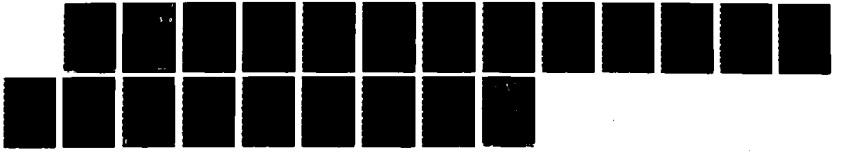
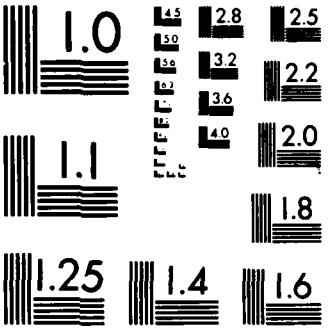
AD-A184 364 COLLECTION OF PREPRINTS AND REPRINTS(U) STANFORD UNIV 1/1
CA COMPUTER SYSTEMS LAB MAR 86 CSL-TN-86-287
MDA903-79-C-0680

UNCLASSIFIED F/G 9/1 NL





Center for Reliable Computing

Contract MDA903-79-C-0680

Collection of
PREPRINTS and
Reprints

Amer, H., and E.J. McCluskey, "Calculation of the Coverage Parameter for the Reliability Modeling of Fault-Tolerant Computer Systems" ISCAS'86

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

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 mechanism. 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 [Bridgman 84] [Ng 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 "c" 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 FTMP [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-0, for example, will force incorrect data on the output bus and the system will immediately fail.

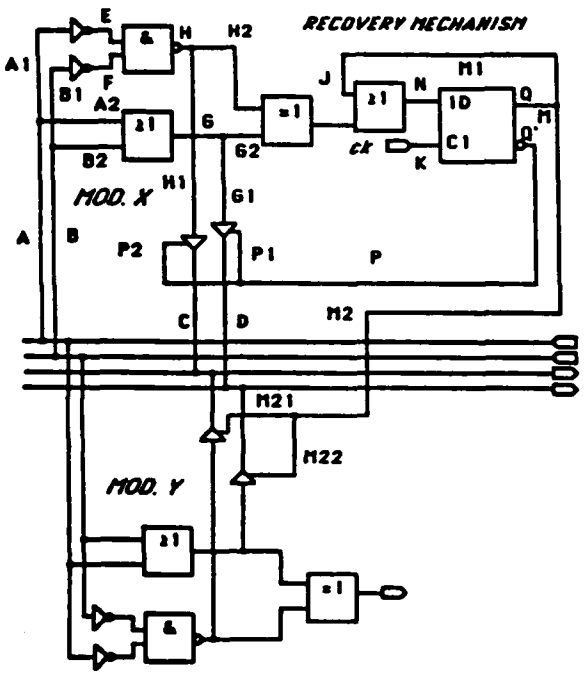


Fig. 1 Fault-tolerant System

When module Y is connected to the bus, detectable or undetectable faults (in mod. Y) will cause a system failure (incorrect data on the output bus). Therefore, they do not need to be distinguished from each other. Also, if a fatal fault occurs in module Y, the system immediately fails even if module X was connected to the bus. In summary, the faults in module Y can be grouped in two classes: Fatal and non-fatal (non-fatal = detectable and undetectable).

Two fault classes need to be defined for the recovery mechanism:

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-0, 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-0, 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, all consist of fatal faults. The total number of faults in group i is equal to C[i].

Table 1 Fault Classification

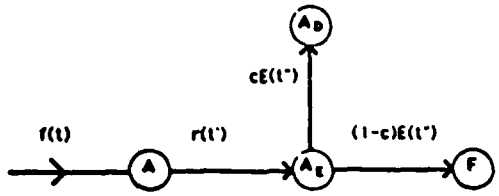
	Group	C[i]	s-a-0	s-a-1	class
Module X	1	4	A,B	A,B	undetectable
	2	8	C,D,G1,H1	C,D,G1,H1	fatal
	3	20	A1,A2,E,F	A1,A2,E,F	detectable
			B1,B2,G,H	B1,B2,G,H	
			G2,M2	G2,M2	
Detector	4	1		J	premature switching
	5	1	J		latent
Switch	6	3		M,M1,M	premature switching
	7	10	K,M,M,M1 M2,M21,M22	P,P1,P2	latent
	8	7	P,P1,P2	K,M2, M21,M22	fatal
Module Y	9	20	C[9] = C[1] + C[3]		non-fatal
	10	8	C[10] = C[2]		fatal

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 but no error exists yet. The fault produces an error (at a rate r(t')) and the system goes from state A to state A_E. If the error is not fatal, the system will go to state A_D (at a rate E(t") 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(t") is the average time for that error to be detected (or cause a system failure). State A_D 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.

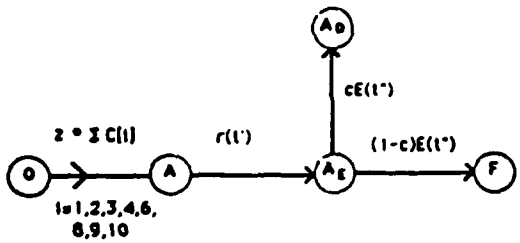
Only permanent faults will be considered here. The fault-handling model should be able to represent all the faults in the 1-out-of-2 system under study. The latent faults (J s-a-0 for

example) cannot be handled like the other faults. A latent fault will only affect the system after an error in module X. The parameter 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 84] 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_E : Fault caused an error
A_D : Module has been detected as faulty
F : System failure state
f(t) : Module 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



0 : Fault-free state

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).

$$c = \frac{\text{Sum } C[i] \quad i=3,4,6,9}{\text{Sum } C[i] \quad i=1,2,3,4,6,8,9,10}$$

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_E) and that from error to recovery (state A_D) or system failure (state F), are almost instantaneous.

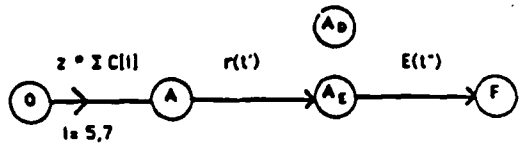


Fig. 4 Modeling the latent faults of the system in Fig 1

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]:

Reliability(t)=1-P(being in state F at time t)

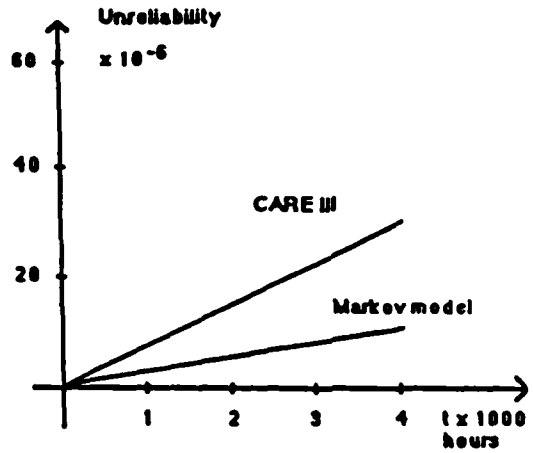


Fig. 5 Unreliability of the system

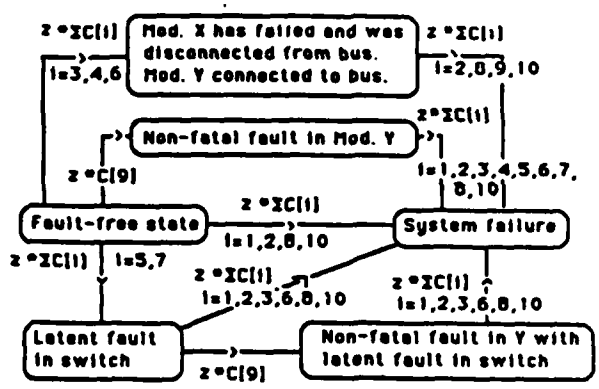


Fig. 6 Reliability model specific to system in Fig 1

The reliability of the whole system will be the product of the reliabilities calculated from the above two models (Figs. 3 and 4).

Unreliability(t):
$1-[1-P\{SF_{non-latent}\}]*[1-P\{SF_{latent}\}]$

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 2z be the failure rate 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 hours [Mil 82] along with a plot of an unreliability prediction for the same system calculated from a Markov 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 reliability model specific to 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 failures 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-82-K0105 and by the Egyptian Government. The logic simulation was performed on a 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

[Arnold 72] T.F. Arnold, "The Concept of Coverage and its Effect on the Reliability Model of a Repairable System," Proc. FTCS-2, Newton, Massachusetts, 1972, pp. 200-204.

[Bavuso 84] S.J. Bavuso, "A User's View of CARE III," Proceedings of the 1984 Annual Reliability and Maintainability Symposium, pp. 417-422, January 1984.

[Bridgman 84] M. Bridgman and W. Ness, "Automated Ultrareliability Models: A Review," Proceedings of the 1984 Annual Reliability and Maintainability Symposium, January 1984, pp. 396-402.

[Geist 83] R.M. Geist and K.S. Trivedi, "Ultra-High Reliability Prediction for Fault-Tolerant Computer Systems," IEEE Trans. Comput., December 1983.

[Hopkins 78] A. Hopkins, T. Basil Smith and J. Lala, "FTMP- A Highly Reliable Fault-Tolerant Multiprocessor for Aircraft," Proc. IEEE, Vol. 66, No. 10, October 1978, pp. 1221-1239.

[Losq 76] J. Losq, "A Highly Efficient Redundancy Scheme: Self-Purging Redundancy," IEEE Trans. Comput., June 1976.

[Mil 82] U.S. Department of Defense, "Military Handbook: Reliability Prediction of Electronic Equipment", MIL-HDBK-217D, January 1982.

[Ng 80] Y. Ng and A. Avizienis, "A Unified Reliability Model for Fault-Tolerant Computers," IEEE Trans. Comput., November 1980.

[Ogus 74] R. Ogus, "Fault-Tolerance of the Iterative Cell Switch Design for Hybrid Redundancy," IEEE Trans. Comput., July 1974.

[Trivedi 81]) K.S. Trivedi and R.M. Geist, "A Tutorial on the CARE III Approach to Reliability Modeling," NASA Contractor Report 3488, December 1981.

[Trivedi 83]) K.S. Trivedi and R.M. Geist, "Decomposition in Reliability Analysis of Fault-Tolerant Systems," IEEE Trans. Comput., December 1983.

PROPERTIES OF TRANSIENT ERRORS DUE TO POWER SUPPLY DISTURBANCES

Mario L. Cortes, Edward J. McCluskey,
Kenneth D. Wagner and David J. Lu

CENTER FOR RELIABLE COMPUTING
Computer Systems Laboratory
Stanford University, Stanford, CA 94305

ABSTRACT

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 rails 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 ([Allan 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 due to the increase of gate propagation delay during 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 basic cell is 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 XNORs 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 54S05) 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.

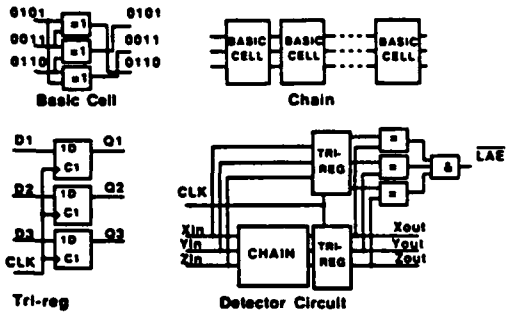


Fig. 1: The Detector Circuit And Its Building Blocks

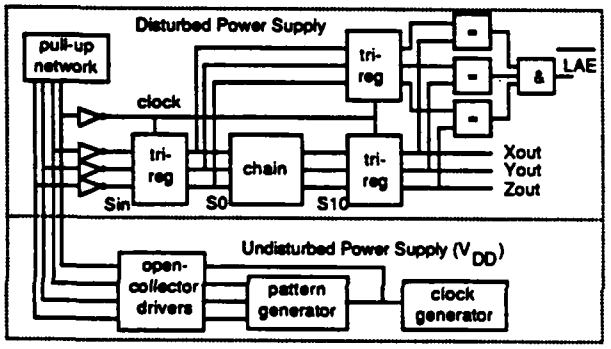


Fig. 2: The Detector Breadboard

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 occurences 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 ΔV_{DD}. Low values of ΔV_{DD} indicate poor tolerance to disturbances.

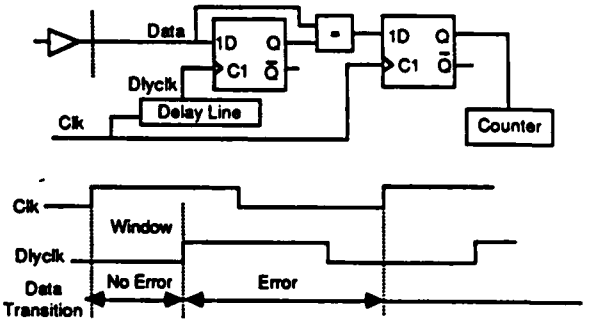


Fig. 3: The Metastability Detector

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 ΔV_{DD} is insensitive to pulse-width variations (1, 2, 4 and 7 microsec. were used). Furthermore, the same value of ΔV_{DD} was measured for DC disturbances. This is not surprising since propagation delays and transients are much shorter than 1 microsecond. ΔV_{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 (ΔV_{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 (ΔV_{DD} = 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].

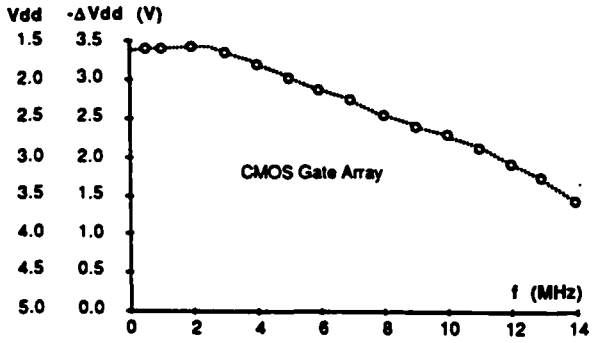


Fig. 4: Tolerance To Power Supply Disturbances vs Frequency for the Gate Array

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.

3.2 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 (ΔV_{DD}) was increased and the error output LAE was observed for errors. In the data points labeled **Input Error**, ΔV_{DD} is the disturbance magnitude to cause an incorrect vector at SO (Chain input in Fig. 2). The CMOS and LSTTL breadboards exhibit a similar behavior to the gate array (Fig. 4).

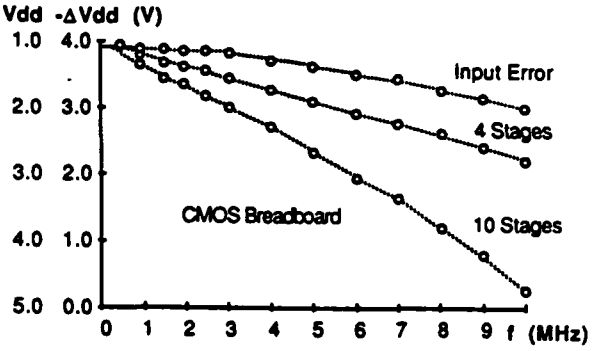


Fig. 5: Tolerance to Power Supply Disturbances vs frequency for the CMOS breadboard

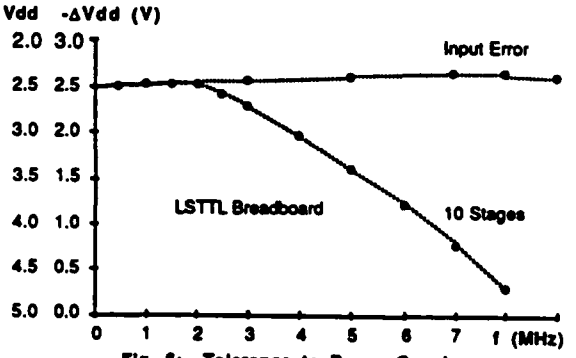


Fig. 6: Tolerance to Power Supply Disturbances vs frequency for the LSTTL breadboard

The error mechanism was observed by probing the breadboards. Increases in disturbance magnitude (ΔV_{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 set-up 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 (SO, 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}, Y_{out}, Z_{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 this voltage 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

window (nsec)	rate (late transitions / sec)
40	very high
60	320
80	2
100	0

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 Zazio, 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

[an 83] Alan Allan, "Noise Immunity of CMOS versus Popular Bipolar Logic Families", **Motorola Application Notes**, AN-708A, 1983.

[Chesney 83] T. Chesney and R. Funk, "Noise Immunity of COS/MOS B-Series Integrated Circuits", **RCA Appl. Notes**, ICAN-6587, 1983.

[Cortes 85] M. L. Cortes, "Experiments on Power Supply Disturbances in CMOS Gate Arrays", **PARA** Tech. Rep. MC85-1, Apr. 85.

[Freeman 85] Greg Freeman, "Metastability, a Perspective for Designers", EE385A, CSL, Stanford U., Stanford CA, 94305, June 3, 1985.

[Key 78] Lt. T. S. Key, "Diagnosing Power Quality-Related Computer Problems", **IEEE 1978 Ind. & Comm. Power Sys. Conf.**, pp. 48-59.

[Lu 84] D. J. Lu, " Experiments in Logic Power Supply Disturbances", **PARA** Tech. Rep. DL84-1, Dec. 84.

[McCluskey 81] E. J. McCluskey and J. F. Wakerly, "A Circuit for Detecting and Analysing Temporary Failures", **Spring Compcon**, Feb. 81, pp. 317-321.

[Wagner 85] K. Wagner and E. J. McCluskey, "Effect of Supply Voltage on Circuit Propagation Delay and Test Applications", **1985 Int. Conf. on Computer-Aided Design**, Santa Clara, CA, 1985.

[Wakerly 82] J. F. Wakerly, "Detector Chip Data Sheet", **PARA** Tech. Mem. 82-3, Jun. 82.

CONCURRENT BUILT-IN LOGIC BLOCK OBSERVER (CBILBO)

Laung-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 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].

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 Zi 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

---

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

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.

B1	B2	B3	Operation Mode
1	1	0	Normal
0	1	0	Reset
1	0	0	Signature analysis
1	0	1	Test generation
0	0	0	Scan

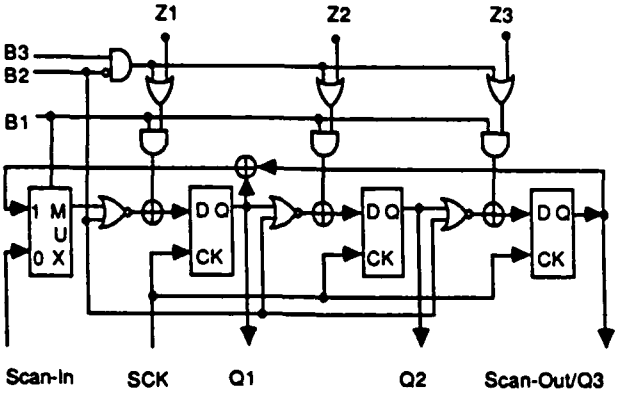


Figure 1. A 3-stage modified BILBO (MBILBO).

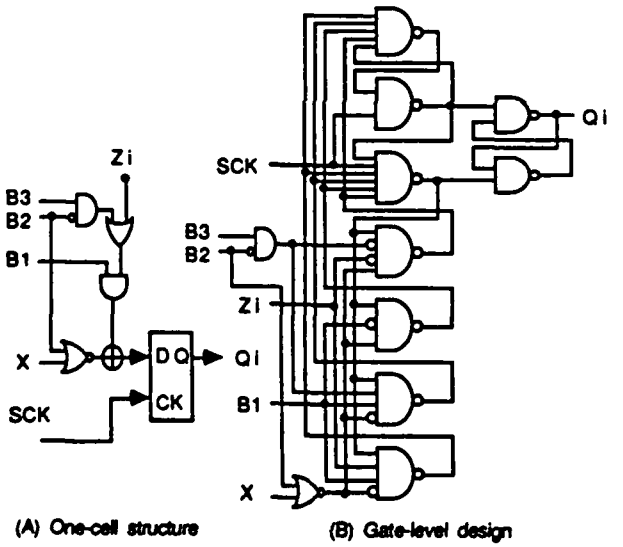


Figure 2. An MBILBO one-cell structure.

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 simulataneously 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.

B1	B2	Operation Mode
-	0	Normal
1	1	Scan
0	1	Test generation and Signature analysis

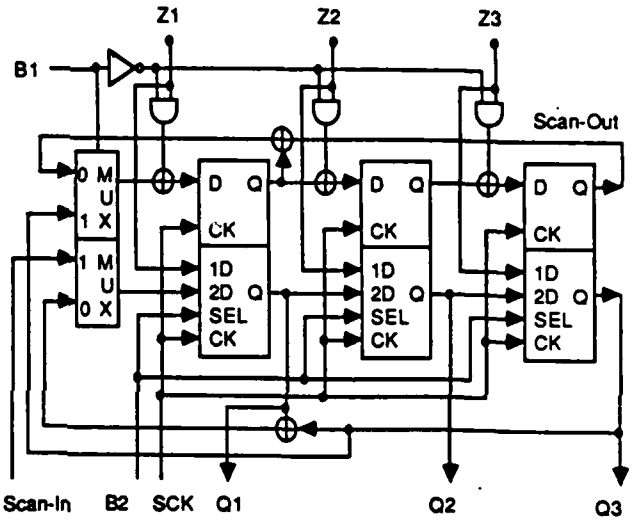
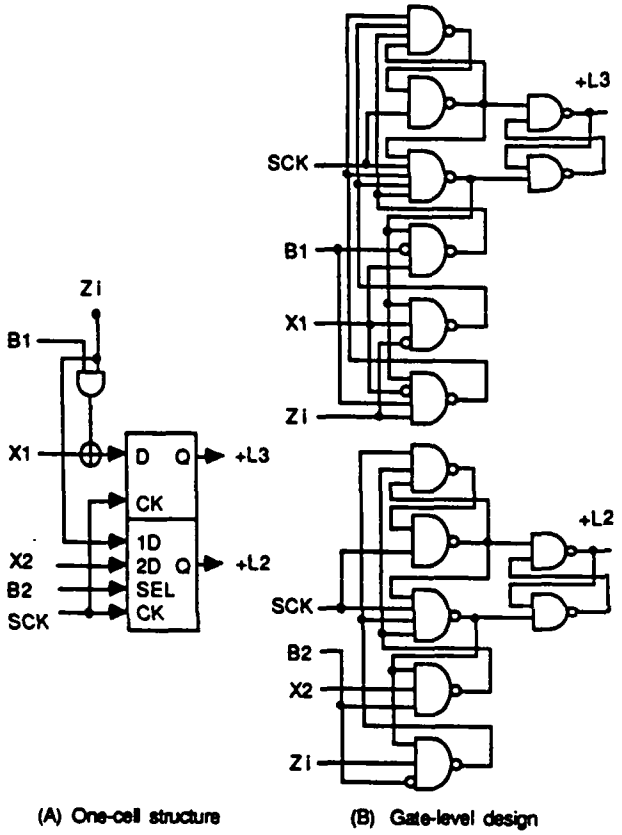


Figure 3. A 3-stage concurrent BILBO (CBILBO) using D-FFs.

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 Zi 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].



(A) One-cell structure (B) Gate-level design

Figure 4. A CBILBO one-cell structure using D-FFs.

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 [Konemann 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.

Clocks	B1	B2	Operation Mode
SCK/CK2	1	0	Normal
TCK/CK2	0	1	Scan
SCK/CK2/CK3	0	0	Reset
SCK/CK3/TCK/CK2	1	1	Test generation and Signature analysis

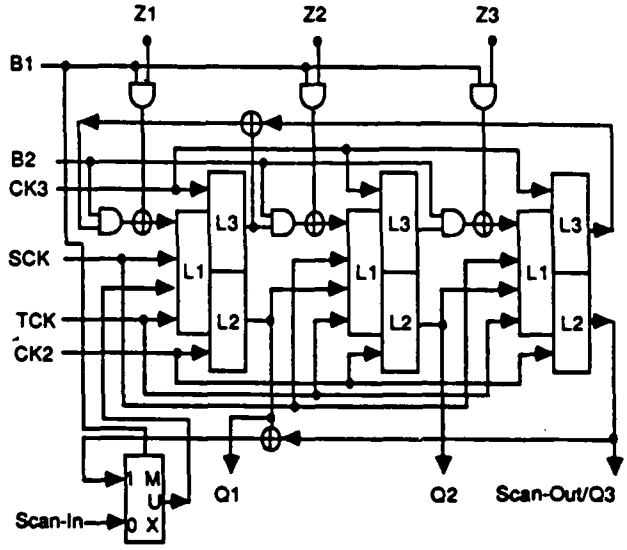
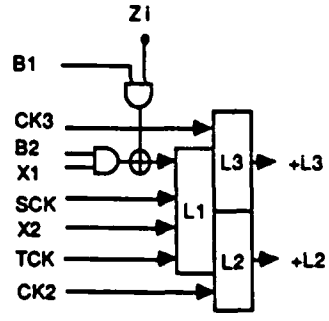
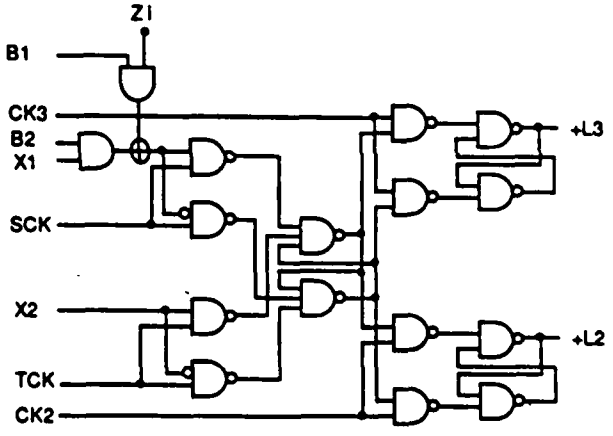


Figure 5. A 3-stage concurrent BILBO (CBILBO) using D-latches.



(A) One-cell structure



(B) Gate-level design

Figure 6. A CBILBO one-cell structure using D-latches.

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 BILBO, has some 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 is still self-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 BILBO approach. Finally, test time is much less than with the scan approach, and can be one-half the test time of the BILBO or MBILBO approach.

ACKNOWLEDGMENTS

The authors wish to thank Professor Samiha Mourad, Dr. Aamer Mahmood, Dr. Zuxi 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

[Barzilai 81] Barzilai, Z., J. Savir, G. Markowsky, and M.G. Smith, "The Weighted Syndrome Sums Approach to VLSI Testing," IEEE *Trans.* on Computers (IEEE-TC), pp. 996-1000, Dec. 1981.

[Benowitz 75] Benowitz, N., D.F. Calhoun, G.E. Alderson, J.E. Bauer, and C.T. Joeckel, "An Advanced Fault Isolation System for Digital Logic," IEEE-TC, pp. 489-497, May 1975.

[DasGupta 81] DasGupta, S., R.G. Walther, and T.W. Williams, "An Enhancement of LSSD Structure and Its Applications to non-LSSD Logic," Proc., 11th Int'l Symposium on Fault-Tolerant Computing (FTCS-11), pp. 32-34, June 1981.

[Eichelberger 77] Eichelberger, E.B., and T.W. Williams, "A Logic Design Structure for LSI Testability," Proc., 14th Design Automation Conf., pp. 462-468, Las Vegas, Nevada, June 1977.

[Frohwerk 77] Frohwerk, R.A., "Signature Analysis: A New Digital Field Service Method," Hewlett-Packard Journal, pp. 2-8, May 1977.

[Konemann 80] Konemann, B., J. Mucha, and G. Zwiehoff, "Built-in Test for Complex Digital Integrated Circuits," IEEE J. Solid-State Circuits, pp. 315-318, June 1980.

[McCluskey 81] McCluskey, E.J., and S. Bozorgui-Nesbat, "Design for Autonomous Test," IEEE Trans. on Circuits and Systems, pp. 1070-1079, Nov. 1981.

[McCluskey 84a] McCluskey, E.J., "Verification Testing -- A Pseudo-Exhaustive Test Technique," IEEE-TC, pp. 541-546, June 1984.

[McCluskey 84b] McCluskey, E.J., "A Survey of Design for Testability Scan Techniques," VLSI Design, pp. 38-61, Dec. 1984.

[McCluskey 85a] McCluskey, E.J., "Built-In Self-Test Techniques," IEEE Design & Test of Computers, pp. 21-28, April 1985.

[McCluskey 85b] McCluskey, E.J., "Built-In Self-Test Structures," IEEE Design & Test of Computers, pp. 29-36, April 1985.

[McCluskey 86] McCluskey, E.J., Logic Design Principles: With Emphasis on Testable Semicustom Circuits, Prentice-Hall, Englewood Cliffs, NJ, 1986.

[Wang 85] Wang, L.-T., and E.J. McCluskey, "Built-In Self Test for Random Logic," Proc., IEEE 1985 Int'l Symposium on Circuits and Systems (ISCAS-85), pp. 1305-1308, Japan, June 1985.

[Wang 86] Wang, L.-T., and E.J. McCluskey, "Condensed Linear Feedback Shift Register (LFSR) Testing -- A Pseudo-Exhaustive Test Technique," to appear in IEEE-TC, Special Issue on Fault-Tolerant Computing, June 1986.

[Williams 83] Williams, T.W., and K.P. Parker, "Design for Testability -- A Survey," Proc. IEEE, Vol. 71, No. 1, pp. 98-112, Jan. 1983.

[Williams 73] Williams, M.J.Y., and J.B. Angell, "Enhancing Testability of Large-Scale Integrated Circuits via Test Points and Additional Logic," IEEE-TC, pp. 47-60, Jan. 1973.

Table 1. CBILBO comparison with other designs.

Parameter vs. Type	D-Latch	SRL	BILBO	MBILBO	Stable SRL	D-FF CBILBO	D-Latch CBILBO
Pin Count	3	7	6	7	9	8	11
Clock/Test Pins	1	3	3	4	4	3	6
Hardware Overhead*	1	2-3	3-4	3-4	3-4	4-5	4-5
Extra Gate Delays	0	0	1-2	0	0	0	2
Test Data Storage*	-	N	1	1	N	2	2
Test Times*	-	>>2	2	2	>>2	1	1

*: 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*

Mario 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[1,2] 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)[3]. It consists of cascaded basic cells, synchronizing elements and checking circuitry. When the complete test set for the Exclusive ORs (XORs) (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 dc disturbances were used; pulse duration longer then device propagation delay. This is a reasonable simplification since short power supply disturbances (100ns or shorter) are very rare[4].

Figure 2 shows the dependency of disturbance magnitude (ΔV_{DI}) 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.

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 voltages occurs in CMOS and LSTTL logic and was discussed earlier[5].

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 Tri-reg (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 80ns 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 4. 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 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).

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

*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 under Contract No. N00014-85-K-0600. The logic simulation was performed on a Megalogician workstation made possible by Daisy Systems Co. (Mountain View, CA).

**Fabricated by Storage Technology Corp.

***74LS, 74HC.

[1] Allan, A., "Noise Immunity of CMOS Versus Popular Bipolar Logic Families", *Motorola Application Notes*, AN-708A; 1983.

[2] Chesney, T. and Funk, R., "Noise Immunity of COS/MOS B-Series Integrated Circuits", *RCA Application Notes*, ICAN-6587; 1983.

[3] McCluskey, E.J. and Wakerly, J.F., "A Circuit for Detecting and Analyzing Temporary Failures", *Spring Compcon, Digest of Papers*, p. 317-321; Feb., 1981.

[4] Key, Lt. T.S., "Diagnosing Power Quality-Related Computer Problems", *IEEE Industrial & Commercial Power Systems Conference*, p. 48-59; 1978.

[5] Wagner, K. and McCluskey, E.J., "Effect of Supply Voltage on Circuit Propagation Delay and Test Applications", *Int. Conf. on Computer-Aided Design*; 1985.

0193-6530/86/0000-0164$01.00 © 1986 IEEE

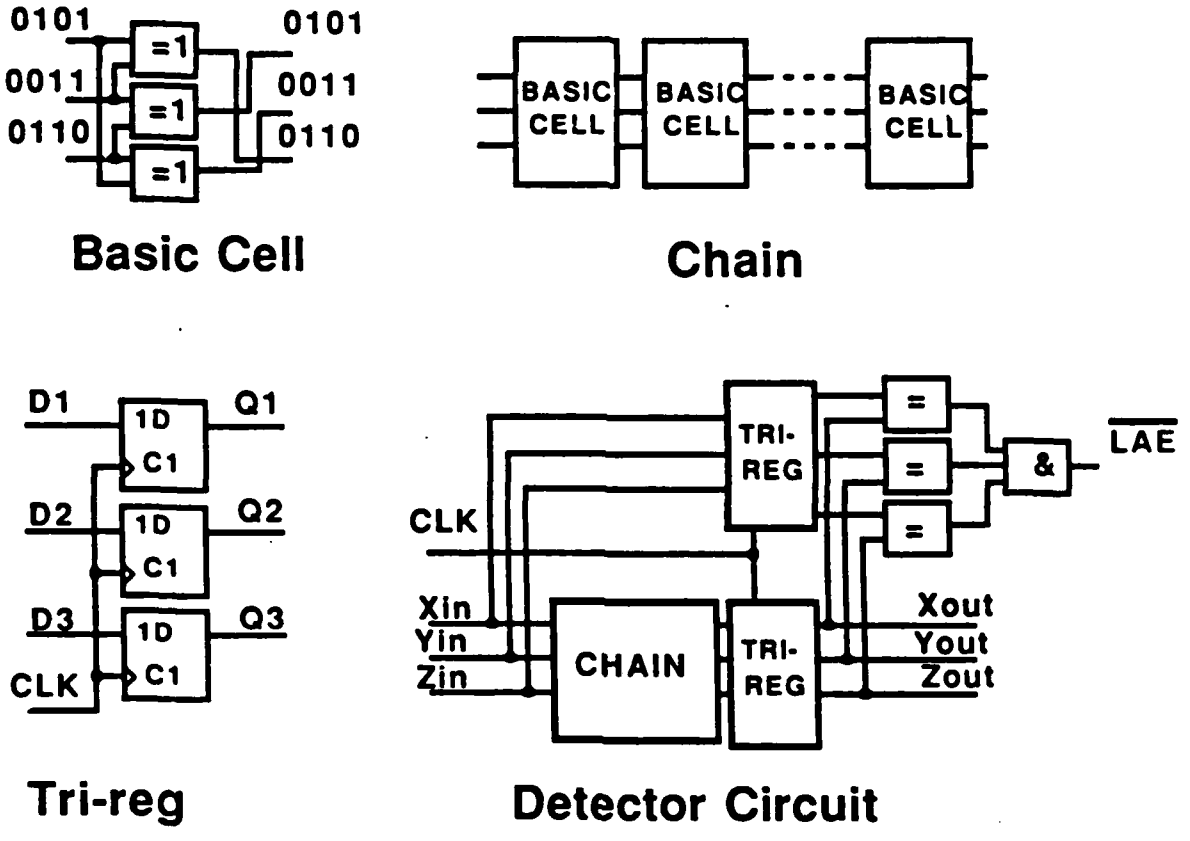


FIGURE 1—Detector circuit and building blocks.

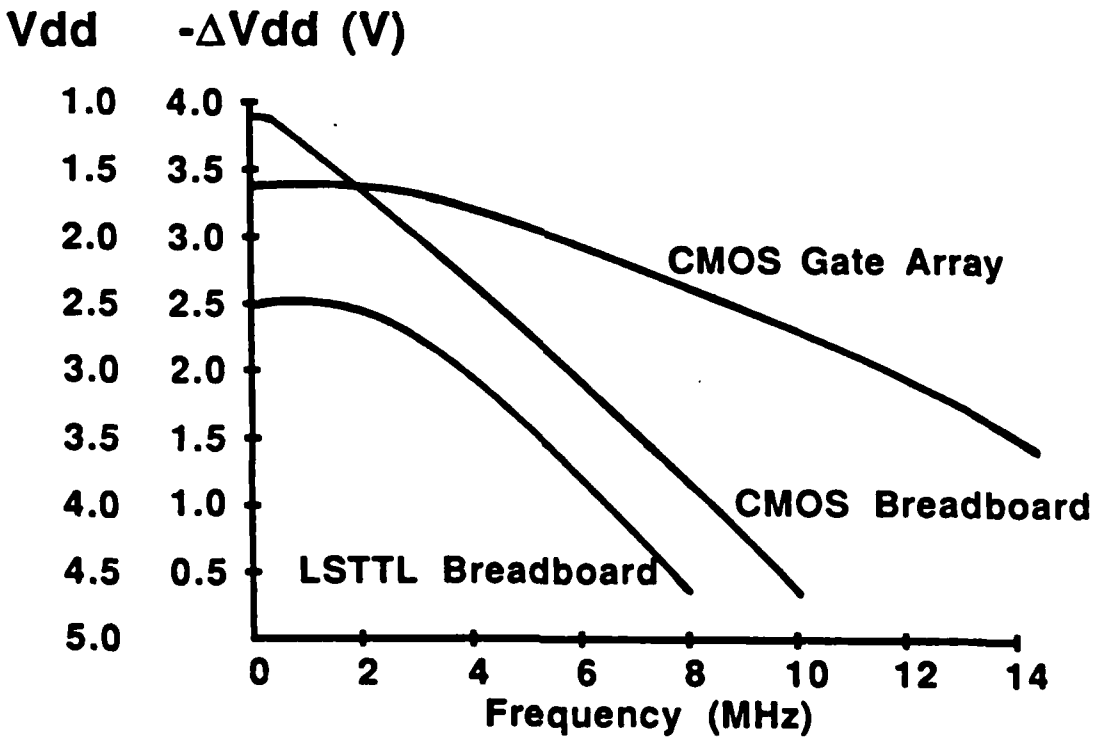


FIGURE 2—Tolerance to power supply disturbances vs. frequency.

DESIGN OF CMOS VLSI CIRCUITS FOR TESTABILITY

Dick L. Liu 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 it 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) 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 non-conducting irrespective of the applied voltage level at the FET gate terminal and is called an *FET stuck-open fault* (sop fault) [Wadsack 78].

The test for a sop fault in a CMOS logic gate consists of two test patterns: an initializing input pattern and a test input pattern. The *initializing input* (T_1) places the output of a faulty CMOS logic gate at logic-1 to detect an NFET sop fault or at logic-0 to detect an PFET sop fault . The *test input* (T_2) 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 sop 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 T_2 pattern will alter the state of the CUT established by the T_1 pattern.

Several researchers proposed design techniques as solutions to the problem of testing for sop faults [McCluskey 81; Reddy 83; Jha 85; Zasio 85]. However, none addressed the problem of testing sop 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 sop 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.

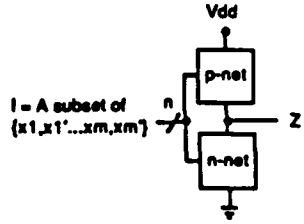


Figure 1. Block diagram of a CMOS gate.

[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.

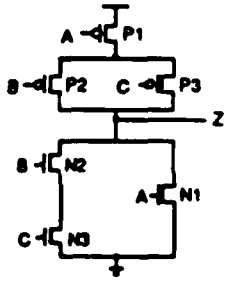


Figure 2. An AOI gate.

[Property 1] (FCG initialization)
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 sop 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_f), (2) turn on enough FETs so that a conduction path from Vdd through P_f to the output exists if the fault is in the p-net, or a conduction path from the output through P_f 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_f [El-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-1. This 2-pattern test contains two input (A 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 P2 sop fault.

	A	B	C	Z	Z*
T1	1	1	1	0	0
T2	0	0	1	1	0

Z*: faulty output

[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.

	A	A'	B	B'	C	C'	D	D'	Z	Z*
T1	1	1	1	1	1	1	1	1	0	0
T2	1	0	0	1	1	0	0	1	1	0

Z*: faulty output

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.

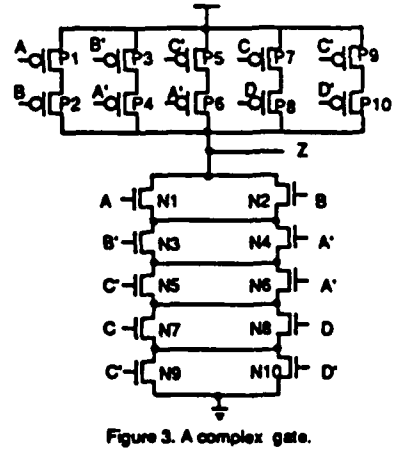


Figure 3. A complex gate.

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_f controlled by the input variable x_i. 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_i. Let $dT/dx_i = f(x_1,...,x_n)$. Since the FCG is irredundant, its transmission T will not be vacuous in x_i. A T_2 input can be found by assigning logic-0 to input variable x_i and assigning proper logic value to other input variables such that $f(x_1,...,x_n) = 1$. (If input variable x_i controls another FET P_x 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_x will not exist.) This test input will be called TP and it will establish one or more conduction paths from Vdd through P_f 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 P_f sop fault.

Category two input variables will turn on every FET, except the faulty P_f, 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 transitons 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 FCGs. However, a sop fault in this type of circuit is difficult to test because the embedded FCGs of this circuit can not 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 complementary 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 toggling the IB'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 P_{IB} will be detected as well. Therefore, it is not necessary to develop separate test patterns for sop faults in the IB.

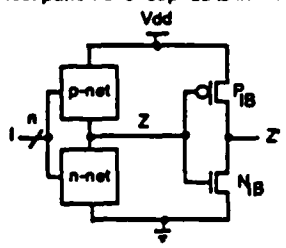


Figure 4. A testable gate structure.

For a TG in general, assume TP is the test input T_2 for a sop fault in the p-net, and TN is the test input T_2 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 1. A 2-pattern test set for a TG.

FCG	a fault in p-net	a fault in n-net
T1	all-1 pattern	all-0 pattern
T2	TP	TN
IB	faulty P_{IB}	faulty N_{IB}
T1	all-0 pattern	all-1 pattern
T2	TN	TP

[Definition 3] A Testable Combinational Circuit (TC) is a multi-level 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 T_1 input 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.

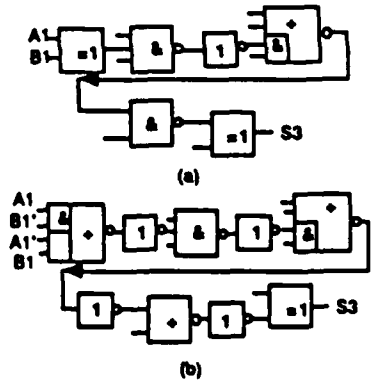


Figure 5. The critical path of a 4-bit adder.
(a) LSI logic design (b) testable design.

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 as one or more shift registers (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 T_1 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 T_1 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 T_2 input pattern is shifted in and stored in the L_1 latches.
2. (T_1 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. (T_2 input) Set P to logic-1 and G to logic-0, then the T_2 input pattern will appear at the CUT inputs. Activate the C clock line once to strobe the CUT response into the scan-path.

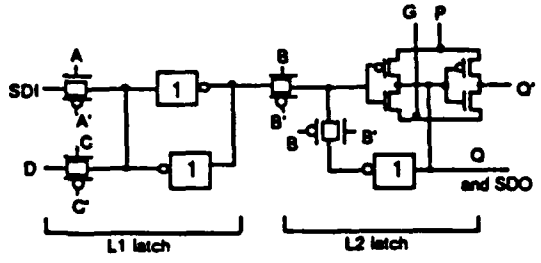


Figure 6. A CMOS Shift register latch design for testability.

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 application 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. Saied Borzorgui-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

[Eichelberger 77] Eichelberger, E.B. and T. Williams, "A logic design structure for LSI testing," Proc. 14th Design Automation Conf., New Orleans, LA, pp.462-468, 1977.

[El-Zig 81] El-Zig, Y.M. and R.J. Cloutier, "Functional-level Test Generation for Stuck-open faults in CMOS VLSI," Dig. papers, IEEE Int. Test Conf., Philadelphia, PA, pp.536-546, 1981.

[Jain 83] Jain, S. and V.D. Agrawal, "Test generation for MOS Circuits using D-Algorithm," Proc. 20th Design Automation Conf., Miami Beach, FL, pp.64-70, 1983.

[Jha 85] Jha, N.K. and J.A. Abraham, "Design of Testable CMOS logic Circuits under Arbitrary delays," IEEE Trans. Computer-aided Design, Vol. CAD-4, No.3, pp.264-269, July 1985.

[Krambeck 82] Krambeck, R.H., C.M.Lee and H.S.Law, "High-Speed Compact Circuits with CMOS," IEEE Journal of Solid-State Circuits, Vol. SC-17, No.3, pp.614-619, June 1982.

[LSI 85] LSI Logic Corp., CMOS Macrocell Manual, Milpitas, CA, July 1985.

[McCluskey 81] McCluskey, E.J. and S. Bozorgui-Nesbat, "Design for Autonomous Test," IEEE Trans. Computer, Vol. C-30, pp.866-875, Nov. 1981.

[McCluskey 84] McCluskey, E.J., "A Survey of Design for Testability Scan Techniques," VLSI Design, Vol. V, No.12, pp.38-61, Dec. 1984.

[Mukherjee 86] Mukherjee, A., Introduction to nMOS and CMOS VLSI Systems Design, Prentice-Hall, Englewood Cliffs, NJ, 1986.

[Reddy 83] Reddy, S.M., M.K. Reddy and J.G. Kuhl, "On testable design for CMOS logic circuits," Proc. 1983 Int. Test Conf., Philadelphia, PA, pp.435-445, 1983.

[Reddy 84] Reddy, S.M., M.K. Reddy and V.D. Agrawal, "Robust tests for Stuck-open Faults in CMOS combinational logic circuits," Dig. Papers, 14th Int. Conf. Fault-Tolerant Computer, Kissimmee, FL, pp.44-49, 1984.

[Wadsack 78] Wadsack, R.L., "Fault modelling and logic simulation of CMOS and MOS Integrated Circuits," The Bell System Technical Journal, Vol.57, No.5, pp.1449-1473, May-June 1978.

[Zasio 85] Zasio, J. J., "Non Stuck Fault Testing of CMOS VLSI," Proc. Compcon Spring 85, San Fransisco, CA, pp.388-391, 1985.

END

10-87

DTIC