC_{MOS}(v)$	MOS Fault coverage of vector set v

were designed and performed on four benchmark circuits.

3. Fault Simulation and Benchmark Circuits

There are many ways to perform fault simulation on a circuit. In this paper, since we are interested in the effectiveness of the test vector sets generated by classical fault simulation methods on MOS circuits, designing an optimal simulation algorithm is not our main objective. We elected to design a serial fault simulation algorithm for synchronous scan design (SD) circuits with randomly generated input vectors and single fault injection.

Due to the growing number of components per integrated circuit chip, and the limited number of external pins available to a chip package, gates internal to a chip are not easily accessible and therefore are difficult to test. Various studies have suggested use of the scan design structure[13, 14] to improve chip testability. With the scan design structure, shift register chains

are added to the circuit design so that data can be shifted into and out of the chip internals. This additional feature allows internal subcircuits to be tested independently of each other. Because of the improvement in testability and the relatively small overhead introduced by the additional shift register chains to the total chip cost, the scan design structure has become widely used in VLSI design. We focus our fault simulation on synchronous circuits designed with the scan design structure. In addition, since performing fault simulations on all possible faulty circuit configurations is very time consuming and costly. (e.g., for a MOS circuit with n signal connections, there are roughly 2^n potential single faults. and 2^{n^*} potential single and multiple faults) our fault simulations have concentrated on simulating circuit configurations having only a single fault and therefore create faulty circuits by injecting one fault to the fault–free circuit at a time.

One way to completely test a circuit for detectable faults is to observe how the circuit responds to all the different combinations of primary inputs and memory states. As the circuit size increases, the number of different input combinations required and the computation time involved to perform such a complete test becomes impractical. In addition, the actual number of input vectors needed to find all the detectable faults is generally much less than the test patterns required for a complete test[15]. While there are various ways to select input test vectors, it has been shown that the use of pseudo random input vectors is generally adequate for generating test vector sets which have reasonable fault coverage (i.e., between 65 to 85 percent)[13, 15]. We therefore generate input vectors for our simulation experiment randomly. Since we are simulating scan design structured circuits, the registers (memory) in the circuit are accessible along with the primary inputs. We define an input vector in our fault simulation to be a combination of all the primary inputs (except the clock inputs) and the registers connected to the scan design shift register chains. Moreover since transistor stuck–open faults and open connection faults are included in the set of potential faults for the MOS fault simulation experiments, a sequence of two input vectors is required to detect the above fault types.

Using scan design type circuits in our simulations, we expand the set of circuit outputs that can be monitored from only circuit primary outputs to primary outputs plus the circuit memories accessible by scan design register chains. We defined a fault as being detected if there is at least one corresponding output difference between the faulty circuit outputs and the fault–free circuit outputs. In addition, we require that this difference between outputs must be between valid logic values (i.e., logic zero and logic one, or logic one and logic zero, but not involving say logic one and unknown state x). Since our simulations are intended for simulating the steady state logic behavior of the circuit, faults caused by timing, transient behavior and charge sharing are not considered and thus are not detected in our simulation experiments. In addition, open connection faults at the gate input of a transistor are also not detected by our fault simulation. This is because a "free–hanging" gate input does not yield enough information to decide whether the faulty transistor is being turned–on or turned–off.

The fault simulation algorithm was implemented based on a logic simulator(*lsim*) developed at Washington University. *Lsim*[16, 17] is a serial discrete event driven simulator, capable of simulating the logic behavior of a circuit at the gate and switch (transistor) level. It operates under the UNIX* operating system and was implemented in the C programming language. An important objective of *lsim* was to provide facilities for the collection of data in studying the logic simulation process and certain features supported by *lsim* are particularly useful for the implementation of fault simulation algorithms. First, *lsim* has a C programming interface such that external programs written in the C language can be built around *lsim* to extend its capabilities. Second, during the simulation of a circuit, *lsim* maintains signal values as long as there are no input changes to alter these values. This signal value storage feature allows non–classical MOS faults to be simulated by *lsim* without adding memory components to the circuit for modeling the faults. Third, all the component inputs/outputs and signals in a circuit are accessible in *lsim*. This allows circuit internals to be monitored and controlled by the users so

*UNIX is a Trademark of AT&T Bell Laboratories

that a non–scan design circuit design can be tested in scan design style by *lsim*. *Lsim* provides control to users which permit assignment of logic states to component inputs/outputs or signals anywhere in a circuit. The "force" command enforces a logic state so that the affected site behaves as if it has been stuck–at a given logic state. The "set" command allows users to assign input values to the appropriate test spots in a circuit.

One important step in expanding *lsim* to perform fault simulation is to provide ways to represent and model the behaviors of different fault types in the simulator. Table 3 shows a summary of modeling classical and MOS faults in *lsim*. Among all the modeling of different fault types using the "force" and "free" commands, the modeling of classical faults follow directly from the definitions of the faults, whereas the modeling of open connection and transistor faults rely on the effect of the commands applied at the appropiate locations in the circuit. In classical

Table 3. *Lsim* Fault Models

FAULT		*LSIM* MODEL
Classical	stuck-at-one	force component input/output to logic state one.
	stuck-at-zero	force component input/output to logic state zero.
MOS	transistor stuck-open	force gate of n(p)–channel transistor to logic state zero (one).
	transistor stuck-on	force gate of n(p)–channel transistor to logic state one (zero).
	open connection	force component input/output to high impedance logic state.
	short connection	create temporary shorting connection in the circuit description.

fault analysis, a fault is defined as component input/output being stuck–at logic value zero or one, and this can be modeled by applying the *lsim* ''force'' command with logic states zero and one at component input or output. An open connection fault at component input or output can be modeled by forcing a high impedance logic state (z) at the corresponding component connection. With transistor stuck–on and stuck–open faults, the faulty transistors have lost their ability to turn–off and turn–on respectively. Since the gate input of a transistor controls the turning on and off of the transistor, a stuck–on fault in a n(p)–channel transistor can be modeled by forcing a logic one (zero) at the gate of the transistor, while a stuck–open fault in a n(p)–channel transistor can be modeled by the gate input of the transistor being forced at logic zero (one).

Due to the nature of shorting faults, which create unintentional wired–or connections in a circuit, they cannot be easily modeled by forcing logic states at circuit connections. One way to model a shorting fault is to change the component and signal connections in the circuit description. In our implementation, a shorting fault between a pair of signals is modeled by creating a temporary shorting signal in the internal *lsim* circuit representation, such that the two signals that are intended to be shorted together are represented by this temporary signal during the simulation. After the analysis of the shorting fault is performed, this temporary signal can be deleted, and the circuit can be brought to its fault–free state by restoring the two shorted signals back to their original forms. This ''create & restore'' scheme allows different shorting faults to be modeled using one circuit description, and avoids paying the cost of loading different circuit descriptions into *lsim* for each separate shorting fault simulation.

In order to answer the question whether the test vectors generated from the classical fault simulation on logic-gate-level circuits are effective for detecting MOS faults, four benchmark circuits have been used to collect data for our fault simulation experiments. The four circuits are:

1) stopw – a stop watch.
2) pqueue – a priority queue.
3) rtpalu – the ALU of a Radiation Treatment Planning (RTP) chip.
4) rtpctrl – the control logic of the same RTP chip.

The stop watch circuit is an electronic timer which determines the elasped time between a start and a stop signal. The priority queue is an event manipulation device capable of maintaining a partially sorted set of records so that the record with the smallest value can be easily retrieved. The RTP chip is a hardware implementation of an algorithm used in radiation treatment planning for cancer therapy. It calculates the expected amount of radiation dosage at a specific point of a subject. Because of the size of the RTP chip, it is further divided into two circuits for our fault simulations. The ALU circuit performs the dosage calculation and the control logic circuit controls the execution of the RTP algorithm.

These circuits are synchronous designs and are the product of several graduate student design projects. Each circuit was designed using either the NMOS or CMOS integrated circuit technology. For the purpose of our fault simulation experiments, we need to generate test vectors for the logic-gate-level circuits and study the effectiveness of these vectors on the corresponding NMOS and CMOS circuits. The logic-gate-level and the NMOS(CMOS) versions of each circuit are generated by replacing the orignal CMOS(NMOS) design with standard logic-gates (and three terminal switches if necessary) and NMOS(CMOS) style gates respectively. Table 4 lists the four circuits with the number of circuit components (N) present at the logic-gate-level, NMOS and CMOS transistor-level versions of the circuits. The size of the test circuits were kept small enough to allow reasonable fault simulation running times and to avoid exceeding the available main memory capacity during program execution. The columns labeled with F_c, F_t, F_o, and F_s are the number of classical, transistor, open connection and adjacent signal pair shorting faults respectively possible for each circuit. Columns F_{NMOS} and F_{CMOS} list the total number of NMOS and CMOS faults for each circuit. The last column indicates the approximate execution time (in

Table 4. Benchmark Circuits

Logic-Gate-Level			
Circuit	N	F_c	time/fault
stopw	420	1812	1.6
pqueue	490	2222	1.5
rtpalu	501	2078	1.7
rtpctrl	520	2626	1.7
average	483	2185	1.6

NMOS Transistor-Level						
Circuit	N	F_t	F_o	F_s	F_{NMOS}	time/fault
stopw	575	844	1408	1154	3406	2.4
pqueue	791	1106	1929	1448	4483	3.2
rtpalu	730	966	1802	1447	4215	3.2
rtpctrl	1199	1690	3124	1807	6621	4.7
average	824	1152	2066	1464	4682	3.4

CMOS Transistor-Level						
Circuit	N	F_t	F_o	F_s	F_{CMOS}	time/fault
stopw	870	1696	2507	1614	5817	5.2
pqueue	1019	1892	2819	1991	6702	7.1
rtpalu	852	1628	2448	1719	5795	6.3
rtpctrl	1171	2258	3373	1815	7446	6.2
average	978	1869	2787	1785	6440	6.2

*(time/fault is per vector in μVAX II/GPX CPU seconds.)

CPU seconds) on a μVAX II/GPX* for simulating a single fault per single test vector for each circuit. Using the CMOS case as example, for a circuit with 6434 CMOS faults (the average number of CMOS faults for the benchmark circuits), the simulation time for one test vector requires about 11 hours of CPU time on a μVAX II/GPX. Even though the number of faults in the simulation list will be reduced as they are being detected, and the simulation time per vector will decrease, this enormous simulation time has prevented us from selecting larger test circuits for the simulation experiments.

4. Simulation Data Analysis

In this section, the empirical results obtained from the fault simulation experiments performed on four benchmark circuits are presented. Figures 1 and 2 plot fault coverage versus number of random input vectors tried in the fault simulations of two of the four benchmark circuits. The figures show that classical fault coverages at logic-gate-level tend to be higher than both NMOS and CMOS fault coverages. For the four benchmark circuits, the absolute NMOS and CMOS fault coverages on average are 17% and 37% lower than the classical fault coverage respectively. This is because faults in MOS circuits may generate high impedance logic states and wired-or connections which are not detected given our conservative fault detection assumptions. Furthermore, the CMOS complementary design structure generates faults that are undetectable without a timing analysis of the circuit outputs. The above data confirm the general literature observations that fault coverages obtained by classical fault simulations do not closely correspond to MOS fault coverages. However, during the analysis of the simulation data, the classical fault coverage curve of a particular circuit was observed to have a very similar shape to the corresponding MOS fault coverage curve. Closer examination of the data shows that a break at a given point on the classical curve is usually echoed by breaks at the corresponding points on the NMOS and CMOS curves. This observation suggests that a correlation may exist between

*VAX is a Trademark of Digital Equipment Corportaion.

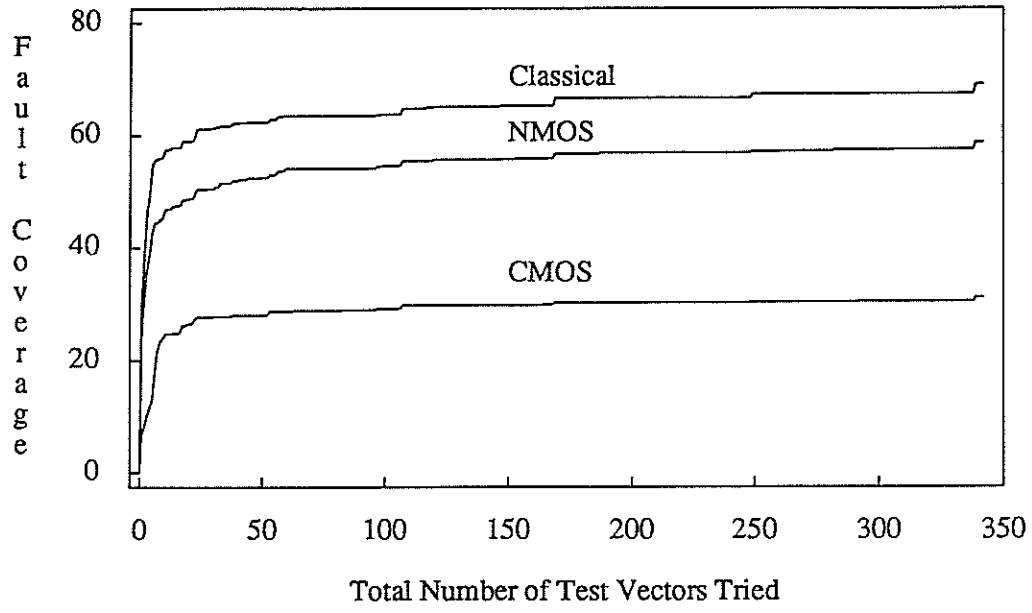

Figure 1. STOPW Circuit Fault Coverages

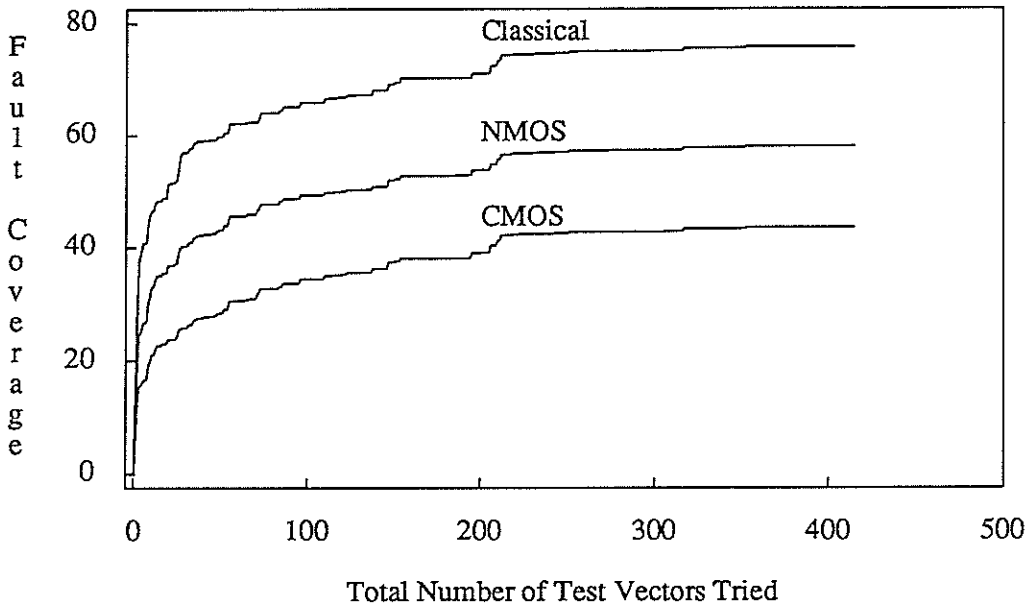

Figure 2. RTPCTRL Circuit Fault Coverages

detection of classical faults and MOS faults by a given test vector. To further study this relationship, the set of random input vectors used in the fault simulation experiments of a circuit were divided into four different groups based on the fault detecting ability of each input vector.

Figure 3 shows the breakdown of the overall set of random inputs into four input vector groups. In the Venn diagram[18], the universial set is taken to be the set of random input vectors, V. V_{CL} and V_{MOS} are subsets of V where every vector in the subsets detects at least one classical fault or at least one MOS fault respectively during the simulation runs. V_{CL} and V_{MOS} divide the Venn diagram into four disjoint subsets of vectors (V_1, V_2, V_3, and V_4) with their corresponding abilities to detect different fault types. Table 5 lists the descriptions and the set equivalence of the four vector groups.

Applying the above categorization scheme, the sets of random test vectors employed in the fault simulation experiments are now examined. Tables 6 and 7 list the number of occurrences of each vector type for the classical/NMOS and classical/CMOS cases. For each circuit, a large majority of the vectors in the entire random test vector either detected no faults (V_1 type vectors) or detected both classical and MOS faults (V_2 type vectors). This indicates that a correlation between the detection of classical and MOS faults by a given test vector exists. Thus generating test vectors for MOS circuits by using logic-gate-level classical fault simulation may be possible. However, input vector types V_3 (detect only classical faults but not MOS faults) and V_4 (detect only MOS faults but not classical faults) do occur during the simulations. To decide whether the test vectors generated from classical fault simulation (V_{CL}) are good approximation to those required to test for MOS faults, the contribution of test vectors in V_4 to the final MOS fault coverage should be studied.

Figure 4 and 5 show the classical and MOS fault coverage curves obtained by simulating faults on the stopw and rtpctrl circuits. On each of the figures, the curve at the top represenets the logic-gate-level classical fault coverage of the circuit, the two sets of curves in the middle and at

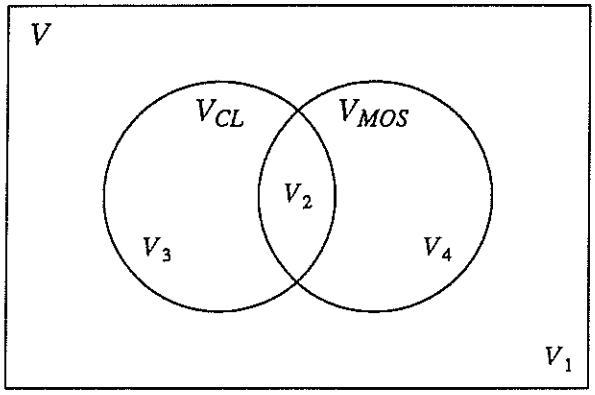

Figure 3. Venn Diagram of Test Vector Categorization

Table 5. Test Vector Categorization Table

Vector Set	Set Equivalence	Descriptions
V_1	$V - (V_{CL} \cup V_{MOS})$	the set of vectors from V which detect neither classical nor MOS fault during the fault simulations.
V_2	$V_{CL} \cap V_{MOS}$	the set of vectors from V which detect both classical and MOS faults during the fault simulations.
V_3	$V_{CL} - (V_{CL} \cap V_{MOS})$	the set of vectors from V which detect only classical faults but not MOS faults during the fault simulations.
V_4	$V_{MOS} - (V_{CL} \cap V_{MOS})$	the set of vectors from V which detect only MOS faults but not classical faults during the fault simulations.

Table 6. Classical/NMOS Fault Detection Correlation

Classical / NMOS							
circuit	no. of vectors in						
	V_1	V_2	V_3	V_4	V	V_{CL}	V_{MOS}
stopw	288	37	3	14	342	40	51
pqueue	155	26	13	27	221	39	53
rtpalu	285	28	10	5	328	38	33
rtpctrl	332	75	7	1	415	82	76
average	265	42	8	12	327	50	54

Table 7. Classical/CMOS Fault Detection Correlation

Classical / CMOS							
circuit	no. of vectors in						
	V_1	V_2	V_3	V_4	V	V_{CL}	V_{MOS}
stopw	290	29	11	12	342	40	41
pqueue	166	30	9	16	221	39	46
rtpalu	284	20	18	6	328	38	26
rtpctrl	332	75	7	1	415	82	76
average	268	39	11	9	327	50	48

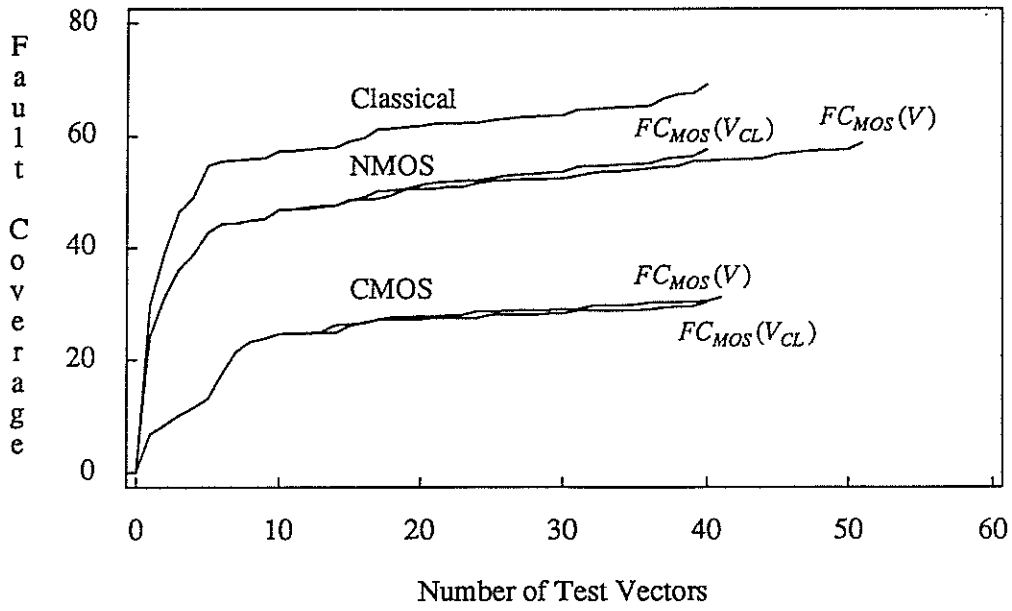

Figure 4. Compacted Fault Coverage Curves of STOPW Circuit

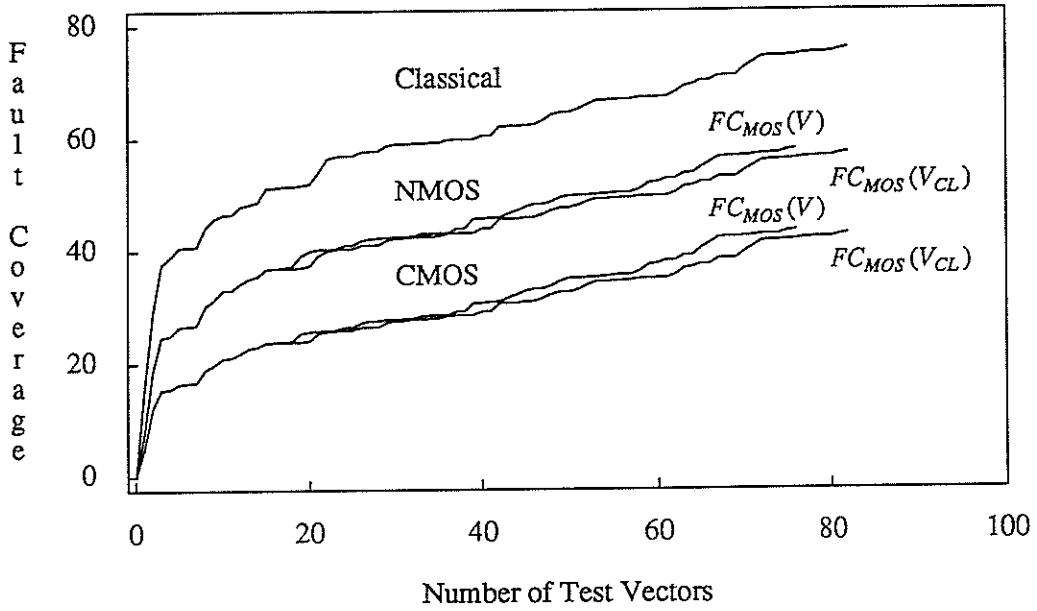

Figure 5. Compacted Fault Coverage Curves of RTPCTRL Circuit

the bottom show the transistor-level NMOS fault coverages and transistor-level CMOS fault coverages respectively. Within each set of curves, the upper curve (one has a higher final fault coverage) is the fault coverage obtained by running the MOS fault simulations with the full set of random vector V. The lower curve (one has a lower final fault coverage) in each curve set plots the fault coverage $FC_{MOS}(V_{CL})$ corresponds to the classically generated vector set, V_{CL}. In all four of the circuits, the MOS fault coverage curves show that the difference between the final fault coverage values on the $FC_{MOS}(V_{CL})$ and $FC_{MOS}(V)$ curves are small (i.e., $[FC_{MOS}(V) - FC_{MOS}(V_{CL})]/FC_{MOS}(V) < 0.1$).

The small differences between $FC_{MOS}(V)$ and $FC_{MOS}(V_{CL})$ implies that although the vector set V_{CL} does not include all the test vectors that will be generated during a MOS fault simulation with random input vector set V, those vectors that were left out of V_{CL} (i.e., the vectors in the set V_4) do not contribute heavily to the final MOS fault coverage. In other words, the inclusion of vectors from V_4 adds little to the set of faults already found with those vectors in V_{CL} alone. Therefore, generating test vectors using the vector set V on MOS fault simulation can be approximated by the vectors set generated by the less costly classical fault simulation.

In addition to the benchmark circuit simulation experiments described above, similiar sets of experiments were performed on a number of smaller circuits (each containing less than 40 transistors). Due to their smaller sizes, exhaustive input combinations were applied and tested on all the small circuits. The correlation between the detection of classical faults and MOS faults by a given test vector, and the small differences between the MOS fault coverages $FC_{MOS}(V)$ and $FC_{MOS}(V_{CL})$ were also observed with these smaller scale but exhaustively tested circuits. Although only a few hundred input vectors were used in simulating the larger benchmark cicruits, we believe that the correlation between the detection of classical faults and MOS faults holds for longer simulation runs with more input vectors. This correlation was observed on the exhaustively tested smaller circuits which are themselves representive subcircuits selected from

the benchmark circuits. Moreover, since two out of the four benchmark circuits are designed to be easily extended to larger circuits, and many VLSI designs are formed by replicating subcircuits, the correlation likely exists for larger circuits.

From the above analysis of fault simulation data, it is observed that the test vector set V_{CL} (generated by performing logic-gate-level classical fault simulation with the random input vector set V) can provide MOS fault coverage approximately equal to that obtained by test vector set V_{MOS} (generated by performing MOS fault simulation on the corresponding transistor level circuit with the same set of input vectors V). Since simulating classical faults on logic-gate-level circuit models is far less computationally expensive than the corresponding simulation on transistor-level MOS models, using logic-gate-level classical fault simulation to generate test vectors is an attractive alternative for MOS fault simulation. However, using classical fault simulation alone is not sufficient for MOS test vector generation. This is because once the test vector set, V_{CL}, was genereated by the classical fault simulation, the corresponding MOS fault coverage obtainable by this vector set still remains unknown.

One possible approach in generating test vectors for MOS circuits consists of the combined use of classical and MOS fault simulations. In this approach, test vectors, V_{CL}, are generated from logic-gate-level classical fault simulation with input vector set, V. The classical vector set V_{CL} is then applied to simulate transistor-level MOS faults for obtaining the corresponding MOS fault coverage. Before this approach can be considered effective, the individual cost for performing this approach versus a complete MOS fault simulation with input vector set, V, must be compared.

Let $T_{CL}(V)$ be the cost of performing classical fault simulation with input vector set V, $T_{MOS}(V)$ and $T_{MOS}(V_{CL})$ represent the MOS fault simulation costs associated with applying vector sets V and V_{CL} respectively; the cost ratio, CR between complete MOS fault simulation and this approach is given as:

$$CR = \frac{T_{MOS}(V)}{T_{CL}(V) + T_{MOS}(V_{CL})} .$$

In this expression, the numerator is the cost for a complete MOS fault simulation, whereas the denominator accounts for the costs of performing the combined approach. Dividing both the numerator and the denominator by $T_{MOS}(V)$, the above equation can be expressed in terms of the cost ratio between classical and MOS fault simulations, and the cost ratio between simulating vectors sets V_{CL} and V at the transistor level:

$$CR = \frac{1}{\frac{T_{CL}(V)}{T_{MOS}(V)} + \frac{T_{MOS}(V_{CL})}{T_{MOS}(V)}} .$$

Assume the cost functions above are measured by the computation time of each simulation task, the cost ratios between the classical and MOS fault simulations can be estimated by:

$$\frac{T_{CL}(V)}{T_{MOS}(V)} \approx \left[\frac{F_c}{F_{MOS}}\right] \times \left[\frac{t_{CL}}{t_{MOS}}\right] .$$

The term $\frac{F_c}{F_{MOS}}$ (see Table 4) is the number of logic-gate-level classical faults over the number of transitor-level MOS faults in the corresponding circuit models. The term $\frac{t_{CL}}{t_{MOS}}$ (see Table 4) represents the time required to simulate a classical fault versus a MOS fault. On the other hand, the cost ratio between simulating vector sets V_{CL} and V in MOS fault simulation is proportional to the relative size of the two vector sets. Thus the ratio can be approximated by:

$$\frac{T_{MOS}(V_{CL})}{T_{MOS}(V)} \approx \frac{V_{CL}}{V} .$$

Applying the average values obtained from the test circuits, CR is measured to be approximately two and four for the NMOS and CMOS circuits respectively. While this analysis indicates that a two to four speedup can be achieved, this assumes that relatively low resultant fault coverages are acceptable. When high coverages (above 90%) are desired, the speedup that can be achieved using this approach is lower (about 20%), however, may still be worthwhile.

5. Conclusion

In this paper, the problem of generating test vectors for MOS circuits using logic-gate-level classical fault analysis was investigated. Logic-gate-level classical and transistor-level MOS fault simulation experiments were performed on four relatively small benchmark circuits and the experimental data were analyzed. The simulation data confirms the general literature observation that classical fault coverage at logic-gate-level does not closely correspond to MOS fault coverage at transistor-level. The average differences between the classical and MOS fault coverages were recorded to be 17% and 37% for NMOS and CMOS circuits respectively over the range of the benchmark circuits.

However, the empirical results also show that fault coverage, $FC_{MOS}(V)$, obtained from MOS transistor-level fault simulation using randomly generated test inputs can be approximated by the fault coverage, $FC_{MOS}(V_{CL})$, obtained using the test vectors generated from classical stuck–at fault simulation on logic–gate–level circuits. The difference between the two MOS fault coverages described above was measured to be less than ten percent. This relatively small difference in the fault coverages obtained from the test vectors generated by the two types of simulations, and the ratio between the total number of potential logic-gate-level classical and transistor-level MOS faults suggests that logic-gate-level classical fault simulation may be used as a slightly less accurate but more cost effective alternative to complete MOS fault simulation (i.e., the MOS fault simulation with the full set of random vectors, V).

Based on the above results an alternative approach to MOS fault simulation was introduced. This approach uses classical fault simulation on the appropriate logic-gate-level abstraction of a MOS circuit to generate the set of test vectors, V_{CL} (that detect classical faults), from a set of random input vectors, V. A logic-gate-level abstraction of a MOS circuit can be obtained by substituting the transistors in the corresponding MOS circuit with logic gates (i.e., NAND, NOR, NOT etc.) and three-terminal switches (pass transistors). After acquiring the test vector set V_{CL}

from logic-gate-level classical fault simulation, this set of test vectors is applied to the MOS circuit to obtain the corresponding MOS fault coverage. This alternative results in a computation time speedup of approximate two to four (two for NMOS, four for CMOS) for relative low coverages and about 20% for fault coverages over 90%.

6. Bibliography

1. Manir, Tulin E., "Source of Failure and Yield Improvement for VLSI and Restructurable Interconnects for RVLSI and WSI: Part I - Source of Failures and yield improvement for VLSI," *Proceedings of the IEEE, vol. 72, pp. 690-708, June, 1984.*
2. Galiay, J., Crouzet, Y., and Vergniault, M. "Physical Versus Logical Fault Models MOS LSI Circuits: Impact on Their Testability," *IEEE Transactions on Computers,* vol. C_29, pp. 51-54, June, 1982.
3. Miczo, Alxender, "VLSI: How Do I Know It Works?," *Technical Report, Honeywell Information Systems, Large Information Systems Division.*
4. Wadsack, R. L., "Fault Coverage in Digital Integrated Circuits," *The Bell System Technical Journal,* vol. 57, pp. 1475-1488, May-June, 1978.
5. Kohavi, Zvi, *Switching and Finite Automata Theory,* McGraw Hill, 1978.
6. Elziq, Yacoub M., "Automatic Test Generation for Stuck-Open Faults in CMOS VLSI," *Proceedings of the 18th Design Automation Conference,* pp. 347-354, Nashville, June, 1981.
7. Wadsack, R. L., "Fault Modeling and Logic Simulation of CMOS and NMOS Integrated Circuits," *The Bell System Technical Journal,* vol. 57, pp. 1449-1474, May-June, 1978.
8. Timoc, C., Buehler, M., Griswold, T., Pina, C., Stott, F. and Hess, L., "Logical Models of Physical Failure," *Proceedings of the International Test Conference,* pp. 546-553, 1983
9. Kodandapani, K. L. and Pradhan, Dhiraj K., "Undetectability of Bridging Faults and Validity of Stuck-At Fault Test Sets," *IEEE Transactions on Computers,* C-29, pp. 55-59, January, 1980.
10. Mei, Kenyon C. Y., "Bridging and Stuck-At Faults," *IEEE Transactions on Computers,* C-23, pp. 720-727, July, 1974.
11. Lala, Parag K., *Fault Tolerant & Fault Testable Hardware Design,* Prentice-Hall International, 1985.
12. Barzilai, Z., Carter, J. L., Iyengar, V. S., Nair, I., Rosen, B. K., Rutledge, J. and Silberman, G. M., "Efficient Fault Simulation of CMOS Circuits with Accurate Models," *Research Report, IBM Thomas J. Watson Research Center,* Yorktown Heights, May, 1986.
13. Goel, Prabhakar, "Test Generation Costs Analysis and Projections," *Proceedings of the 17th Design Automation Conference,* pp. 273-280, June, 1980.
14. Waicukauski, John A., Eichelberger, Edward B., Forlenza, Donato O., Lindbloom, Eric and McCarthy, Thomas, "Fault Simulation for Structured VLSI," *VLSI Systems Design,* pp. 20-32, December, 1985.
15. Agrawal, Vishwani D. and Agrawal, Prathima, "An Automatic Test Generation System for Illiac IV Logic Boards," *IEEE Transactions on Computers,* pp. 1015-1017, September,

1972.

16. Chamberlain, Roger D., "Lsim: A Gate-Switch Level Logic Simulator," *Technical Report WUCCSD-85-13*, M.S. Thesis, Department of Computer Science, Washington University, St. Louis, Missouri, May, 1985.

17. Chamberlain, Roger D. and Franklin, Mark A., "Collecting Data About Logic Simulation," *IEEE Transactions on Computer-Aided Design*, CAD-5, pp. 405-412, July, 1986.

18. Gill, Authur, *Applied Algebra for the Computer Sciences*, Prentice-Hall, Inc., 1976.