Cortes, M., E.J. McCluskey, K.D. Wagner, and D.J. Lu, "Properties of Transient Errors Due to Power Supply Disturbances" ISCAS'86
Wang, L.T., and E.J. McCluskey, "Concurrent Built-In Logic Block Observer (CBILBO)" ISCAS'86
Cortes, M., and E.J. McCluskey, "Modeling Power-Supply Disturbances in Digital Circuits" ISSCC'86
Liu, D., and E.J. McCluskey, "Design of CMOS VLSI Circuits for Testability" CICC'86

CRC Technical Report No. 86-1
(CSL TN No. 86-287)
March 1986

CENTER FOR RELIABLE COMPUTING
Computer Systems Laboratory
Departments of Electrical Engineering and Computer Science
Stanford University
Stanford, California 94305 USA

ABSTRACT

This Technical Report contains preprints and reprints.
TABLE OF CONTENTS

Title                                                                 Page
Properties of Transient Errors Due to Power Supply Disturbances; by  M. Cortes, E.J. McCluskey, K.D. Wagner, and D.J. Lu, IEEE Int'l Conf. on Circuits and Systems, San Jose, CA, May 5-7, 1986 .......................... 5
Concurrent Built-In Logic Block Observer (CBILBO); by L.T. Wang  and E.J. McCluskey, IEEE Int'l Conf. on Circuits and Systems, San Jose, CA, May 5-7, 1986 .......................... 9
CALCULATION OF THE COVERAGE PARAMETER FOR THE RELIABILITY MODELING OF FAULT-TOLERANT COMPUTER SYSTEMS

Hassanein Amer and Edward J. McCluskey

Center for Reliable Computing
Computer Systems Laboratory, Stanford University
Stanford, CA 94305, USA

ABSTRACT

Programs for calculating the reliability of fault-tolerant systems do not explicitly take into account the effect of failures in the hardware recovery mechanisms. This paper shows how to incorporate the failures in the recovery mechanism of a simple redundant system, in the fault-handling (coverage) model of CARE III and how to calculate the required coverage parameters, specifically the probability that a failure is not lethal. It is also shown that CARE III gives a conservative estimate of the reliability of the redundant system.

1 INTRODUCTION

Analytical models have been developed to estimate the reliability of computer systems. These models can be applied to a large class of fault-tolerant systems [Bridgeman 84] [kg 80]. The user must calculate the required parameters for these models. One of these parameters is the coverage, the conditional probability of successful error recovery given that an error has occurred. Error recovery consists of error detection, isolation and system reconfiguration. The sensitivity of the reliability to a small error in the coverage estimation is well known [Arnold 72]. The reliability of the hardware responsible for the error recovery has to be taken into account [Losq 76] [Ogus 74].

CARE III is a well-known analytical model for reliability calculation. It has a separate model for the coverage where it is assumed that the isolation of a detected error and the recovery from it will always be successful [Bavuso 84] [Geist 83] [Trivedi 81] [Trivedi 83]. Thus, coverage in CARE III consists only of error detection. Furthermore, CARE III cannot model latent faults in stand-by spare modules or recovery mechanisms. Latent faults are faults that will not generate errors until a fault occurs in the active module.

In this paper, a hypothetical fault-tolerant system is designed to point out the difficulties encountered in the calculation of the coverage parameter(s). The failures in the hardware recovery mechanism are included in the coverage model of CARE III.

In Sec. 2, the system is described. The results of logic simulation on a Daisy Megalogician (CAD system) are reported. The faults are divided into different classes and some examples are given to show the effect of the faults in the recovery mechanism on the reliability of the system. The fault classes resemble those described in [Losq 76]. Only permanent faults are modeled.

In Sec. 3, the coverage parameters and the reliability of the system are calculated using the models in CARE III. Special attention is given to the calculation of "e" or the probability of a failure not being lethal to the system [Bavuso 84]. The reliability calculated using CARE III is then compared to a reliability prediction obtained from a Markov model specific to the system under study. It is shown that the CARE III models produce a conservative estimate of the reliability.

Throughout this paper, a failure will refer to a physical defect in a component while a fault will be the model describing that failure. On the other hand, an error occurs when a component performs its function incorrectly.

2 THE FAULT-TOLERANT SYSTEM

The fault-tolerant system consists of two identical modules (X and Y). Module X is active (i.e., connected to the bus) in the error-free condition. A detector controls a switch that connects module Y to the bus in case of an error in X. So, Y is a powered back-up spare in the system.

Figure 1 shows the system. It consists of an OR gate whose function is replicated by a NAND gate and two inverters. An EXCLUSIVE-OR gate compares the outputs of the OR and NAND gates to detect any error. These two outputs are connected to the bus through two buffers with 3-state outputs. The switch is implemented by a D latch that is initially set so that module X is connected to the bus. If the EXCLUSIVE-OR gate detects a discrepancy between the outputs of the OR and NAND gates, the switch disconnects module X from the bus and connects module Y.

This system could be made much more reliable by having two redundant switches as in the Bus Guardians of the FMP [Hopkins 78]. Also, both modules could be systematically exercised or "flexed" to detect latent faults. The emphasis in this paper is on the calculation of the parameters necessary for the reliability models, not on the design of reliable systems.

The fault model used in this analysis will be a permanent single stuck-at fault model. The faults are divided into five classes in order to calculate the coverage parameters. A DAISY Megalogician was...
used to perform the logic simulation of the system and to determine the classification of the faults. Three of these classes correspond to faults in modules X and Y:

1) Undetectable faults: Since a fault in lead A, for example, will have the same effect on both inputs of the EXCLUSIVE-OR gate, it is undetectable and the system fails (incorrect data on the output bus).

2) Detectable faults: A fault in lead E, for example, will only affect one of the inputs of the EXCLUSIVE-OR gate. It will be detected and the switch will connect module Y to the bus.

3) Fatal faults: C s-a-o, for example, will force incorrect data on the output bus and the system will immediately fail.

4) Faults causing premature switching: J s-a-1, for example, will cause module Y to be connected to the bus even though module X is fault-free.

5) Latent faults: A latent fault in the recovery mechanism will not produce an error until a fault occurs in the active module. J s-a-o, for example, will not produce an error until the EXCLUSIVE-OR gate detects an error in module X.

There are also fatal faults in the switch. If lead P is s-a-o, for example, both modules (X and Y) will be disconnected from the bus and the system fails.

The fault classification is shown in Table 1. The faults are divided into ten groups. Each row in Table 1 corresponds to a group and each group corresponds to one of the classes described above. More than one group can correspond to the same class.

Groups 2, 8, and 10, for example, consist of fatal faults. The total number of faults in group i is equal to C(i).

Table 1 Fault Classification

<table>
<thead>
<tr>
<th>Group C(i)</th>
<th>s-a-0</th>
<th>s-a-1</th>
<th>class</th>
</tr>
</thead>
<tbody>
<tr>
<td>Module X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>undetectable</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>5</td>
<td>detectable</td>
</tr>
<tr>
<td>3</td>
<td>6</td>
<td>7</td>
<td>premature</td>
</tr>
<tr>
<td>4</td>
<td>8</td>
<td>9</td>
<td>switching</td>
</tr>
<tr>
<td>5</td>
<td>10</td>
<td>11</td>
<td>switching</td>
</tr>
<tr>
<td>6</td>
<td>12</td>
<td>13</td>
<td>switching</td>
</tr>
<tr>
<td>7</td>
<td>14</td>
<td>15</td>
<td>premature</td>
</tr>
<tr>
<td>8</td>
<td>16</td>
<td>17</td>
<td>switching</td>
</tr>
<tr>
<td>9</td>
<td>18</td>
<td>19</td>
<td>switching</td>
</tr>
<tr>
<td>10</td>
<td>20</td>
<td>21</td>
<td>fatal</td>
</tr>
</tbody>
</table>

3 RELIABILITY CALCULATION USING CARE III

CARE III is a very sophisticated and powerful reliability model (Bridgman 84). It can be divided into two parts: the aggregate model and the fault-handling (coverage) model. The latter describes the recovery process in detail. More information about CARE III can be found in [Bavuso 84] [Trivedi 81].

Figure 2 shows the single fault-handling model. A fault (with rate f(t)) causes the system being modeled to go to state A. The fault is active and no error exists yet. The fault produces an error (at a rate r(t')) and the system goes from state A to state A3. If the error is not fatal, the system will go to state A4 (at a rate E(3)) and with a conditional probability c). If the error is fatal, the system will go to state F. Both t' and t" are random variables. It is assumed that they follow the exponential distribution. 1/r(t') is the average time for a fault to produce an error and 1/E(3) is the average time for that error to be detected (or cause a system failure). State A4 indicates that the error was detected; it is assumed in CARE III that the isolation of the error and the recovery from it, will always be successful.

Only permanent faults will be considered here. The fault-handling model should be able to represent all the faults in the 1-out-of-2 system under study. The latent faults (J s-a-o for
example) cannot be handled like the other faults. A latent fault will only affect the system after an error in module X. The parameter r(t) (time for fault to produce an error) will be many orders of magnitude larger than that of a detectable fault in module X for example. The system could be divided into two subsystems: 1) Modules X and Y. 2) The recovery mechanism. The double fault-handling model in CARE III [Bavuso 81] can be used to describe the dependence between the faults in the two subsystems. However, it will be impossible to distinguish the latent faults from the rest of the faults in the recovery mechanism. A solution for this problem is to divide the faults in the system into two types: 1) Latent faults. 2) All other faults in the system. Hence, the fault-handling model has to be used twice.

A : Fault is active
A_0 : Fault caused an error
A_D : Module has been detected as faulty
F : System failure state
r(t) : Failure rate
r(t') : Rate at which fault produces an error
E(t') : Rate at which error is detected
c : Probability that a faulty module is not lethal

Fig. 2 CARE III Single Fault-Handling Model

Fig. 3 CARE III applied to system in Fig 1

In Fig. 3, the fault-handling model is applied to the faults in modules X and Y as well as the non-latent faults in the recovery mechanism. The parameter c (probability that the system can recover from the error) will be equal to the ratio of the non-fatal faults to the total number of faults (non-latent).

\[
c = \frac{\text{Sum } C(i)}{\text{Sum } C(i) + \text{c}} = \frac{3,4,6,9}{3,4,6,8,9,10}
\]

Since the system is very simple and the clock cycle is many orders of magnitude smaller than the mean time between faults, the parameters \(r(t')\) and \(E(t')\) are both assumed to be large and constant, i.e., the transitions from fault (state A) to error (state A_0) and that from error to recovery (state A_P) or system failure (state F), are almost instantaneous.

\[z \cdot IC[i] \quad r(t') \quad E(t') \quad A \quad A_0 \quad A_P \quad F\]

\[i = 5,7\]

Fig. 4 Modeling the latent faults of the system in Fig 1

The latent faults in the recovery mechanism are treated separately as shown in Fig. 4. The parameter c, in this case, is equal to zero because any fault in the system, while the recovery mechanism is disabled, will lead to a system failure. \(E(t')\) is assigned a large constant while \(r(t')\) is equal to the transition rate between states 0 and A in Fig. 3.

The reliability is calculated as follows [Trivedi 81]:

\[
\text{Reliability}(t) = 1 - P(\text{being in state F at time t})
\]

Unreliability

\[x \cdot 10^{-6}\]

Fig. 5 Unreliability of the system

\[z \cdot IC[i] \quad \text{Mod. X has failed and was disconnected from bus.} \quad z \cdot IC[i] \quad \text{Mod. Y connected to bus.}\]

\[i = 3,4,6\]

\[z \cdot IC[i] \quad \text{Non-fatal fault in Mod. Y} \quad z \cdot IC[i] \quad \text{System failure}\]

\[i = 1,2,3,4,5,6,7,8,9,10\]

\[z \cdot IC[i] \quad \text{Fault-free state} \quad z \cdot IC[i] \quad \text{Latent fault in switch}\]

\[i = 1,2,3,6,8,10\]

\[z \cdot IC[i] \quad \text{Non-fatal fault in Y with latent fault in switch}\]

\[i = 1,2,3,6,8,10\]

Fig. 6 Reliability model specific to system in Fig 1
The reliability of the whole system will be the product of the reliabilities calculated from the above two models (Figs. 3 and 4).

Unreliability(s) = 1 - [1 - P(SF non-latent)] [1 - P(SF latent)]

where SF non-latent denotes the event of a system failure due to a non-latent fault and SF latent denotes the event of a system failure due to a latent fault. Let Z be the fault-handling factor of any gate, latch, or fanout branch in the circuit and assume that the s-a-0 and s-a-1 faults are equally likely, each with a failure rate of \( z \). The unreliability is shown in Fig. 5 for \( z = 0.0001 / 10^6 \). A reliability model specific to the system under study (Fig. 6).

5 SUMMARY and CONCLUSIONS

A very simple redundant system was designed. It consists of two identical modules, one active and the other a powered back-up spare. The recovery mechanism consists of a detector and a switch. The faults were divided into five classes assuming single stuck-at fault model. The classification of the faults was determined by a Daisy Megalogician. The five fault classes were then used to determine the coverage parameters for the fault-handling model in CARE III. Even though only one type of failure (permanent) was modeled, it was found that the fault-handling (coverage) model had to be used twice to account for the latent failures in the recovery mechanism. Each time the model was used, a different set of coverage parameters was calculated.

A model of the system was also built and the results were compared to those obtained with the models in CARE III. It was found that CARE III gives a conservative estimate of the reliability of the system.

In conclusion, the failure in the recovery mechanism made it necessary to "adapt" CARE III in order to accurately represent the system. Even though CARE III does not distinguish between active and stand-by spare modules, it gives a conservative estimate of the reliability of a stand-by redundant system.

ACKNOWLEDGMENTS

This work was supported in part by the Department of the Army under Contract Number DAAG-62-KO105 and by the Egyptian Government. The logic simulation was performed on a Daisy Megalogician workstation made possible by Daisy Systems Co. (Mountain View, CA). The views, opinions, and/or findings contained in this document are those of the authors and should not be construed as an official Department of the Army position, policy, or decision, unless so designated by other official documentation.

REFERENCES


This paper shows that failures caused by power supply disturbances can be modeled as delay faults. This conclusion results from experiments where voltage sags are injected in the power supply lines of gate arrays and breadboard circuits. The susceptibility of the circuits to the occurrence of errors increases with clock frequency. This dependency can be attributed to the increase in propagation delay with lower power supply voltages. Errors are caused by violation of timing constraints of the circuits. It was also found that supply disturbances can cause metastability.

1 INTRODUCTION

Power supply disturbances are known to cause errors in the operation of digital systems. In the literature ([Allen 83] [Chesney 83]), the susceptibility of circuits to power supply disturbances has been characterized by measurements where logic gates with constant input signals have their power supply disturbed. By using constant inputs, the important effect of power disturbances on propagation delay is underestimated and noise immunity problems are assumed to be the only cause of errors. This is not a reasonable assumption in systems where logic signals are changing with time. Experimental results show that propagation delay variation is the dominant effect and that noise immunity plays a small role in error occurrence. This paper shows that failures caused by power supply disturbances can be modeled as delay faults. Experiments were performed on a CMOS gate array and breadboard circuits implemented with 74HC and 74LS catalog parts. It was found that the susceptibility of the circuits to power supply voltage disturbances is related to the operating frequency. Errors are more likely to occur as operating frequency increases. In the breadboards, the error mechanism was observed directly by monitoring voltage waveforms at internal nodes. It was found that, in most cases, errors were caused by violation of timing constraints during the increase of gate propagation delay due to the disturbances. In the gate array, internal waveforms could not be observed directly. In order to analyze the internal behavior, logic simulation was performed on a gate array model. All output waveforms observed experimentally were successfully reproduced by the logic simulation, confirming that errors were caused by delay effects. In another experiment, a metastability detector was built in order to analyze the output waveforms of the circuits under test. It was found that power supply disturbances can cause metastability. The circuits implemented for the experimental work are described in Sec. 2. The experimental procedure and data analysis is presented in Sec. 3 followed by the conclusions in Sec. 4. Preliminary results on power supply disturbances are presented in [Lu 84] and [Cortes 85].

2 DESCRIPTION OF THE EXPERIMENTAL SYSTEM

The circuits utilized in the experiment are the detector chip, the detector breadboards and the metastability detector. A description of the chip and the hardware utilized in the experiment can be found in [Lu 84] and [Wakerly 82]. A brief description of the circuits is presented next.

2.1 DETECTOR CHIP. This chip was proposed by McCluskey [81]. Its sole purpose is to monitor itself for temporary errors. In this experiment, the detector chip used is a CMOS gate array fabricated by STC (Storage Technology Corporation). Figure 1 shows the elements of the detector chip and their interconnections. The circuit consists of a set of three XORs wired in such a way that it has the following interesting properties: the complete test set consists of all combinations of even-weight 3-bit vectors; any even-weight input vector produces the same vector at the output; any odd-weight input vector produces an even-weight output vector; therefore when the complete test set (Fig. 1) is applied, the output lines of the cell match their inputs. Fifteen basic cells are cascaded to form a chain. If the circuit is error-free the output of a chain is a delayed version of the input patterns. A set of Flip-Flops is added to synchronize the signals. An equality checker built out of 3 XORs and an AND gate provides an active-low error signal denoted LAE for Look-Ahead Error.

2.2 DETECTOR BREADBOARDS. In order to permit the observation of internal signals, a discrete version of the detector chip was built. It consists of one module made of 10 basic cells (Fig. 2). A clock generator, a 3-channel programmable pattern generator and a voltage level translator (open collector inverters 5420) are also included on the same board. The power supply disturbances are applied to the circuit under test (Fig. 2). The rest of the board is supplied with 5 V DC. The clock and pattern generators are
implemented with 74LS parts. The portion of the board that is subjected to disturbances (circuit under test, Fig. 2) was implemented with 74HC catalog parts in the CMOS breadboard and with 74LS catalog parts in the LSTTL breadboard.

![Detector Circuit and its Building Blocks](image)

Fig. 1: The Detector Circuit And its Building Blocks

2.3 METASTABILITY DETECTOR. This circuit is actually a late transition detector and was based on a circuit suggested by Greg Freeman (Fig. 3) [Freeman 85]. An error signal is generated whenever a data transition occurs after DLYCLK rising edge. The counter contents indicate the error rate. This circuit was implemented in high-performance CMOS parts (74HC). This is an appropriate scheme to detect metastability when it is impossible to access the output of the metastable memory element. This is the case for the detector chip and the breadboard detectors where all signals are buffered. When metastable signals (metastable voltage levels) drive the buffers, their outputs are likely to be observed externally as valid logic signals because of the high gain of the buffers. Half of the metastability occurrences result in late transitions at the buffer output.

3 EXPERIMENTAL RESULTS AND ANALYSIS

In this section the various experimental procedures are presented and the data is analyzed. A sequence of different experiments was performed on both the detector chip and detector breadboards. 3.1 DETECTOR CHIP. Two types of disturbances were applied: DC and Pulsed (negative pulses). The susceptibility of the chip to disturbances was measured by increasing the magnitude of the disturbance until the first error was observed at the LAE output pin. The magnitude of the disturbance is denoted \( \Delta V_{DD} \). Low values of \( \Delta V_{DD} \) indicate poor tolerance to disturbances.

![Metastability Detector](image)

Fig. 3: The Metastability Detector

DC vs Pulsed. A first set of experiments was conducted to study the effect of duration of the disturbances on error occurrence. It was found that \( \Delta V_{DD} \) is insensitive to pulse-width variations (1, 2, 4 and 7 microsec. were used). Furthermore, the same value of \( \Delta V_{DD} \) was measured for DC disturbances. This is not surprising since propagation delays and transients are much shorter than 1 microsecond. \( \Delta V_{DD} \) may exhibit a non-negligible sensitivity to pulse width for very short duration disturbances (100 nsec or shorter). However, voltage sags with such short duration are very rare [Key 78]. Therefore, it was reasonable to limit the experiments to DC disturbances only, and extend the results to typical pulsed disturbances. The remaining experiments were conducted using DC only.

DC Disturbances. Figure 4 shows the variation of the tolerance to disturbance (\( \Delta V_{DD} \)) with clock frequency for the CMOS gate array. The nominal supply voltage is 5 V. One can identify two regions in the plot: 1) a flat region (\( \Delta V_{DD} \approx 3.4 \) V) for frequencies below 2.5 MHz, that shows a weak dependency on frequency; and 2) a region where the tolerance to disturbances decreases monotonically with frequency (from 2.5 MHz up to 12 MHz) that will be referred to as the frequency-dependent region.

At low frequencies (flat region) it was originally thought that errors occurred only because of noise margin violations: \( V_{DD} \) was too low to guarantee the \( V_{IH} \), \( V_{IL} \) for the CMOS gates.
There is now evidence that other effects may contribute to this behavior as well (Section 3.2). In the frequency-dependent region, delay effects were dominant. Errors were caused by violations of timing constraints due to the increase of gate propagation delay during the disturbances. For example, at 10 MHz and for disturbances larger than 2.3 V (Fig. 4), the delay through the combinational logic of the detector chip (the chain in Fig. 1) is longer than the clock period (100 nsec.). The increase in propagation delay with lower supply voltages occurs in CMOS and LSTTL logic and is discussed in (Wagner 85).

Internal nodes of the gate array could not be observed directly. In order to confirm that the observed output waveforms resulted from delay effects, a logic simulation model of the gate array was implemented on a Megalogician Logic Simulator (Daisy Systems Corp.). In the simulation model, the gates have a constant delay, the clock frequency is varied and errors occur because of timing constraint violations. This is equivalent to having the clock frequency fixed and varying the gate delay parameter in the logic simulation model. In this way, the external behavior of a circuit-level dependency (delay vs power supply) can be studied with a logic level equivalent model (timing constraint violation). The analysis of the circuit was substantially simplified by the use of the logic simulation tool. All the output waveforms observed experimentally were successfully reproduced by the logic simulation. This confirms that transient errors due to power supply disturbances, in the frequency-dependent region, can be modeled as delay faults.

CMOS AND LSTTL BREADBOARD DETECTORS. Similar experiments were performed on breadboard versions of the gate array: the CMOS BreadBoard (74HC catalog parts) and the LSTTL BreadBoard (74LS TTL catalog parts). The breadboards allow reconfiguration of the chain (changes in chain length) and easy observation of the error mechanisms. The results are presented in Fig. 5 for the CMOS breadboard and in Fig. 6 for the LSTTL breadboard. The data points labeled 10 stages, 4 stages were obtained on CMOS breadboards having different chain lengths (number of cells); the disturbance magnitude ($\Delta V_{DD}$) was increased and the error output LAE was observed for errors. In the data points labeled Input Error, $\Delta V_{DD}$ is the disturbance magnitude to cause an incorrect vector at $S0$ (Chain input in Fig. 2). The CMOS and LSTTL breadboards exhibit a similar behavior to the gate array (Fig. 4).

![Fig. 4: Tolerance To Power Supply Disturbances vs Frequency for the Gate Array](image)

![Fig. 5: Tolerance to Power Supply Disturbances vs Frequency for the CMOS breadboard](image)

![Fig. 6: Tolerance to Power Supply Disturbances vs Frequency for the LSTTL breadboard](image)

The error mechanism was observed by probing the breadboards. Increases in disturbance magnitude ($\Delta V_{DD}$) caused the signal transitions at the chain output ($S10$ in Fig. 2) to approach the clock edge. An error was detected by LAE when setup time was violated, at $S10$. This observation led to the conjecture that metastability can occur.

The shape of the curves depend on the variation of delay-per-gate with supply voltage and length of the longest delay path in the system. When the chain length in the CMOS breadboard was changed from 10 to 4 stages (Fig. 5), the tolerance to disturbances increased from 1.20 V to 2.57 V (at 8 MHz).

In the flat region, as the disturbance increased, erroneous input vectors appeared at the chain input ($S0$, Fig. 2). This erroneous injection...
was responsible for the shape of the flat region in the plots. This was caused by timing degradation in the CMOS breadboard and noise immunity problems in the LSTTL breadboard.

3.3 METASTABILITY. Measurements with the CMOS breadboard (Section 3.2) showed that the conditions for metastability were present. For some combined frequency and voltage conditions, set-up time violations were observed. The goal of this experiment was to detect metastability using the circuit described in Section 2 (Fig. 3). No attempt was made to fully characterize the metastability. Metastability was detected in both the CMOS breadboard and the detector chip. For simplicity, only the CMOS breadboard experiment is described here. The signals labeled $X_{OUT}$ and $Y_{OUT}$ (Fig. 2) were connected (one at a time) to the data input of the metastability detector (Fig. 3). Both circuits used the same clock signal. The window in Fig. 3 represents the minimum duration of metastable states which would cause an error count. For $4.0$ MHz and $V_{DD} = 2.32$ V metastability was detected. Note that the voltage drop is precisely $V_{DD}$ for $4.0$ MHz in Fig. 5. The occurrence was very sensitive to voltage and frequency. Any drop in the power supply voltage caused the condition to disappear. The error rates observed (late transitions count) are shown in Table 1.

**TABLE 1: Metastability Error-Rate versus Duration**

<table>
<thead>
<tr>
<th>window (nsec)</th>
<th>rate (late transitions / sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>40</td>
<td>very high</td>
</tr>
<tr>
<td>60</td>
<td>320</td>
</tr>
<tr>
<td>80</td>
<td>0</td>
</tr>
<tr>
<td>100</td>
<td>0</td>
</tr>
</tbody>
</table>

This experiment shows that metastability can occur in fully synchronous systems as well.

4 SUMMARY AND CONCLUSIONS

One of the consequences of the power-supply/delay dependency is that computer systems designed with tight timing will tolerate only very small power supply disturbances. As an example, suppose the CMOS breadboard is operating with a 10% delay margin at 5 V / 9.3 MHz. It will tolerate a maximum voltage dip of 0.5 V (Fig. 5). This tolerance may be further reduced by other factors like parameter variations, temperature variation, noise, etc. In applications where speed is not critical, the lowest clock frequency to meet the specifications should be used, thereby improving the tolerance to power disturbances. Digital systems used for process control in industrial plants are examples of such systems. When general purpose computer systems are used, they are exposed to unnecessary risk because usually the clock frequency is factory preset to obtain high performance (tight timing). Evidence of metastability was a byproduct of the experiments. It was shown that metastability can occur in a fully synchronous environment. The experimental results and conclusions presented here can be extended to other logic families with supply-voltage / propagation-delay dependency [Wagner 85].

ACKNOWLEDGEMENTS

This work was supported in part by International Business Machines (IBM) under a contract with Palo Alto Research Associates (PARA), Palo Alto, CA 94303, in part by the Brazilian National Commission for Nuclear Energy (CNEN) and in part by the Innovative Science and Technology Office of the Strategic Defense Initiative Organization administered through the Office of Naval Research under Contract No. N00014-85-K-0600. The authors are grateful to John Iazio, now with AIDA Corporation, for his help in producing the detector CMOS gate-array chip. The logic simulation was performed on a Megalogician workstation provided by Daisy Systems Co. (Mountain View, CA).

REFERENCES

CONCURRENT BUILT-IN LOGIC BLOCK OBSERVER (CBILBO)

Leung-Terng Wang* and Edward J. McCluskey

CENTER FOR RELIABLE COMPUTING, Computer Systems Laboratory
Departments of Electrical Engineering and Computer Science
Stanford University, Stanford, CA 94305 USA

ABSTRACT

A concurrent built-in logic block observation (CBILBO) technique for on-chip or on-board self-test is presented in this paper. This technique is derived by combining the scan and BILBO techniques together. It allows test pattern generation and response analysis to be performed simultaneously during self-testing. With this approach, test time is reduced to one-half compared to the BILBO approach, and it is much less than with the scan approach.

1 INTRODUCTION

Design for testability (DFT) techniques [McCluskey 84b] can be used to reduce test cost of VLSI/board circuits. Two DFT techniques that are commonly used are scan testing [McCluskey 84b] [Williams 73] [Eichelberger 77] and built-in self-test [McCluskey 85a] [McCluskey 85b] [Wang 85].

Scan testing requires that every D flip-flop (D-FF) and D-latch be reconfigurable into a scan path D-FF (SPFF) [Williams 73] [McCluskey 81] or an LSSD shift register latch (SRL) [Eichelberger 77]. Test patterns and output responses are scanned in and out of the chip or board during test mode. With this approach, sequential logic can be transformed into combinational logic and test logic is self-testable. However, test time can be very long due to the serial scan in-and-out property.

Built-in self-test (BIST) requires that test patterns be applied using on-chip or on-board test pattern generators (TPGs) and output responses be compacted to a single word (signature) using output response analyzers (ORAs). The TPG can be a binary counter, a syndrome driver counter [Barzilai 81], a constant weight counter [McCluskey 84a], a built-in logic block observer (BILBO) [Konemann 80], or a linear feedback shift register (LFSR) [Wang 86]. The ORA is usually a multiple-input or parallel signature analyzer (MISA) [Konemann 80] which is an LFSR with an Exclusive-OR (XOR) gate placed at the data input of each SPFF or SRL [Benowitz 75] [Frohwerk 77]. The TPG and the MISA can be reconfigured from the corresponding input and output registers of the circuit under test [McCluskey 81], respectively. A BILBO can also be employed to serve the above purposes. If BILBOS are used, to achieve 100% single stuck fault coverage, consecutive logic blocks must be either tested at different times or tested alternately as described in [Williams 83].

* L.T. Wang is also with Digital Design Automation, Daisy Systems Corp., Mountain View, CA 94039.

This paper presents a new self-testing scheme so that test time can be reduced to one-half compared to the BILBO approach. The concurrent BILBO (CBILBO) [Wang 85] is constructed in such a manner that, during self-testing, consecutive logic blocks are simultaneously tested. If exhaustive or pseudo-exhaustive test patterns [McCluskey 84a] [Wang 86] are used, the CBILBO can achieve 100% single stuck fault coverage.

2 BUILT-IN LOGIC BLOCK OBSERVER (BILBO)

The structure described in [Konemann 80] applies to circuits that can be partitioned into independent modules (logic blocks). Each module is assumed to have its own input and output registers, or such registers are added to the circuit where necessary. The registers are redesigned so that for test purposes they can act as either autonomous LFSRs (TPGs) for test generation or MISAs for signature analysis. The redesigned register is called a BILBO (Built-in logic block observer).

The BILBO is operated in four modes: normal mode, reset mode, test generation or signature analysis mode, and scan mode. This technique is most suitable for circuits that can be partitioned so that input and output registers of the resulting modules can be reconfigured independently. If consecutive modules have to be tested simultaneously, since the test generation and signature analysis modes cannot be separated, the signature data from the previous module must be used as test patterns for the next module. In this case, a detailed simulation is required in order to achieve 100% single stuck fault coverage.

3 MODIFIED BILBO (MBILBO)

One technique that overcomes the above BILBO problem is described in [McCluskey 81]. It uses an additional control input to separate test generation mode from signature analysis mode, and eliminates propagation gate delay by integrating the additional circuitry into the original flip-flop design.

Such a modified BILBO (MBILBO) design is shown in Fig. 1. It is a revised version of the design given in [McCluskey 81], where only three modes of operation are considered: normal mode, test generation mode, and signature analysis mode. The modification is obtained from the original BILBO by adding one more OR gate to each 2:1 input. The control input B3 is always set to 0 except when the register has to be configured into a TPG. In that case, B3 is set to 1. Fig. 2 shows an MBILBO one-cell structure which integrates the additional
circuitry into a D flip-flop. With this approach, all MBILBO inputs are effectively placed in parallel, and thus no additional gate delay is introduced. There may be some decrease in speed due to increased loading.

<table>
<thead>
<tr>
<th>B1</th>
<th>B2</th>
<th>B3</th>
<th>Operation Mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>Normal</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>Reset</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>Signature analysis</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>Test generation</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>Scan</td>
</tr>
</tbody>
</table>

Fig. 1. A 3-stage modified BILBO (MBILBO).

**4 CONCURRENT BILBO (CBILBO)**

In the MBILBO approach, system registers (either input or output registers) are reconfigured into either TPGs or MISAs, but not both at the same time. To achieve 100% single stuck fault coverage, this requires that consecutive modules be tested at different times or tested alternately. For circuits where test time is a critical parameter, use of the concurrent BILBO (CBILBO) approach presented here can reduce test time to one-half.

The CBILBO combines the design of a TPG and an MISA together. It generates test patterns and compacts output responses simultaneously during self-testing. The CBILBO can be implemented using either D flip-flops (D-FFs) or D-latches.

Fig. 3 shows a 3-stage CBILBO using D-FFs. Each CBILBO stage consists of signature logic (including an AND gate and an XOR gate), a D-FF, and a two-port D-FF. The top 3 D-FFs and signature logic constitute a 3-stage MISA and the bottom two-port D-FFs constitute a 3-stage TPG.

<table>
<thead>
<tr>
<th>B1</th>
<th>B2</th>
<th>Operation Mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>Normal</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>Scan</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>Test generation and Signature analysis</td>
</tr>
</tbody>
</table>

Fig. 2. An MBILBO one-cell structure.

This D-FF type CBILBO is controlled by two control inputs, B1 and B2. When B2=0, the circuit is operated in normal mode. It functions as a parallel read-in register with the inputs Z1 gated directly into the two-port D flip-flops. When B1=1 and B2=1, the register is configured into a serial read-in shift register. Test data can be scanned-in via the serial input port or scanned-out via the serial output port. Setting B1=0 and B2=1 converts the register into a combination of a TPG and an MISA. Fig. 4 shows a CBILBO one-cell structure which integrates the additional circuitry into the flip-flop. With this approach, no additional gate delay is introduced although some speed degradation may still exist.

Fig. 5 shows a 3-stage CBILBO using D-latches. Its functions are controlled by four non-overlapping clocks (SCK, TCK, CK2 and CK3) and two control inputs (B1 and B2). Each CBILBO one-cell structure consists of signature logic (including two AND gates and one XOR gate), and three D-latches (L1, L2 and L3). The L1 and L2 latches act as one TPG.
stage (a two-port D-FF) to generate test patterns to the next circuit under test (CUT). The signature logic and the L1 and L3 latches act as one MISA stage (a D-FF) to compact output responses from the previous CUT. Fig. 6 shows a gate-level design of the CBILBO one-cell structure using D-latches. The structure is very similar to the stable SRL (SSRL) for interfacing LSSD logic to non-LSSD logic [DasGupta 81].

In summary, the D-latch type CBILBO requires three additional non-overlapping clocks compared to the D-FF type CBILBO. However, unlike the CBILBO which uses D-FFs and has an inevitable, essential hazard [McCluskey 86], this CBILBO using D-latches is hazard-free.

5 SUMMARY AND CONCLUSIONS

A modified BILBO (MBILBO) design is first presented. Compared to the original BILBO [Konsmann 80], this MBILBO separates test generation mode from signature analysis mode, and is thus suitable for built-in self-test. By integrating the additional circuitry into the system registers, no additional gate delay is introduced. Design of two kinds of concurrent BILBOs (CBILBOs) using D-flip-flops (D-FFs) and D-latches are then discussed. Both CBILBOs further reduce test time to one-half compared to the BILBO or MBILBO approach.
Table 1 compares various design approaches. It can be seen that the proposed CBILBO (either D-FF type or D-latch type), like the LSSD SRL, the stable SRL (SSRL) [DasGupta 81], and the CBILBO, has no disadvantages. For example, it needs more pins; it increases hardware overhead; and it may introduce additional gate delay during normal operation.

Although there is slight performance degradation due to the addition of extra gate delay in the D-latch type CBILBO, both CBILBOs offer many distinct features. First, they have all the benefits offered by the scan approach. For example, any sequential logic can be transformed into combinational logic and test logic as still testable. Secondly, they allow interfacing LSSD logic with non-LSSD logic as offered by the SSRL approach. Thirdly, neither software test generation nor fault simulation are required as in the CBILBO approach. Finally, test time is much less than with the scan approach, and can be one-half the test time of the CBILBO or MBILBO approach.

ACKNOWLEDGMENTS

The authors wish to thank Professor Samina Moursad, Dr. Amer Mahwood, Dr. Zusi Sun, Greg Freeman, Hua-Ju Wang, and Joe McCluskey for their valuable comments. This work was supported in part by the National Science Foundation under Grant No. MCS-8200129, and in part by Daisy Systems Corporation under the Honors Cooperative Program (HCP).

REFERENCES


*Table 1. CBILBO comparison with other designs.*

<table>
<thead>
<tr>
<th>Parameter vs. Design</th>
<th>CBILBO D-FF</th>
<th>CBILBO D-latch</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pin Count</td>
<td>3</td>
<td>7</td>
</tr>
<tr>
<td>Clock/Test Pins</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>Hardware Overhead</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>Extra Gate Delays</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Test Data Storage</td>
<td>N</td>
<td>1</td>
</tr>
<tr>
<td>Test Time</td>
<td>&gt;=2</td>
<td>2</td>
</tr>
</tbody>
</table>

*: Values are given by ratio. N: Number of test patterns for each scan path.
SESSION XIII: VLSI MODELING AND PACKAGING

THAM 13.4: Modeling Power-Supply Disturbances in Digital Circuit*
Maria L. Cortes, Edward J. McCluskey,
Kenneth D. Wagner, David J. Lu
Stanford University
Stanford, CA

POWER SUPPLY DISTURBANCES are known to cause errors in the operation of digital systems. In the literature, the susceptibility of circuits to power supply disturbances (PSD) has been characterized by measurements where logic gates with constant input signals have their power supply disturbed. By using constant inputs, the important effect of power disturbances on propagation delay is underestimated and noise immunity problems are assumed to be the only cause of errors. In systems where logic signals are changing with time, this is not a reasonable assumption. Experimental results show that propagation delay variation is the dominant effect and that noise immunity plays a small role in error occurrence. It will be shown in this paper that failures caused by power supply disturbances can be modeled as delay faults.

The subjects of the experiments were: a CMOS gate array** and discrete versions of the gate array implemented with catalog parts***. Figure 1 shows the logic diagram of the circuit used in the experiment (detector circuit)†. It consists of cascaded basic cells, synchronizing elements and checking circuitry. When the complete test set for the Exclusive OR (XOR) (Figure 1) is applied to a basic cell its output signals match the input signals. Therefore, if the circuit is error-free the output of a chain is a delayed version of the input pattern and the error output signal (LAE) is inactive.

The experimental conditions were as follows: Voltage dips (negative pulses) were injected in the power rails of the detector circuits while the complete test set was repeatedly applied to its inputs and the error output (LAE) observed with an oscilloscope. For a given operating frequency the magnitude of the disturbances increased until the first error was observed. Only one disturbance was used, pulse duration longer then device propagation delay. This is a reasonable simplification since short power supply disturbances (100ns or shorter) are very rare.

Figure 2 shows the dependency of disturbance magnitude (Δ VPP) on clock frequency for the three circuits: the CMOS gate array (detector chip) and the CMOS (CMOS breadboard) and LSTTL (LSTTL breadboard) discrete versions. The nominal supply voltage is 5V.

*This project was supported in part by International Business Machines (IBM) under a contract with Palo Alto Research Associates (PARA), in part by the Brazilian National Commission for Nuclear Energy (CNEN) and in part by the Innovative Science and Technology Office of the Strategic Defense Initiative Organization and was administered through the Office of Naval Research, ONR Contract N00014-84-K-0600. The logic simulation was performed on a Megalogician workstation made possible by Daisy Systems Co. (Mountain View, CA).

**4LS, 74HC.


One can identify two regions in the plot: (1) a flat region for lower frequencies (below 2.5MHz for the gate array) and (2) a frequency dependent region (above 2.5MHz for the gate array) where the tolerance to disturbances decreases as the clock frequency increases. The three curves exhibit the same behavior: the CMOS breadboard has a small flat region, noticeable in an expanded plot. The shape of the curve depends on the variation of delay per gate with supply voltage and length of the longest delay path in the system.

For low frequencies (flat region), noise immunity problems are one of the causes of the observed errors. In the frequency dependent region, delay effects are dominant. Errors occur because, during the disturbances, gate propagation delay increases and timing constraints are violated. The increase in propagation delay with lower supply voltage occurs in CMOS and LSTTL logic and was discussed earlier.

In the gate array experiment, internal nodes could not be observed directly. To confirm that errors were caused by delay effects, a logic simulation model of the gate array was built. All waveforms obtained experimentally were successfully reproduced by the logic simulation. This confirms that transient errors due to power supply disturbances in the frequency dependent region, can be modeled as delay faults.

In the breadboard experiments, it was possible to observe the error mechanism directly. Increasing disturbances caused signal transitions at the input of Type 2 (Figure 1) to approach the clock edge. An error was detected by LAE when set-up time was violated. This observation led to the conjecture that metastability can occur.

A metastability detector was built (late transition detector) and metastable states lasting fifteen or shorter were detected. This shows that power supply disturbances can cause metastability (even in a fully synchronous circuit).

In another experiment with the CMOS breadboard, the length of the longest delay path in the circuit was changed by reducing the number of basic cells in the chain (Figure 1) from 10 to 1. The shape of the curve (not shown here) varied accordingly and the tolerance to supply voltage disturbance was higher for the shorter path circuit.

One of the consequences of the power-supply/delay dependency is that computer systems designed with tight timing will tolerate only very small power supply disturbances. As an example, suppose the CMOS breadboard is operating with a 10% delay margin at 5V/9.3MHz. It will tolerate a maximum voltage dip of 0.5V. Figure 2. This tolerance may be reduced further by other factors, e.g., parameter variations, temperature variation, noise, etc.

In applications where speed is not critical, the lowest clock frequency to meet the specifications should be used, thereby improving the tolerance to power disturbances. Digital systems used for power control in industrial plants are examples of such systems. When general-purpose computer systems are used they are exposed to unnecessary risk because usually the clock frequency is factory preset to obtain high performance (tight timing).

The experimental results and conclusions presented can be extended to other logic families with power-supply-voltage/propagation-delay dependency.


Basic Cell

Chain

Tri-reg

Detector Circuit

**FIGURE 1**—Detector circuit and building blocks.

<table>
<thead>
<tr>
<th>Vdd (V)</th>
<th>-ΔVdd (V)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>4.0</td>
</tr>
<tr>
<td>1.5</td>
<td>3.5</td>
</tr>
<tr>
<td>2.0</td>
<td>3.0</td>
</tr>
<tr>
<td>2.5</td>
<td>2.5</td>
</tr>
<tr>
<td>3.0</td>
<td>2.0</td>
</tr>
<tr>
<td>3.5</td>
<td>1.5</td>
</tr>
<tr>
<td>4.0</td>
<td>1.0</td>
</tr>
<tr>
<td>4.5</td>
<td>0.5</td>
</tr>
<tr>
<td>5.0</td>
<td></td>
</tr>
</tbody>
</table>

**FIGURE 2**—Tolerance to power supply disturbances vs. frequency.
DESIGN OF CMOS VLSI CIRCUITS FOR TESTABILITY

Dick L. Lu and Edward J. McCluskey

Center for Reliable Computing, Computer Systems Laboratory
Departments of Electrical Engineering and Computer Science
Stanford University, Stanford, California 94305 USA

ABSTRACT

This paper describes a design technique which facilitates testing for stuck-open faults in CMOS VLSI circuits with scan-path. In this technique, the combinational circuitry is implemented with specially designed gates which can be tested with a simplified 2-pattern test for stuck-open faults. The simplified 2-pattern test cannot be invalidated by arbitrary circuit delays and can be applied through the scan-path by specially designed shift register latches.

1 INTRODUCTION

Due to advances in VLSI technology, hundreds and thousands of devices can be fabricated on an IC chip. A VLSI circuit is extremely difficult to test due to its high device-to-pin ratio. To alleviate this testing problem, it is advantageous to use a Design For Testability (DFT) design technique during the design of VLSI circuits.

One example of DFT is the scan-path technique which facilitates IC testing in two ways: (1) scan-paths partition an IC into several blocks which can be tested independently, and (2) scan-path design transforms a sequential circuit into a combinational circuit for which Automatic Test Pattern Generation (ATPG) is relatively straightforward [McCluskey 84]. However, certain faults in a CMOS IC may cause a combinational circuit to exhibit sequential behavior, thus invalidating the sequential-to-combinational transformation of the scan-path design. Such a fault causes an FET to remain nonconducting irrespective of the applied voltage level at the FET gate terminal and is called an FET stuck-open fault (sof fault) [Wadsack 78].

The test for a sof fault in a CMOS logic gate consists of two test patterns: an initializing input pattern and a test input pattern. The initializing input (T1) places the output of a faulty CMOS logic gate at logic-1 to detect an NFET sof fault or at logic-0 to detect a PFET sof fault. The test input (T2) then sensitizes the effect of the fault to the output node of the logic gate and propagates this effect to an observable output. The generation of a 2-pattern test is a nontrivial task.

[Jain 83] and [Reddy 83] have reported that arbitrary delays in the Circuit Under Test (CUT) can invalidate a 2-pattern test. Generation of hazard-free test patterns which are able to detect sof faults even under arbitrary delay conditions is required [Reddy 84]. Not only are test patterns of this nature much more difficult to generate than stuck-at fault patterns, but they are almost impossible to apply to a scan-path IC. This is because the process of shifting in the T2 pattern will alter the state of the CUT established by the T1 pattern.

Several researchers proposed design techniques as solutions to the problem of testing for sof faults [McCluskey 81; Reddy 83; Jha 85; Zasio 85]. However, none addressed the problem of testing sof faults for scan-path ICs adequately. This paper presents a solution to this problem. It is based on a testable CMOS combinational circuit design technique.

2 A TESTABLE CMOS LOGIC CIRCUIT DESIGN

This section describes a technique for designing testable CMOS combinational circuits. The proposed technique is based upon two assumptions about the CUT. First, it assumes that there is only one sof fault in the CUT. Second, the combinational part of the CUT does not contain transmission gates.

2.1 A fully complementary gate structure

A CMOS combinational circuit is constructed by interconnecting CMOS gates. Figure 1 shows a block diagram of a CMOS gate which consists of a pull-up network of PFETs (p-net) and a pull-down network of NFETs (n-net). Most CMOS gates used in a logic circuit are what may be referred to as fully complementary gates. These gates are defined as follows.

[Definition 1] A Fully Complementary Gate (FCG) is a CMOS gate with the following properties: (1) each logic gate input is connected to the transistor gate terminals of both a PFET and an NFET, and (2) the p-net provides conduction paths for all input combinations for which output Z is logic-1; the n-net provides conduction paths for all input combinations for which output Z is logic-0 [Mukherjee 86].

An example of an FCG is shown in Fig. 2. It realizes an AOI gate. Notice that an FCG can realize any combinational Boolean function if double-rail inputs are available.

\[ Z = \text{AND}(A, B, C) \]

Figure 1. A Fully Complementary Gate (FCG).

Figure 2. An AOI gate.
The output node \(Z\) of an FCG can be initialized to logic-0 if all of its inputs are set to logic-1. Similarly, the output Z can be initialized to logic-1 if all the inputs are set to logic-0. (During the initialization, if both \(x\) and its complement \(x'\) are inputs to an FCG, they are assigned the same logic value.)

Normally it is not possible to set both input \(x\) and \(x'\) to the same logic value unless they are primary inputs. However, such an input condition can be established in a testable combinational circuit to be described later in this section.

2.2 Stuck-open fault test patterns

To detect a \(s-o\) fault in an FCG requires two patterns. The initializing input \(T_1\) can be easily determined using property one. The test input \(T_2\) is derived by assigning values to each input variable such that it can (1) turn on the faulty FET \(P_j\), (2) turn on enough FETs so that a conduction path from \(V_{dd}\) through \(P_j\) to the output exists if the fault is in the \(p\)-net, or a conduction path from the output through \(P_j\) to ground exists when the fault is in the \(n\)-net, and (3) turn off enough FETs so that no conduction path exists which does not include \(P_j\) [ES-Zig 81]. This type of 2-pattern test uses an all-1 pattern or an all-0 pattern as the \(T_1\) input, hence it will be called a simplified 2-pattern test.

The next two examples illustrate the derivation of simplified 2-pattern tests for FCGs. The first example is an AOI gate, and the second is an 8-input complex gate.

**Example 1** Find a 2-pattern test for FET P2 sop fault in the AOI gate shown in Fig. 2.

This circuit is taken from [Jain 83]. In that paper, this circuit is used to illustrate that some 2-pattern tests may be invalidated due to timing skews in multiple input changes (e.g. \(T_1=1\,0\,0\) and \(T_2=0\,0\,1\)). The authors proposed a 2-pattern test (i.e. \(T_1=0\,1\,1\) and \(T_2=0\,0\,1\)) in which only one input makes a transition.

In our approach, the \(T_1\) input is the all-1 pattern because the faulty FET P2 is in the \(p\)-net. The \(T_2\) input sets A to logic-0, B to logic-0 and C to logic-0. This 2-pattern test contains two input \((A\,\text{and}\,B)\) changes. While changing both \(A\) and \(B\) to 0, a conduction path between \(V_{dd}\) and \(Z\) is established to provoke the fault. It is apparent that the order of these two input transitions will not affect the detectability of the P2 sop fault.

![A test for P5 sop fault.](image)

**Example 2** Find a 2-pattern test for FET P5 sop fault in the circuit of Fig. 3.

[Reddy 83] presented this circuit to demonstrate that because of arbitrary delays no 2-pattern test exists for the P5 sop fault. However, if every input variable and its complement can be set to logic-1 at the same time, a valid 2-pattern test for this fault can be derived.

To detect the P5 sop fault, a simplified 2-pattern test as listed below is applied to the complex gate. This 2-pattern test contains four input transitions. Input B and D changes have no effect on the output node \(Z\). Input A and C changes provoke the fault.

![A test for P5 sop fault.](image)

Although this test involves multiple input transitions, its validity still holds in the presence of arbitrary delays. There are four inputs \((A, B, C, D)\) which are not changed in this 2-pattern test. These inputs block all the conduction paths between \(V_{dd}\) and \(Z\), except for the path consisting of FET P5 and P6. Therefore, node \(Z\) cannot be accidentally changed to logic-1 before the \(T_2\) input is applied to the circuit.

![A complex gate.](image)

The following theorem formally states the validity of simplified 2-pattern tests under arbitrary circuit delays.

**Theorem 1** If an FCG implementation of a Boolean function is irredundant, then for each of its FET sop faults, there exists a simplified 2-pattern test which cannot be invalidated by arbitrary circuit delays.

**Proof:** Assume the faulty transistor in an FCG is the PFET \(P_1\) controlled by the input variable \(x_1\). Since the fault is in the \(p\)-net, the \(T_1\) input of the simplified 2-pattern test is the all-1 pattern. The \(T_2\) input can be derived using the method of Boolean difference. First, find the transmission \(T\) of the \(p\)-net. Second, find the Boolean difference of \(T\) with respect to \(x_1\). Let \(d(T|x_1) = \{x_1,...x_n\}\). Since the FCG is irredundant, its transmission \(T\) will not be vacuous in \(x_1\). A \(T_2\) input can be found by assigning logic-0 to input variable \(x_1\) and assigning proper logic value to other input variables such that \(f(x_1,...x_n) \neq 1\). If input variable \(x_1\) controls another FET \(P_2\) in the \(p\)-net, a \(T_2\) input can still be derived by assigning value to other input variables so that a conduction path through \(P_2\) will not exist.) This test input will be called TP and it will establish one or more conduction paths from \(V_{dd}\) through \(P_1\) to the output node. Let these paths be collectively called path A.

Comparing pattern \(T_1\) to \(T_2\), there may be multiple input variables that change from logic-1 to logic-0 to turn on PFETs in the FCG. Among all the input variables making transitions, some control FETs in conduction paths other than path A. These are classified as category one input variables. The other input variables belong to category two.

Category one input variables are set to logic-0 because their complement inputs are set to logic-1 within pattern TP so as to block
other conduction paths in the p-net. Category one input changes cannot establish a conduction path between Vdd and the output node. Otherwise, TP is not a valid test pattern for the P1 sop fault.

Category two input variables will turn on every FET, except the faulty P1, in path-A. If a subset of these input variables will establish a different conduction path C in the p-net, then these input changes may invalidate the 2-pattern test. However, if path C exists in the p-net then it will make path A redundant. Therefore, no other conduction path can exist because the circuit is irredundant. As a result, even if the 2-pattern test consists of multiple input changes, the order of input transitions will not affect the validity of the test.

Similarly, the test for an NFET sop fault can be proved to be valid under arbitrary delays.

Q.E.D.

2.3 A testable combinational circuit

A combinational circuit can be constructed by interconnecting testable gates. However, a sop fault in this type of circuit is difficult to test because the embedded FCGs of this circuit cannot be properly initialized by using an all-1 or an all-0 pattern at the circuit inputs. However, if an inverting buffer is added to every FCG which fans out to other FCGs, we can easily initialize an embedded FCG by setting all the circuit inputs to logic-1 or logic-0. This observation leads to the concept of a testable gate and a testable combinational circuit.

[Definition 2] A Testable Gate (TG) consists of a fully complement gate connected to an inverting buffer (IB).

Figure 4 shows a block diagram of a TG. It is interesting to note that any sop fault in the IB can be detected by testing the TG’s input. This implies that applying a simplified 2-pattern test to detect a sop fault in the n-net will toggle the input of the IB so that sop fault in FET PB will be detected as well. Therefore, it is not necessary to develop separate test patterns for sop faults in the IB.

![Figure 4. A testable gate structure.](image)

For a TG in general, assume TP is the test input T2 for a sop fault in the p-net, and TN is the test input T2 for a sop fault in the n-net. A 2-pattern test set for detecting a sop fault in each part of a TG is listed in Table 1.

<table>
<thead>
<tr>
<th>FCG</th>
<th>Fault in p-net</th>
<th>Fault in n-net</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1</td>
<td>all-1 pattern</td>
<td>all-0 pattern</td>
</tr>
<tr>
<td>T2</td>
<td>TP</td>
<td>TN</td>
</tr>
<tr>
<td>IB</td>
<td>faulty P1</td>
<td>faulty N1</td>
</tr>
<tr>
<td>T1</td>
<td>all-0 pattern</td>
<td>all-1 pattern</td>
</tr>
<tr>
<td>T2</td>
<td>TN</td>
<td>TP</td>
</tr>
</tbody>
</table>

[Definition 3] A Testable Combinational Circuit (TC) is a multilevel combinational circuit which consists of two types of logic gates: testable gates and fully complementary gates. Testable gates are used in all but the last level of the circuit. Fully complementary gates are used in the last level of the circuit.

(Property 2) (TC initialization)
If all the inputs of a TC are set to logic-0, then all of its logic gate inputs are set to logic-0. Likewise, if all the inputs of a TC are set to logic-1, then all of its logic gate inputs are set to logic-1. (During the initialization of a TC, if both x and its complement x' are inputs to the TC, they are assigned the same logic value.)

Due to the above property, it is straightforward to generate the T1 inputs for any logic gate embedded in a TC. This TC design technique resembles the CMOS domino circuit technique [Krambeck 82]. While a domino circuit offers the speed advantage over a conventional static CMOS circuit, a TC provides the testing advantage over a conventional static CMOS circuit.

Since the IB of every TG adds overhead to a TC design, it is worthwhile to investigate the impact of such a design technique in terms of circuit area and speed. A 4-bit adder circuit (FA4) macrofunction described in [LSI 85] is used as an example.

First, we examined the area overhead by comparing the FET counts of each design. The testable design uses 8% more FETs than the original design. This number is much smaller than the overhead figure of converting a conventional gate into a TG (<50%). This is because the last level of a TC consists of conventional gates which do not introduce any area overhead. Also, the original design uses 5 inverters which are not required in the testable design.

To evaluate how the testable design can affect the circuit performance, the critical paths of two adder designs are compared. These paths are shown in Fig. 5. The original design has six conventional gates in its longest path. The same path in the testable design contains the equivalent of nine conventional gates. Using device parameters from the Stanford University’s Center for Integrated Systems 2um CMOS process, SPICE simulations show negligible differences in the critical path delay of the two designs. This is because the IBs increase the drive capability of each TG in the critical path.

![Figure 5. The critical path of a 4-bit adder.](image)

(a) LSI logic design (b) testable design.

3 APPLICATION TO SCAN-PATH ICs

Many VLSI circuits are designed with the scan-path technique. This technique requires that every memory element in a sequential circuit be connected to a register (i.e., the scan-paths). The scan-paths can be easily tested independent of the sequential circuit configuration. The remaining combinational circuitry is then tested by applying test patterns through the scan-paths. However, to detect a sop fault in the testable combinational circuit, it is not necessary to shift two patterns into the scan-path. The TG input pattern can be generated by the scan-path itself.

For example, in the LSSD scheme, the scan-path is made out of Shift Register Latches (SRL) [Eichelberger 77]. A CMOS SRL...
implementation which is capable of generating the T1 input is shown in Fig. 6. Two control inputs (P and G) are added to the SRL. During normal mode, the P input works as the power line, and the G input as the ground line. The latch outputs (Q and Q') take on complementary values. During test mode, if both P and G are set to logic-1, the SRL outputs are forced to logic-1 and the scan-path generates an all-1 pattern. Similarly, if both P and G are logic-0, the scan-path generates an all-0 pattern for the CUT.

A procedure to apply the simplified 2-pattern test through the scan-path is described below.

1. (shift) Apply the appropriate number of A, B clock pulses to the scan-path so that the T2 input pattern is shifted in and stored in the L1 latches.
2. (T1 Input) Activate and hold the B clock line at logic-1. Set P and G to logic-1 such that an all-1 pattern appears at the inputs to the CUT, or set P and G to logic-0 if an all-0 pattern is required.
3. (T2 Input) Set P to logic-1 and G to logic-0, then the T2 input pattern will appear at the CUT inputs. Activate the C clock line once to strobe the CUT response into the scan-path.

Figure 6. A CMOS Shift register latch design for testability.

4 SUMMARY AND CONCLUSIONS

Testing for CMOS sop faults in a scan-path IC is difficult because of the following problems: (1) Detection of a sop fault requires the generation of two test patterns to the circuit. (2) Arbitrary delays in the CUT can invalidate a 2-pattern test. (3) It is almost impossible to apply the 2-pattern tests through a scan-path.

This paper describes a design technique which can facilitate the testing of sop faults in a scan-path IC. The technique has the following features:

1. It is based on a testable combinational circuit design.
2. Every logic gate in the testable circuit can be tested with a simplified 2-pattern test set for its sop fault.
3. This test set cannot be invalidated by arbitrary circuit delays.
4. For scan-path ICs, the simplified 2-pattern test can be easily applied through the scan-path by specially designed shift register latches.

A 4-bit adder is designed according to the proposed technique. The testable design uses 8% more FETs than the original design and imposes no performance penalty.

ACKNOWLEDGEMENTS

The authors would like to thank Dr. Saeid Bozorgi-Nesbat, Mario Cortes, Greg Freeman, Steve Millman and Ken Wagner for their valuable comments and suggestions. This work was supported in part by the Defense Advanced Research Projects Agency (DARPA) under Grant No. MDA 903-79-C-0680.

REFERENCES


END

10-87

DTIC