CMOS Open-Fault Detection in the Presence of Glitches and Timing Skews

ROCHIT RAJSUMAN, MEMBER, IEEE, ANURA P. JAYASUMANA, MEMBER, IEEE, AND YASHWANT K. MALAIYA, MEMBER, IEEE

Abstract—**Two-pattern or multipattern test sequences may fail to detect CMOS stuck-open faults in the presence of glitches. The available methods to augment CMOS gates for testing stuck-open faults are found to be inadequate in the presence of glitches. A new testable CMOS design technique is presented. Some extra transistors are used in such a way that the CMOS gate is converted to a pseudo nMOS/pMOS gate during testing. With the proposed design technique, CMOS open faults can be detected regardless of timing skews/delays, glitches, or charge sharing among the internal nodes. The major advantage of the proposed testable design technique is that it allows the use of a single test vector to detect a stuck-open fault. This significantly reduces the complexity of test generation and the time consumed for testing. The design procedure is simple and all the classical algorithms and automatic test-pattern generating programs (ATPG's) can be used to generate tests for circuits designed according to this new technique. Even random testing techniques can be used efficiently to detect the open faults in these CMOS circuits.**

I. Introduction

In recent years, CMOS technology has been predominantly used in digital circuits. Several studies have shown that a significant fraction of probable faults in CMOS are not covered by the classical stuck-at fault model [1]–[5]. An example is the FET stuck-open fault. The detection of such a fault requires a sequence of two test vectors instead of a single test vector [1]–[4]. The first pattern is applied to initialize the output of the gate and the second pattern to detect the fault [5], [6]. If a stuck-open fault occurs in the n-part (p-part), then the first pattern sets the output to logic ONE (logic ZERO). The second pattern then attempts to provide a low-resistance path between the output and the ground (power supply) through the faulty transistor. In the presence of timing skews or unequal delays along the different paths of the circuit, spurious logic values may occur during transition from the first test pattern to the second test pattern [7], [8]. These spurious logic values may cause the failure of the test sequence. Robust two-pattern tests have been suggested which avoid this problem. In such sequences, the Hamming distance between the initialization pattern and the second test pattern is unity. Thus the possible intermediate state is avoided, and the sequence does not fail because of timing skews/delays. However, recently it has been shown that the glitches caused by the delays in prior logic can invalidate the two-pattern tests [11], [12].

The generation of robust test sequences is a complex process. The requirement of large CPU time makes the test generation costly. It is also possible that a two-pattern robust sequence may not exist for certain faults in a combinational circuit [7], [8]. Testable design schemes have been proposed [7]–[10] to overcome these problems. These schemes employ extra transistors in fully CMOS (FCMOS) gates and augment CMOS circuits to detect stuck-open faults. The test-generation complexity for these circuits is less compared to that of FCMOS circuits. However, these augmented circuits also require two-pattern or multipattern test sequences.

In the next section, we identify the conditions under which the schemes proposed in [7]–[10] fail to detect the stuck-open faults. A new design technique is proposed which allows the detection of any stuck-open fault with a single test pattern. The use of a single test pattern overcomes the problems associated with timing skews and propagation delays. It also simplifies the test-generation phase significantly and reduces the testing costs.

II. Glitches, Robust Tests, and Testable Designs

Below we illustrate how the two-pattern and multipattern test sequences may fail to detect stuck-open faults in CMOS gates, even when testable designs [7]–[10] are used. It has been reported that the robust test sequences are capable of detecting such stuck-open faults even if the inputs to a gate are affected by timing skews/delays [7]–[10]. However, the situation is more complex when such timing skews/delays cause a glitch at an input to the gate. For example, consider the circuit shown in Fig. 1 This circuit may be a part of a large combinational block. Consider the operation of this circuit under the input patterns shown in Fig. 2. The voltages at various lines are

Manuscript received April 6, 1988; revised March 13, 1989. This work was supported in part by the SDIO/IST program and is monitored by the ONR.

R. Rajsuman is with the Department of Computer Engineering, Case Western Reserve University, Cleveland, OH 44106.

A. P. Jayasumana is with the Department of Electrical Engineering, Colorado State University, Fort Collins, CO 80523.

Y. K. Malaiya is with the Department of Computer Science, Colorado State University, Fort Collins, CO 80523.

IEEE Log Number 8928787.

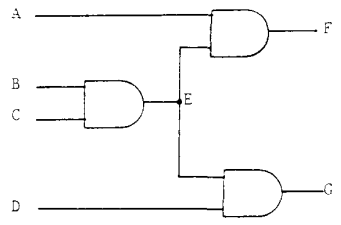


Fig. 1. Example circuit to show the formation of glitches due to variable delays.

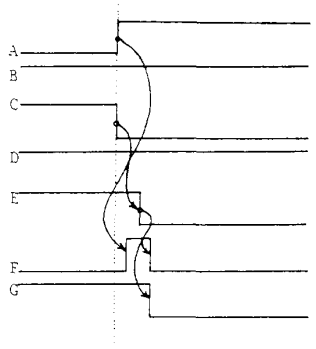


Fig. 2. Input and output node voltages of circuit shown in Fig. 1.

also shown in Fig. 2. As the inputs $[ABCD]$ change from [0111] to [1101], the output vector $[FG]$ changes from [01] to [00]. However, before the steady state [00] is achieved, the input F becomes momentarily 1, causing a glitch. The formation of a glitch and the width of the glitch formed depend on various delays involved. This example shows the formation of glitches due to delays in the prior logic. Glitches may also be caused by several other factors, such as switching of some transistors, external electromagnetic interference, and ionization radiation. The failure of a test sequence in such a case can be explained in a manner similar to the way in which charge distribution causes the test to fail [7], [8]. Testability of a circuit may be effected drastically by the presence of glitches. Glitches may also invalidate the robust test sequences used for the detection of stuck-open faults even in the augmented gates as suggested in [7]–[10].

A. Failure of Robust Test Sequences

Before the failure of robust test sequences in the presence of glitches is illustrated, we need to define the terminology used in this paper.

A glitch will be called *negative* (*positive*) if the steady-state voltage level at line/node of interest is logic ONE (logic ZERO) and goes to logic ZERO (logic ONE) momentarily because of the glitch. The presence of a glitch will be denoted by G.

A *zero vertex* 0v (*one vertex* 1v) is an input vector to a logic gate, which produces an output logic value ZERO (ONE) in the fault-free gate.

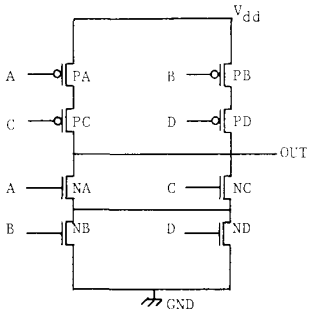


Fig. 3. CMOS example gate to show the failure of robust test sequences in the presence of glitches.

A *robust test sequence* is a sequence of two vectors which are different from each other in only one bit position.

To illustrate the failure of robust test sequences in the presence of glitches, consider the CMOS complex gate shown in Fig. 3. If I is a pair of input vectors, that causes the output to change from 1 to 0, this is denoted by $I \rightarrow (1,0)$. Consider the fault where transistor NA is stuck-open. To detect this fault, a test T is used, where $T = (t_1, t_2)$. Here t_1 is the initialization vector and t_2 is the actual test vector which sensitizes the fault. The initialization pattern t_1 can be chosen from the set

$$[ABCD] = [0101, 0001, 0100, 1010, 0010, 1000, 0000].$$

The second pattern t_2 of the sequence can be chosen from the set

$$[ABCD] = [1100, 1001, 1101].$$

To avoid the potential invalidation of the test sequence due to timing delays, a robust sequence which has unity Hamming distance between its patterns t_1 and t_2 can be chosen from the following set:

$$[ABCD, ABCD] = [(0100, 1100), (1000, 1100), (0001, 1001), (1000, 1001), (0101, 1101)].$$

The first pattern sets the output node to logic ONE and the second pattern creates a high-impedance state in the presence of the fault, i.e., $T \rightarrow (1, Z)$.

A positive glitch at node C caused by the prior logic during the second pattern may turn ON the transistor NC momentarily. This will create a low-resistance path from the output to ground. If the glitch is sufficiently wide, the output node may discharge. This effect will appear as

$$T \rightarrow (1, G, Z) \rightarrow (1,0).$$

Consequently the test sequence fails to detect the fault. This example shows that any possible robust test sequence may be potentially invalidated, and thus the fault could remain undetected. Similarly, one can show that the negative glitches may cause the failure of all possible robust test sequences for stuck-open faults in the p-part. The reason for the failure of a sequence is the presence of glitches that interfere with the high-impedance state. Hence the failure of a sequence is possible irrespective of the

method used for initialization. Such failures are possible not only with two-pattern sequences, but also with multiple-pattern sequences. This result can be formally stated as follows.

Theorem 1: In a FCMOS gate, except in a NOT gate, in at least one part (either the n-part or the p-part), all the FET stuck-open faults may remain undetected in the presence of circuit glitches, even when robust test sequences are used.

It should be noted that if a test sequence remains valid in the presence of glitches for a FET stuck-open fault, then any possible sequence for the dual transistor may be invalidated because of glitches. This is because the FCMOS gate has complementary n- and p-parts. A test sequence remains valid for a stuck-open fault in one part in the case where only one conduction path exists from output to ground/power supply. However, in such a case, there will be more than one conduction path in the dual part from the output node to power supply/ground. Glitches may spuriously switch an undesirable path in the dual part and may cause the failure of the test sequence. This may be stated in the form of a corollary.

Corollary: It is not possible to have a FCMOS gate, other than an inverter, in which two-pattern or multipattern tests for stuck-open faults are guaranteed to remain valid in the presence of glitches.

B. Failure of Test Patterns in Augmented Gates

References [7]–[10] propose several methods to augment CMOS gates to enhance the stuck-open fault testability. In all these techniques, some extra MOSFET's are used. Also, two-pattern or multipattern test sequences are used to detect stuck-open faults. These test sequences are made robust by controlling the additional FET's. This section will illustrate that the procedures given in [7]–[10] may fail to detect stuck-open faults in the presence of glitches.

First, consider an augmented CMOS gate as suggested by Reddy and Reddy [7]. The possible invalidation of two-pattern test sequences can be illustrated by considering the gate shown in Fig. 4. Two possible implementations are given in [7]. Both implementations utilize two extra MOSFET's controlled by a signal C_1. One nMOS (pMOS) is used in series with the n-part (p-part), while other pMOS (nMOS) is used in parallel with the p-part (n-part). Let p be a PFET in the original p-part and T_2 be a test for the corresponding stuck-open fault. Then according to [7], a proper initializing input T_1' and test input T_2' for the augmented gate is

$$T'_1 = T_2, C_1 = 1 \quad \text{and} \quad T_2' = T_2, C_1 = 0.$$

It is reported that only C_1 changes from T_1 to T_2. Hence, test sequence cannot be invalidated due to timing skews.

It can be shown that the test sequence may fail in presence of a negative glitch. For the circuit of Fig. 4, T_1' is given by $[C_1\ ABCD] = [1011X]$, and T_2' by $[C_1\ ABCD] = [0011X]$. Here T_1' sets the output to logic ZERO and T_2'

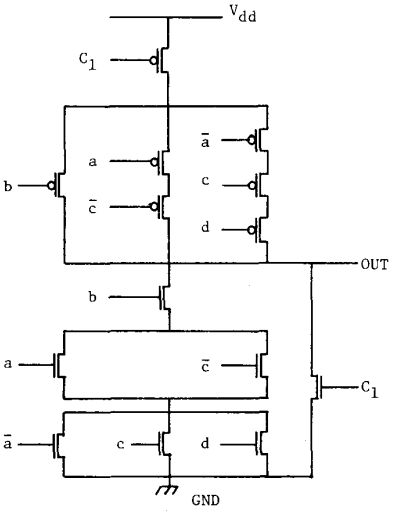


Fig. 4. CMOS augmented gate from [8].

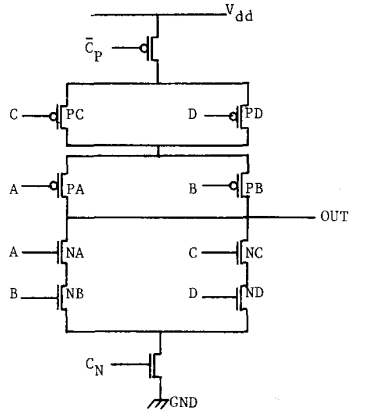


Fig. 5. CMOS augmented gate from [11].

creates a high-impedance state in the presence of transistor P_1 stuck-open. If a negative glitch occurs at input B while T_2' is being applied, the output node may charge to logic ONE. Hence, the sequence will fail to detect the fault. Similarly, it can be shown that the test patterns for any other PFET stuck-open fault also may fail due to negative glitches.

Other testable design schemes [9], [10] also use two extra MOSFET's to make the CMOS gates testable. However, an extra pMOS and an nMOS transistor are used in series with the p-part and the n-part, respectively. Also, the two transistors are controlled by two separate control signals C_n and C_p'. Liu and McClusky [10] have used an additional inverter at the output of every gate. However, the purpose of this inverter is just to propagate the fault effect. The designs of both [9] and [10] use two-pattern or multipattern sequences to detect the stuck-open faults in the presence of glitches.

Consider the gate shown in Fig. 5, with the fault *PA* stuck-open. A three-pattern sequence for this fault is given in [10]. The first two patterns of the sequence detect the

stuck-open fault and the last two patterns detect stuck-on fault in the dual transistor. As we are interested only in the stuck-open faults, only the first two patterns of the sequence will be considered. The second pattern of the sequence can be chosen from the set

$$[C_p'C_nABCD] = [010101, 010110, 010100].$$

It can be easily seen that a negative glitch at input B, while T_2 is being applied, can charge the output node. Hence the test sequence may fail to detect the PA stuck-open fault in the presence of glitches.

Recently it has been suggested that the invalidation of two-pattern or multipattern tests may be avoided by intentionally increasing the delays in some paths by using some extra transistors [13]. However, due to the use of a high-impedance state in two-pattern sequences, the test still could be invalidated by the presence of glitches.

Reddy *et al.* [7], [8], indicate the possibility of the failure of two-pattern test sequences due to charge sharing among the internal nodes. The probability of failure of a sequence due to charge sharing among the internal nodes is small. However, such failure is possible. None of the schemes [7]–[10] for augmenting CMOS gates, however, addresses this problem.

The discussion and the examples given in this section suggest that any possible test sequence may fail to detect CMOS stuck-open faults in the presence of glitches. Reference [11] shows how to avoid the formation of glitches in some specific cases. In general, however, it would not be feasible to identify every possible glitch and redesign the circuit to avoid all of them. As we have mentioned, the glitches can be caused by several factors. A test sequence may fail even if it is robust against timing delays. It is also clear that the available methods to modify the circuits do not make the design robust testable against glitches. An entirely new testing procedure has to be used to avoid this problem. In the next section we consider a new testable design of CMOS gates in which all single stuck-open faults can be detected irrespective of timing skews/delays, glitches, charge sharing among the internal nodes, or any other cause mentioned in the above discussion.

III. Design of Robust Testable CMOS Gates

In Section II it was shown that the two-pattern or multipattern tests could fail in the presence of circuit glitches. The main reason for such test invalidation is the high-impedance state that the last pattern of the test sequence creates at the output. The existing schemes examine the charge at the output during this high-impedance state to determine whether or not the fault is present.

During testing, glitches may violate the high-impedance state and charge/discharge the output node spuriously. Thus, the test sequence may fail to detect the fault. As long as the sequential behavior of the faulty FCMOS gate is used for stuck-open fault detection, glitches could invalidate the test patterns. This implies that the faulty circuit needs to be kept combinational during testing. This further implies that, to avoid such possible test invalidations, a high-impedance state should not be allowed during testing. In other words, a testing procedure which uses single test vectors [14] should be used for the testing of CMOS gates. Such procedures are used for testing nMOS circuits. Below we present a design technique, which allows the detection of a CMOS stuck-open fault by a single test vector.

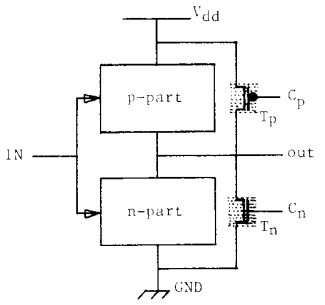


Fig. 6. Proposed design for single-pattern testable CMOS gate. Shade indicates that the ON resistances of T_p and T_n are considerably higher than the ON resistances of the n-part and p-part, respectively.

The proposed scheme allows the testing of the n-part and the p-part separately, thus facilitating the use of a single test vector to detect a stuck-open fault. It requires that the output node of a CMOS gate be connected to the power supply (ground) during the testing of the n-part (p-part). The resistances of these connections should be considerably higher than the ON resistances of the n-part or p-part. In fact, resistance ratios should be made similar to the ratio of load resistance (depletion transistors) to the ON resistance of the network in nMOS gates.

In FCMOS, the n-part and p-parts are complementary to each other. If a vector switches the p-part ON, it switches the n-part OFF and vice versa. Hence, we need to add a p-type transistor T_p parallel to the p-part, and an n-type transistor T_n parallel to the n-part, as shown in Fig. 6. These extra FET's also require two additional control signals, C_p and C_n, which keep them OFF during the normal operation but ON during testing. During the normal operation of the circuit, $C_p = 1$ and $C_n = 0$. During testing of the n-part $C_p = C_n = 0$, and during testing of the p-part $C_p = C_n = 1$. This augmentation is shown in Fig. 6. The dimensions of T_p and T_n should be chosen such that the ON resistances of T_p and T_n are considerably higher than the ON resistances of the n-part and the p-part, respectively. When $C_n = C_p = 0$, for example, the gate is essentially transformed to a pseudo nMOS gate [15]. Such pseudo nMOS gates do provide the correct logic operation, except for the obvious disadvantage of power dissipation. In the proposed design, the gate is transformed to a pseudo nMOS gate (pseudo pMOS gate) when testing for stuck-open faults in the n-part (p-part), as shown in Fig. 7. Therefore, the standard rules in designing pseudo nMOS (pseudo pMOS) type structures can be used to determine the sizes of T_p and T_n, respectively. A simplified set of rules to determine the sizes of T_p and T_n is given in the

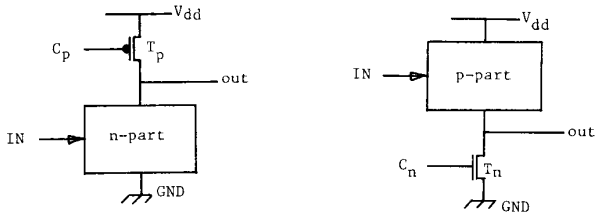


Fig. 7. Transformation of proposed SPT-CMOS gate into a pseudo nMOS or a pseudo pMOS gate during testing. Under 0v (for testing the n-part) the p-part is OFF and under 1v (for testing the p-part) the n-part is OFF.

Appendix. Further, any algorithm that is used to generate test patterns to detect stuck-open faults in nMOS circuits (i.e., [14]) can be used to generate tests for stuck-open faults in these augmented CMOS circuits.

In the proposed single-pattern testable (SPT-CMOS) gates, any stuck-open fault can be detected by a single test vector. This result can be formally stated as follows.

Theorem 2: By augmenting any FCMOS combinational gate as shown in Fig. 6, any single stuck-open fault in the functional part can be detected by a single test vector. These tests are not invalidated by timing skews/delays, glitches, or charge sharing among the internal nodes.

Proof: The proof of this theorem is constructive. Consider a stuck-open fault in the n-part. The test vector for this fault is $C_p = C_n = 0$, and a zero vertex (0v) covering the interested nMOS transistors. The 0v will turn the p-part OFF. Therefore, the augmented gate will appear as pull-down n-part and pull-up T_p (i.e., a pseudo nMOS gate). If a fault is present, it will cause a high-resistance path between the output and the ground. Hence, the output will appear to be logic ONE. If the fault is not present 0v will provide a low-resistance path from the output to the ground. As the ON resistance of T_p is considerably higher than the ON resistance of the n-part, the output will appear as logic ZERO. Therefore, a single test vector will detect the fault.

Similarly, it can be shown that $C_p = C_n = 1$ and a 1v, covering the interested pMOS transistor, can detect the pMOS stuck-open fault. It should be noticed that only a single test vector is required to detect the fault. The test vector brings the output node to a definite logic level and does not create a high-impedance state. If a glitch appears during testing, the output value may change momentarily. The steady-state value, however, will not be affected. The output voltage will recover after the glitch due to the path through T_n or T_p. Hence, the test cannot be invalidated because of timing skews/delays, glitches, or charge sharing among the internal nodes. In the traditional FCMOS designs, however, such a glitch could charge or discharge the output spuriously, and the output will not recover after the glitch because the second pattern creates a high-impedance state. QED.

It is also interesting to see how efficiently one can test an augmented CMOS gate for stuck-open faults. The result is given as follows.

Theorem 3: The functional transistors in an augmented CMOS gate, as shown in Fig. 6, can be tested for all single stuck-open faults by a sequence of maximal length $2n$, where n is the number of transistors in the unaugmented n-part or the p-part.

Proof: For testing the augmented gate for a stuck-open fault, 0v (for n-part) or a 1v (for p-part) is applied. This 0v or 1v is chosen such that they cover the FET of interest. Generally a number of FET's are covered by a single vector. For the worst case, when only one FET is covered by a vector, we will need at most n test vectors to test one part. Thus, to test the complete gate for all single stuck-open faults, at most $2n$ vectors are required. QED.

It should be noted that the additional transistors cannot be tested in this design. However, an open fault in one of these transistors is benign and does not affect the normal circuit operation. Furthermore, single-fault assumption implies that the extra transistors are fault free if a fault exists in the functional part and vice-versa.

Theorem 3 gives an upper bound for the length of the test sequence. In general, the length of the test sequence is much smaller. This is mainly due to the fact that a test vector examines the continuity of a path from the output to the ground/power supply. A number of transistors get tested by a single test vector. For example, the test set for a primitive gate (NAND, NOR, NOT) requires only $n+1$ vectors.

IV. Discussion

The major advantage of the CMOS testable design technique presented in Section III is that any stuck-open fault can be detected by a single test vector. This reduces the testing time drastically. It reduces the test application time by 50 percent because it requires a single test vector instead of a sequence of two vectors. Furthermore, it eliminates the complexity in test sequence generation. Complexity in generating two-pattern or multipattern sequences is a major cost factor in testing for CMOS stuck-open faults. The complexity and cost associated with generating robust test sequences is even higher.

As only a single pattern is required to test for a given fault, the tests for the augmented gates presented in Section III can be generated by simple procedures. All the classical algorithms, such as the D-algorithm and automatic test-pattern generating programs (ATPG's) for nMOS, can generate tests for such augmented gates. The scheme detects the stuck-open faults deterministically and ensures the detection irrespective of the problems identified in [7], [8], and [11]. Also, the proposed scheme can be used with random or pseudorandom testing procedures for detecting stuck-open faults. Such test techniques are very inefficient for the detection of stuck-open faults with the traditional FCMOS designs even with the prior testable design techniques. This is because the probability of testing a fault depends on two successive vectors.

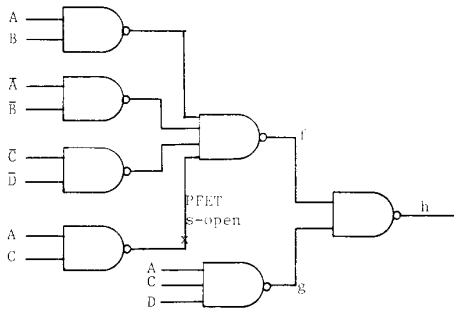


Fig. 8. FCMOS circuit from [8], for which no two-pattern robust test sequence exists for the stuck-open fault.

Some of the existing testable design schemes such as those proposed in [7] are not suitable for multilevel circuits. This is mainly due to the problems associated with the propagation of the effect of a fault to the circuit output. The proposed scheme is free from this drawback. As the scheme uses single test vectors and none of the parts is kept intentionally OFF, the effect of a fault is propagated without any problem. In fact, testing of CMOS gates by this procedure is comparable to the testing of nMOS gates. The proposed scheme can be used to implement two or multilevel circuits without any difficulty. To illustrate this, consider the circuit shown in Fig. 8 [7]. Detecting the stuck-open fault in a PFET driven by x requires a three-pattern sequence to avoid test invalidation by circuit delays. The sequence is given in [7] as $\{ABCD\} = [1001, 1011, 1010]$. However, using the proposed scheme, the same fault can be detected by a single vector $\{C_pC_n ABCD\} = [111010]$.

A disadvantage in the proposed scheme is the slightly higher power dissipation during testing. This, however, is not a significant disadvantage, as it occurs only during testing. Secondly, the scheme requires two additional FET's with two extra control signals. This consumes extra silicon area. However, if we compare the hardware overhead with that of other schemes [7]–[10], the proposed scheme requires either comparable or less hardware. Because of connecting two extra transistors to the output, the output capacitance of the gate will increase slightly. However, this change is also quite small and is comparable to that in other schemes [7]–[10].

The detection of faults in dynamic CMOS logic families has received attention recently [16]. The impact of the proposed design to enhance the fault coverage in dynamic logic gates is presently under investigation. The usefulness of the proposed design technique for the detection of stuck-short faults in CMOS circuits is also under investigation [17], [18].

APPENDIX

A simplified set of rules to calculate the size of T_p and T_n can be obtained by equating the current through the n-part and the p-part. Consider an inverter which has a pull-up transistor T_p and the pull-down part of the unaugmented gate. The current through the nMOS part is given by [15]

$$I_{dsn} = \frac{B_n}{2}(V_{inv} - V_{tn})^2 \tag{1}$$

where I_{ds} is drain-to-source current, V_t is the threshold voltage, V_{inv} is the gate threshold voltage, and B is the MOS transistor gain factor. Current through T_p is given by

$$I_{dsp} = B_p\left[(-V_{dd} - V_{tp})V_{dsp} - \frac{V_{dsp}^2}{2}\right] \tag{2}$$

where V_{dsp} is the drain-to-source voltage for the pMOS transistor. Equating the above two equations we get

$$V_{inv} = V_{tn} + 2\left(\frac{B_p}{B_n}\right)^{1/2}\left[(-V_{dd} - V_{tp})V_{dsp} - \left(\frac{v_{dsp}^2}{2}\right)\right]^{1/2}. \tag{3}$$

For equal noise margins $V_{inv} = 0.5V_{dd}$. Assume $V_{tn} = V_{tp} = 0.2V_{dd}$ and $V_{dd} = 5$ V. Hence $B_n/B_p = 1/6$. Similarly for pseudo pMOS, $B_p/B_n = 1/6$.

These simplified rules give the optimal design. In practice some deviation from the B_n/B_p ratio of $1/6$ is acceptable. This ratio should be chosen to obtain the minimum area without degrading the noise margins significantly. SPICE simulations indicate a possible range of $1/10$ to $1/2$.

Using the above guideline for $B_n/B_p = 1/6$, the size of T_p and T_n can be calculated for NAND and NOR gates. It should be noted that the NOT gate does not need any extra transistor; the NAND gate needs only T_n and the NOR gate needs only T_p. For 2-μm technology (pMOS size $L = 2$ μm, $W = 12$ μm, and nMOS size $L = 2$ μm, $W = 6$ μm), the sizes of T_p and T_n are given below.

Gate	T_p	T_n
inverter	NA	NA
two-input NAND	NA	$L = 4$ μm, $W = 2$ μm
three-input NAND	NA	$L = 6$ μm, $W = 2$ μm
two-input NOR	$L = 2$ μm, $W = 4$ μm	NA
three-input NOR	$L = 2$ μm, $W = 6$ μm	NA

REFERENCES

[1] J. Galiay, Y. Crouzet, and M. Vergniault, "Physical versus logical fault models in MOS LSI circuits and impact on their testability," *IEEE Trans. Comput.*, vol. C-24, pp. 527–531, June 1980.

[2] Y. K. Malaiya, A. P. Jayasumana, and R. Rajsuman, "A detailed examination of bridging faults," in *Proc. Int. Conf. Computer Des.*, 1986, pp. 78–81.

[3] K. W. Chiang and Z. G. Vransic, "On fault detection in CMOS logic network," in *Proc. 20th Des. Automation Conf.*, June 1983, pp. 50–56.

[4] R. Rajsuman, Y. K. Malaiya, and A. P. Jayasumana, "On accuracy of switch level modeling of bridging faults in complex gates," in *Proc. 24th Des. Automation Conf.*, June 1987, pp. 244–250.
[5] Y. M. El-Ziq, "Automatic test generation for stuck-open faults in CMOS VLSI," in *Proc. 18th Des. Automation Conf.*, June 1981, pp. 347–354.
[6] S. K. Jain and V. D. Agarwal, "Test generation for MOS circuits using D-algorithm," in *Proc. 20th Des. Automation Conf.*, 1983, pp. 64–70.
[7] S. M. Reddy and M. K. Reddy, "Testable design for CMOS logic circuits," *IEEE Trans. Comput.*, vol. C-35, pp. 742–754, Aug. 1986.
[8] S. M. Reddy, M. K. Reddy, and J. G. Kuhl, "A testable design for CMOS logic circuits," in *Proc. Int. Test Conf.*, 1983, pp. 435–445.
[9] B. Gupta, Y. K. Malaiya, Y. Min, and R. Rajsuman, "CMOS combinational circuit design for stuck-open/short fault testability," in *Proc. Int. Symp. Electron. Devices, Circuits Syst. (ISELDECS)*, Dec. 1987, pp. 789–791.
[10] D. L. Liu and E. J. McClusky, "Designing CMOS circuits for switch level testability," *IEEE Design Test*, vol. 4, no. 4, pp. 42–49, Aug. 1987.
[11] P. S. Moritz and L. M. Thorsen, "CMOS circuit testability," in *Proc. IEEE Custom Integrated Circuits Conf.*, 1985, pp. 311–314.
[12] R. Rajsuman, Y. K. Malaiya, and A. P. Jayasumana, "CMOS stuck-open fault testability," *IEEE J. Solid-State Circuits*, vol. 24, no. 1, pp. 193–194, Feb. 1989.
[13] S. D. Sherlekar and P. S. Subramaian, "Conditionally robust two pattern tests and CMOS design for testability," *IEEE Trans. Computer-Aided Des.*, vol. 7, pp. 325–332, Mar. 1988.
[14] R. Rajsuman, A. P. Jayasumana, and Y. K. Malaiya, "Testing of complex gates," *Electron Lett.*, vol. 23, no. 16, pp. 813–814, July 1987.
[15] N. H. E. Weste and K. Eshraghian, *Principles of CMOS VLSI Design*. Reading, MA: Addison-Wesley, 1985, ch.2.
[16] R. Rajsuman, A. P. Jayasumana, and Y. K. Malaiya, "Testability analysis of bridging faults in dynamic CMOS," in *Proc. Int. Symp. Electron. Devices, Circuits Syst. (ISELDECS)*, Dec. 1987, pp. 630–632.
[17] R. Rajsuman, A. P. Jayasumana, and Y. K. Malaiya, "CMOS stuck-open fault detection using single test patterns," to be presented at the 1989 IEEE/ACM Des. Automation Conf.
[18] R. Rajsuman, "THCMOS: Testable high-speed CMOS gates," Dept. Computer Engineering, Case Western Reserve Univ., Cleveland, OH, Tech. Rep., 1989.

Rochit Rajsuman (S'84–M'88) received the B. Tech. degree from K. N. Institute of Technology, India, in 1984, the M.S. degree from the University of Oklahoma, Norman, in 1985, and the Ph.D. degree from Colorado State University, Fort Collins, in 1988, all in electrical engineering.

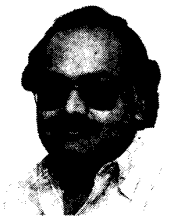
He joined the Department of Computer Engineering at Case Western Reserve University, Cleveland, OH, in August 1988 as an Assistant Professor. His research interests include VLSI design, testing, VLSI fault modeling, design for testability, fault-tolerant computing, and computer architecture.

Dr. Rajsuman was honored by the Colorado State Research Foundation for his research on CMOS testing while he was at Colorado State University. In 1987 he received the ACM SIGDA award for his presentation at the Design Automation Conference. As a graduate student at the University of Oklahoma, he received the Best Poster Presentation award at the annual university research day. For his academic excellence during his undergraduate studies, he continually received National Scholarships from the Government of India. He is a member of the IEEE Computer Society, the IEEE Circuits and System Society, the ACM special interest group on design automation, the Electrochemical Society, Tau Beta Pi, and Eta Kappa Nu.

Anura P. Jayasumana (S'83–M'85) was born in Colombo, Sri Lanka. He received the B.Sc. degree in electronic and telecommunication engineering, with First Class Honors, from the University of Sri Lanka, Moratuwa, in December 1978. He received the M.S. degree in electrical engineering in June 1982, and the Ph.D. degree in electrical engineering in December 1984, both from the Michigan State University, East Lansing.

From 1979 to 1980 he served as an Electrical Engineer at the National Engineering Research and Development Center of Sri Lanka, and as an Assistant Lecturer in the Department of Electronic and Telecommunication Engineering at the University of Sri Lanka, Moratuwa. Since January 1985 he has been at Colorado State University, Fort Collins, where he is an Associate Professor in the Department of Electrical Engineering. He is engaged in research on testing and design automation of VLSI. His research interests also include data communication networks.

Dr. Jayasumana is a member of Phi Kappa Phi, the IEEE Computer Society, the IEEE Industrial Electronics Society, the IEEE Communications Society, the IES Technical Committee on Factory Communications, and the CS Technical Committee on Microprogramming. He was the winner of the award for the best student in electrical engineering at the University of Sri Lanka, Moratuwa, in 1978, and the College of Engineering Outstanding Academic Achievement Award, Michigan State University, in 1982 and 1983.



Yashwant K. Malaiya (S'76–M'78) was born in Saugor, MP, India. He received the B.Sc. degree from the Government Degree College, Damoh, in 1969. He received the M.Sc. degree in physics from the University of Saugor in 1971 and the M.Sc. Tech. degree in electronics from the Birla Institute of Technology and Science, Pilani, in 1974. In 1978 he received the Ph.D. degree in electrical engineering from Utah State University, Logan.

During 1978–1982 he was an Assistant Professor in the Department of Computer Science at the State University of New York at Binghamton. Since 1982 he has been an Associate Professor in the Computer Science Department at Colorado State University, Fort Collins. He has published extensively in the areas of hardware reliability, software reliability, fault modeling, testing, and testable design. He has also been a consultant for industry. His current interests are testable design, register-level design, fault-modeling, software reliability, and microarchitecture.

Dr. Malaiya is a member of the IEEE Computer Society, the IEEE Reliability Society, and several IEEE-CS Technical Committees. He is also a member of the ACM and several special interest groups of the ACM. He is the chairman of the IEEE-CS Technical Committee on Microprogramming, a member of the IEEE-CS Technical Activities Board, and the Editor of *MicroArch*, a quarterly newsletter.