

Analysis of Detection Capability of Parallel Signature Analyzers

Yinghua Min, Yashwant K. Malaiya, and Boping Jin

Abstract—All signature analyzers are prone to aliasing errors. Here a rigorous mathematical analysis is presented to identify error conditions under which aliasing can occur for several common types of serial signature analyzers (SSA's) and parallel signature analyzers (PSA's). The PSA's are faster and require less hardware than the SSA's; however, for PSA's some double errors are special cause of concern. Such aliasing errors are analyzed and it is shown that PSA pairs can be identified for which these errors are disjoint. New reconfigurable PSA designs are presented which use a two-signature scheme to detect all double errors.

Index Terms—Built-in self-test, data compression, fault detection, linear feedback shift registers, signature analysis.

I. INTRODUCTION

Although the serial signature analyzers (SSA's) are often used to illustrate signature analysis, the parallel signature analyzers (PSA's) offer superior performance. A 16 bit PSA will compress 15 times more information than an SSA of the same size. The PSA's are widely used to compress the output vectors in BIST (built-in-self-test) [1]–[3]. 32-bit microprocessors like 68020 and 80386 use PSA's in the control part to improve testability. The main disadvantage of using a signature analyzer is the possible occurrence of an *aliasing error*, which will cause the output response from a faulty circuit to produce the same signature as a fault-free circuit.

A single bit error cannot cause an aliasing error in PSA's. However Hassan *et al.* [4] have pointed out that an error in a PSA input z_j at time t_i followed by an error in the input Z_{j+h} at time t_{i+h} has no effect on the signature. Errors of this type are examined in detail here. A combination of two single errors during the formation of a signature is termed a *double error* and if it causes aliasing, it is called an *aliasing double error*. The two components of a double error may occur during different clock periods. This type of aliasing error cannot be avoided even if more bits are added in the PSA [5]. Hassan [6] has proposed a two-signature testing scheme that reduces the frequency of aliasing from one in 2^r (for single signature scheme) to one in 2^{2r} , for an r -input PSA. The second signature is obtained by reversing the order in the input sequence. However, even in this scheme, aliasing double errors can occur. Also, it has been pointed out by Raghemi-Azar and Maxwell [7] that depending on the polynomial chosen, reversing the order of test vectors may or may not reduce aliasing.

Probabilistic treatments of the aliasing problem are given by Williams *et al.* [8], Gupta and Pradhan [9], and Ivanov and Agarwal [16]. They have shown that the probability of aliasing is a function of the probability of any bit being in error and the number of bits

Manuscript received October 4, 1988; revised September 15, 1989. This work was supported in part by an SDIO/IST contract monitored by ONR.

Y. Min was with the Department of Computer Science, Colorado State University, Fort Collins, CO 80523. He is now with the Center for Fault-Tolerant Computing, CAD Laboratory, Institute of Computing Technology, Academia Sinica, Beijing, China 100080.

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

B. Jin is with the School of MPC&CE, Macquarie University, NSW 2109, Australia.

IEEE Log Number 9100082.

in the signature analyzer. This raises an important question. Is there any structural approach that will significantly enhance the detection capability without arbitrarily increasing the size of the signature analyzer? If we assume that errors with higher multiplicity occur with decreasing probability, then double errors become the most important class of errors because single errors are always detected by PSA's. The double errors are of significance for SSA's but they have to be considered for PSA's. Here we take a deterministic approach to present a two-signature scheme which will provide complete coverage of aliasing double errors.

II. ALIASING ERRORS IN SIGNATURE ANALYZERS

The structure of a signature analyzer is generally specified by a characteristic polynomial $P(x)$ of degree r .

$$P(x) = p_0x^r + p_1x^{r-1} + \dots + p_r \quad (1)$$

where $p_0 = p_r = 1$. $P(x)$ is generally chosen to be primitive. Binary vectors can also be represented as polynomials with binary coefficients. The i th input sequence, $n_{1i}, n_{2i}, \dots, n_{ki}$, providing a k -bit vector to the SA, can be described by the polynomial

$$N_i(x) = n_{1i}x^{k-1} + n_{2i}x^{k-2} + \dots + n_{(k-1)i}x + n_{ki} \quad (2)$$

Two types of SSA's are possible. The SSA with feedback XOR gates in the middle, as shown in Fig. 1(a), is termed SSA-MF (serial signature analyzer with middle feedback). The following theorem is a well-known result [10].

Theorem 1: If an SSA-MF with characteristic polynomial $P(x)$ and with input sequence $N(x)$ generates the output sequence $Q(x)$, and if the final contents of the shift register are $R(x)$, then

$$N(x) = Q(x)P(x) + R(x) \quad (3)$$

Thus, $R(x)$ is the remainder of the polynomial division.

An SSA with feedback XOR gates on the side, as shown in Fig. 1(b), will be called SSA-SF (serial signature analyzer with side feedback). Smith [10] has shown that the content of the LFSR after the division is not the remainder. Shen and Su [11] and Maxwell [13] have briefly discussed its significance.

Theorem 2: If an SSA-SF with the characteristic polynomial $P(x)$ and with the input sequence $N(x)$ generate the output sequence $Q(x)$, and if $S(x)$ is the final contents of the shift register, then

$$x^r N(x) = (x^r Q(x) + S(x))P(x) + T(x) \quad (4)$$

Here $S(x)$ may be termed forward quotient, since no matter what input sequence follows $N(x)$, the next r -bit quotient is $S(x)$, while $T(x)$ is the remainder of $x^r * N(x)$ divided by $P(x)$.

Similarly, in the PSA's also the feedback XOR gates can be in the middle or on the side. In addition, the input XOR gates can also be either in the middle, or on the side. The four possible PSA configurations are shown in Fig. 2. These are, respectively, termed PSA-MF-MI (parallel signature analyzer with middle feedback and middle inputs), PSA-MF-SI (with middle feedback, side inputs), PSA-SF-MI (with side feedback, middle inputs), and PSA-SF-SI (with side feedback, side inputs). The PSA-MF-SI and PSA-SF-SI structures can take the advantage of a parity check tree which may be already present for other purposes.

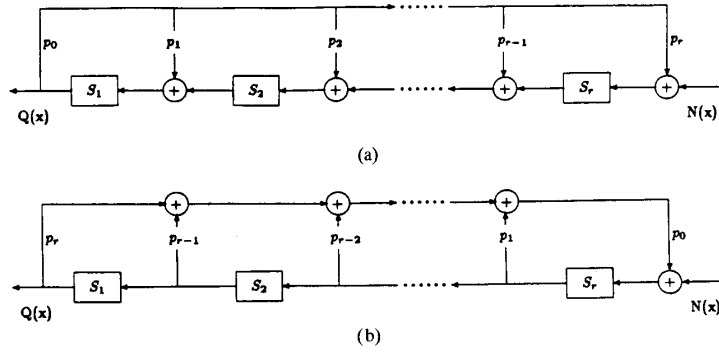


Fig. 1. The two types of serial signature analyzers. (a) SSA-MF. (b) SSA-SF.

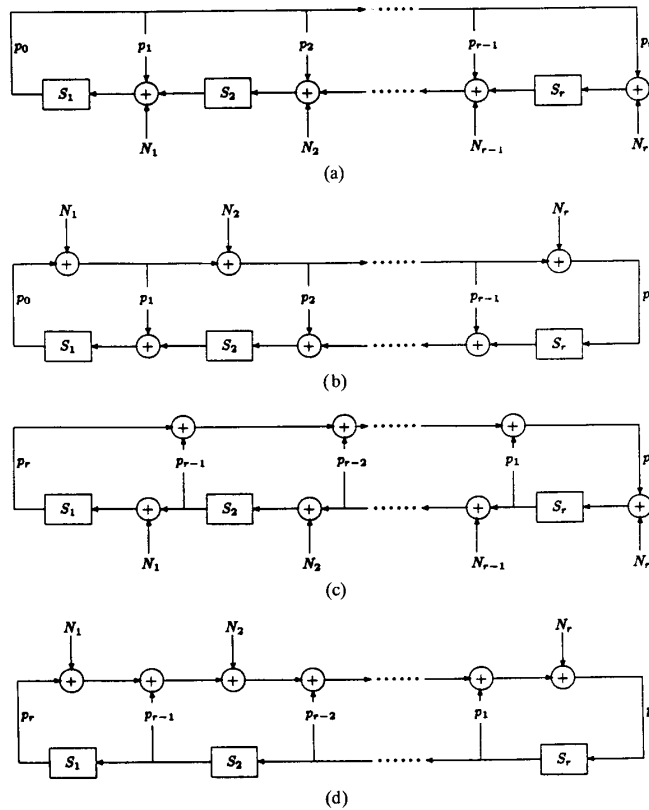


Fig. 2. The four types of parallel signature analyzers. (a) PSA-MF-MI. (b) PSA-MF-SI. (c) PSA-SF-MI. (d) PSA-SF-SI.

For an SSA, if two input sequences $N_1(x)$ and $N_2(x)$ produce the same signature, then the input sequence

$$E(x) = 0 \pmod{P(x)}.$$

$$E(x) = N_1(x) + N_2(x) \neq 0$$

will cause the LFSR to return to $00 \dots 0$ from the initial state $00 \dots 0$. Thus, $E(x)$ represents an aliasing error. The next theorem follows from the Theorems 1 and 2 [10].

Theorem 3: For an SSA-MF or an SSA-SF, $E(x)$ is an aliasing

Definition 1: Consider an r -bit PSA and an r -bit SSA with the same characteristic polynomial and initial state $00 \dots 0$. An input sequence $N(x)$ to an SSA is called an *equivalent input sequence* [12], [13] to input sequences $N_1(x), N_2(x), \dots, N_r(x)$ to a PSA, if the outputs of the highest bit, S_1 , as well as the signatures for the two signature analyzers are identical.

Theorem 4 [14]: If inputs $N_1(x), N_2(x), \dots, N_r(x)$ are inputs to a PSA-MF-MI, and the input $N(x)$ is the equivalent input sequence to the SSA with the same characteristic polynomial and initial state $00\dots 0$, then

$$N(x) = \sum_{j=1}^r x^{r-j} N_j(x). \quad (5)$$

The proof of Theorem 4 is given by Sridhar *et al.* [14]. The proofs of all the following theorems, except Theorems 8 and 12, have been omitted. They can be found in [12]. The proofs of Theorems 8 and 12 are given in the Appendix.

Theorem 5: Inputs $N_1(x), N_2(x), \dots, N_r(x)$ to a PSA-MF-SI are equivalent to the input $N(x)$, shown in (6), to the SSA with the same characteristic polynomial and the initial state $00\dots 0$.

$$N(x) = \sum_{j=1}^r \sum_{i=j}^r x^{r-i} p_i N_j(x). \quad (6)$$

Theorem 6: Inputs $N_1(x), N_2(x), \dots, N_r(x)$ to a PSA-SF-MI are equivalent to the input $N(x)$, shown in (7), to the SSA with the same characteristic polynomial and the initial state $00\dots 0$.

$$N(x) = \sum_{j=1}^r \left(\sum_{i=0}^{r-j} x^{r-j-i} p_i \right) N_j(x). \quad (7)$$

Theorem 7: Inputs $N_1(x), N_2(x), \dots, N_r(x)$ to a PSA-SF-SI are equivalent to the input $N(x)$ shown in (8), to the SSA with the same characteristic polynomial and the initial state $00\dots 0$.

$$N(x) = \sum_{j=1}^r N_j(x). \quad (8)$$

Definition 2: Let the following polynomials represent the input sequences.

$$\begin{aligned} N_1(x) &= a_{11}x^{k-1} + a_{21}x^{k-2} + \dots + a_{k1} \\ &\dots \dots \dots \\ N_r(x) &= a_{1r}x^{k-1} + a_{2r}x^{k-2} + \dots + a_{kr} \end{aligned}$$

The input matrix is defined as follows:

$$A_{k \times r} = \begin{bmatrix} a_{11} & \dots & a_{1r} \\ \vdots & & \vdots \\ a_{k1} & \dots & a_{kr} \end{bmatrix}$$

For two input matrices, $A_{k \times r}$ and $A'_{k \times r}$, the matrix

$$E_{k \times r} = [e_{ij}]_{k \times r} = A_{k \times r} + A'_{k \times r} \neq 0 \quad (9)$$

is called an *error matrix*. E_j represents its j th column. Any diagonal in $E_{k \times r}$ parallel to the diagonal $\{e_{k1}, e_{(k-1)2}, \dots, e_{(k-r)r}\}$ is called a subdiagonal. An error is an aliasing error, if nonzero $E_{k \times r}$ causes the initial state $00\dots 0$ to return after the input matrix has been applied. The double errors considered in the next section represent an error matrix with two nonzero entries.

Example 1: For a PSA-MF-MI with the characteristic polynomial $P(x) = x^5 + x^2 + 1$, an error matrix and the corresponding states

of the PSA are shown below:

$E_{4 \times 5}$	Initial State	00000
01000	State 1	01000
10110		2 00110
00101		3 01001
10010		4 00000

After the four steps, the PSA returns to the initial state, which means $A_{4 \times 5}$ and $A'_{4 \times 5}$ will produce the same signature if their error pattern is given by $E_{4 \times 5}$. Thus, $E_{4 \times 5}$ is an aliasing error.

Theorem 8: For PSA-MF-MI, $E_{k \times r}$ is an aliasing error iff

$$\sum_{j=1}^r x^{r-j} E_j(x) = 0 \pmod{P(x)} \quad (10)$$

The proof is given in the Appendix.

Corollary: If the parity of any subdiagonal of $E_{k \times r}$ is even, then $E_{k \times r}$ is an aliasing error.

It is clear that in this case the error matrix satisfies (10). $E_{4 \times 5}$ in Example 1 is an example of this type of error matrix. For r k -step inputs, there are $2^{kr} - 1$ possible errors. It is easy to see from (10) that there are $2^{kr-r} - 1$ aliasing errors. The corollary is a special case of Theorem 8, but includes a very special case considered in [10], that iff $e_{ij} = 1, e_{i-h, j+h} = 1, E_{k \times r}$ is an aliasing error. Theorem 8 is very general and it describes all possible aliasing errors for PSA-MF-MI. For other types of PSA's, the following three theorems present the necessary and sufficient conditions.

Theorem 9: For PSA-MF-SI, $E_{k \times r}$ is an aliasing error iff

$$\sum_{j=1}^r \sum_{i=j}^r x^{r-i} p_i E_j(x) = 0 \pmod{P(x)}. \quad (11)$$

Theorem 10: For PSA-SF-MI, $E_{k \times r}$ is an aliasing error iff

$$\sum_{j=1}^r \left(\sum_{i=0}^{r-j} x^{r-j-i} p_i \right) E_j(x) = 0 \pmod{P(x)}. \quad (12)$$

Theorem 11: For PSA-SF-SI, $E_{k \times r}$ is an aliasing error iff

$$\sum_{j=1}^r E_j(x) = 0 \pmod{P(x)}. \quad (13)$$

III. ALIASING DOUBLE ERRORS

This section presents an analysis of aliasing double error patterns for the four types of PSA's with primitive characteristic polynomial $P(x)$. In what follows, it is assumed that the only two nonzero elements in $E_{k \times r}$ are $e_{i_1 j_1}$ and $e_{i_2 j_2}$. Without loss of any generality, it can be assumed that $j_1 \leq j_2$.

Theorem 12: For PSA-MF-MI, $E_{k \times r}$ is an aliasing double error iff

$$i_1 - i_2 = j_2 - j_1 \pmod{(2^r - 1)}. \quad (14)$$

The proof is given in the Appendix. This theorem specifies that for two errors to be an aliasing double error they must occur on opposite vertices of a square in the input matrix. The sides of the square are equal in the mod $(2^r - 1)$ sense.

Using the lemma below a similar theorem can be obtained for PSA-MF-SI.

Lemma: Let $P(x)$ be a primitive polynomial of degree r , and let $F(x)$ and $R(x)$ be arbitrary nonzero unequal polynomials of degrees less than r . Then, there exists a unique integer m_0 , $0 < m_0 < 2^r - 1$, satisfying the following equation:

$$x^{m_0} F(x) + R(x) = 0 \pmod{P(x)}.$$

Theorem 13: For PSA-MF-SI, $E_{k \times r}$ is an aliasing double error iff it is one of the following cases:

Case 1: $j_1 = j_2$, and $i_1 - i_2 = 0 \pmod{2^r - 1}$.

Case 2: $j_1 < j_2$, and $p_{j_1}, p_{j_1+1}, \dots, p_{j_2-1} = 0$, and $i_1 - i_2 = 0 \pmod{2^r - 1}$.

Case 3: $j_1 < j_2$, and $p_l = 1, j_1 \leq l < j_2$, and there exists a unique m_0 which is not equal to $j_2 - j_1, 0 < m_0 < 2^r - 1$, and $i_1 - i_2 = m_0 \pmod{2^r - 1}$, where m_0 satisfies the following equation:

$$\sum_{j=j_1}^r p_j x^{r-j} + x^{m_0} \sum_{j=j_2}^r p_j x^{r-j} = 0 \pmod{P(x)}. \quad (15)$$

Theorem 14: For PSA-MF-MI, $E_{k \times r}$ is an aliasing double error iff it is one of the following cases:

Case 1: $j_1 = j_2$, and $i_1 - i_2 = 0 \pmod{2^r - 1}$.

Case 2: $j_1 < j_2$, and for any $j_1 \leq l < j_2, p_{r-l} = 0$, and $i_1 - i_2 = j_2 - j_1 \pmod{2^r - 1}$.

Case 3: $j_1 < j_2$, and there exists an l such that $p_{r-l} = 1, j_1 \leq l < j_2$, and then there exists a unique m_0 which is not equal to $j_2 - j_1, 0 < m_0 < 2^r - 1$, and $i_1 - i_2 = m_0 \pmod{2^r - 1}$, where m_0 satisfies the following equation:

$$\sum_{j=0}^{r-j_1} p_j x^{r-j_1-j} + x^{m_0} \sum_{j=0}^{r-j_2} p_j x^{r-j_2-j} = 0 \pmod{P(x)}. \quad (16)$$

Theorem 15: For PSA-SF-SI, $E_{k \times r}$ is an aliasing double error iff

$$i_1 - i_2 = 0 \pmod{2^r - 1}. \quad (17)$$

Example 2: Let us consider a PSA-MF-MI, a PSA-MF-SI, a PSA-SF-MI, and a PSA-SF-SI, all with the characteristic polynomial $P(x) = x^5 + x^2 + 1$. Suppose a double error in N_1 and N_4 , namely, $j_1 = 1$, and $j_2 = 4$. It can be an aliasing error for PSA-MF-MI, if $i_1 - i_2 = 3 \pmod{31}$. For instance, $\{e_{4,1}, e_{1,4}\}$ or $\{e_{5,1}, e_{2,4}\}$ are aliasing errors. For PSA-MF-SI, it corresponds to the case 3 of Theorem 13. Equation (15) becomes

$$x^2 + 1 + x^{m_0} = 0 \pmod{P(x)}.$$

Obviously, $m_0 = 5$. Therefore, the PSA-MF-SI has aliasing errors whenever $i_1 - i_2 = 5 \pmod{31}$, such as $\{e_{6,1}, e_{1,4}\}, \{e_{7,1}, e_{2,4}\}$. For PSA-SF-MI, it is in the case of Theorem 14, (16) becomes

$$x^3 + 1 + x^{m_0} = 0 \pmod{P(x)}.$$

The solution is $m_0 = 29$. Therefore, the error is an aliasing error whenever $i_1 - i_2 = 29 \pmod{31}$, such as $\{e_{30,1}, e_{1,4}\}, \{e_{31,1}, e_{2,4}\}$. For PSA-SF-SI, the error can be an aliasing error as long as $i_1 - i_2 \pmod{31}$, such as $\{e_{1,1}, e_{1,4}\}, \{e_{2,1}, e_{2,4}\}$, and so on.

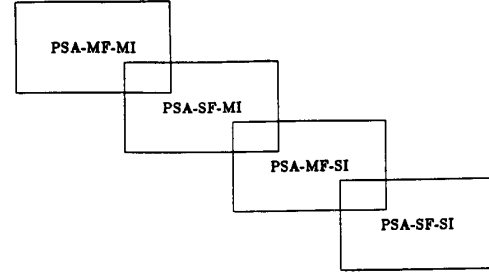


Fig. 3. The overlap among the four aliasing error pattern sets.

It is interesting to examine the aliasing double error pattern sets of the four types of PSA's. An aliasing error for one type of PSA may not be an aliasing error for another. Using Theorems 12–15, the following propositions follow.

1) The aliasing double error pattern sets of PSA-MF-MI and PSA-MF-SI with the same primitive characteristic polynomial are disjoint when $k \leq 2^r - 1$.

This follows from the fact that any double error satisfying (14) cannot satisfy the conditions for Theorem 13.

2) The aliasing double error pattern sets of PSA-MF-MI and PSA-SF-SI are disjoint.

3) The aliasing double error pattern sets of PSA-SF-MI and PSA-SF-SI are disjoint.

4) The aliasing double error pattern sets of PSA-MF-MI and PSA-SF-MI intersect.

This is because there are some double errors, which cannot be detected by both PSA-MF-MI and PSA-SF-MI. $\{e_{1,2}, e_{2,1}\}$ in Example 2 is an example.

5) The aliasing double error pattern sets of PSA-MF-SI and PSA-SF-SI intersect. $\{e_{1,1}, e_{1,2}\}$ in Example 2 is an example.

6) The aliasing double error pattern sets of PSA-SF-MI and PSA-MF-SI intersect.

The overlapping among the four aliasing error pattern sets is depicted in Fig. 3. From the above observations we can see that if two types of PSA's are properly selected in a system, then any double error that cannot be detected by one PSA, can be detected by the other.

IV. A PSA DESIGN CAPABLE OF DETECTING ALL DOUBLE ERRORS

It is well-known that using two PSA's, instead of one, decreases the probability of aliasing errors from 2^{-r} to 2^{-2r} [6]. If the two are of the same type or have nonempty intersection of their error pattern sets, aliasing double errors will still exist, regardless of the number of bits used. However, as the discussion in the previous section suggests, if the two PSA's are with empty intersection of their aliasing error pattern sets, there will be no aliasing double errors.

Doubling the number of bits can be feasible in some cases. However, when only limited hardware overhead is allowed, a reconfigurable scheme presented here can be used [12]. The same r -bit PSA can be of one type in one period, and of another type in next period. In this case, twice the test time is required to reduce the hardware overhead. The two signatures can be obtained at different times. This will reduce the continuous duration required for the self-test, which may be desired for real time applications etc.

Consider the expanded XOR gate shown in Fig. 4. It has an extra control line C . When C is 1, it is just an XOR gate. When C is 0, the output is equal to one of the inputs. By using this expanded XOR gate, the PSA shown in Fig. 5 can be configured either as a PSA-MF-MI or as a PSA-MI-SI. Two separate phases are used. In phase 1, test response is compressed by the PSA-MF-SI to produce the signature S1. In phase 2, the same test response is used by the PSA-MF-MI, and signature S2 is obtained. The combination of signatures S1 and S2 will definitely detect all double errors.

In other design environments, an alternative design of Fig. 6 can be used. The PSA-SF-SI in the upper part can be used as a normal register when the self-test is not being done. The parity-tree may also be present in many systems for other purposes. Then the PSA-SF-SI would be the only significant overhead. The PSA's may be used in parallel or at separate times. Compared to an LFSR of double length, this design has lower hardware overhead. It is capable of detecting all double errors, which cannot be obtained by using an LFSR of double length.

Another technique for avoiding aliasing double errors has recently been presented by Iwasaki *et al.* [15] which however uses Reed-Solomon codes. This scheme also requires two signatures.

V. CONCLUSIONS

A mathematical analysis of aliasing errors in the four types of PSA's is presented. Necessary and sufficient conditions are presented for each of the four PSA's, under which an error pattern causes aliasing. Double errors are examined which are of special significance for PSA's. The disjointness of aliasing double errors for the PSA's is discussed. This observation is used to present a reconfigurable design capable of detecting all possible double errors.

APPENDIX

Proofs of two key theorems in the paper are included here. For the other proofs please see [12].

Theorem 8: For PSA-MF-MI, $E_{k \times r}$ is an aliasing error iff

$$\sum_{j=1}^r x^{r-j} E_j(x) = 0 \pmod{P(x)}. \quad (10)$$

Proof: If (10) is satisfied, then

$$\sum_{j=1}^r x^{r-j} E_j(x) = Q_1(x)P(x).$$

By (5) the input matrix $A_{k \times r}$ produces the signature $R(x)$ satisfying

$$\sum_{j=1}^r x^{r-j} N_j(x) = Q_2(x)P(x) + R(x)$$

The input matrix $A'_{k \times r} = A_{k \times r} + E_{k \times r}$ produces the signature $R'(x)$ satisfying

$$\sum_{j=1}^r x^{r-j} (N_j(x) + E_j(x)) = Q'(x)P(x) + R'(x).$$

On the other hand,

$$\begin{aligned} \sum_{j=1}^r x^{r-j} (N_j(x) + E_j(x)) &= \sum_{j=1}^r x^{r-j} N_j(x) + \sum_{j=1}^r x^{r-j} E_j(x) \\ &= Q_2(x)P(x) + R(x) + Q_1(x)P(x) \\ &= (Q_1(x) + Q_2(x))P(x) + R(x). \end{aligned}$$

Thus, $R'(x) = R(x)$. Conversely, suppose the signatures of the two input matrices are the same, then

$$\sum_{j=1}^r x^{r-j} N_j(x) = Q(x)P(x) + R(x)$$

$$\sum_{j=1}^r x^{r-j} (N_j(x) + E_j(x)) = Q'(x)P(x) + R'(x)$$

and $R(x) = R'(x)$. Adding these two equations results in (10). Q.E.D.

Theorem 12: For PSA-MF-MI, $E_{k \times r}$ is an aliasing double error iff

$$i_1 - i_2 = j_2 - j_1 \pmod{2^r - 1}. \quad (14)$$

Proof: In the case of a double error, the sufficient and necessary condition (10) becomes

$$x^{k-i_1+r-j_1} + x^{k-i_2+r-j_2} = 0 \pmod{P(x)}.$$

That is,

$$x^{k-i_1+r-j_1} (1 + x^{i_1-i_2+j_1-j_2}) = 0 \pmod{P(x)}$$

$$1 + x^{i_1-i_2+j_1-j_2} = 0 \pmod{P(x)}.$$

Since $P(x)$ is primitive, therefore,

$$i_1 - i_2 + j_1 - j_2 = 0 \pmod{2^r - 1}$$

That is,

$$i_1 - i_2 = j_2 - j_1 \pmod{2^r - 1}$$

Q.E.D.

ACKNOWLEDGMENT

The authors would like to thank R. Rajsuman and B. Gupta for their suggestions and comments.

REFERENCES

- [1] D. K. Bhavsar and R. W. Heckelman, "Self-testing by polynomial division," in *Proc. Int. Test Conf.*, Oct. 1981, pp. 208-216.

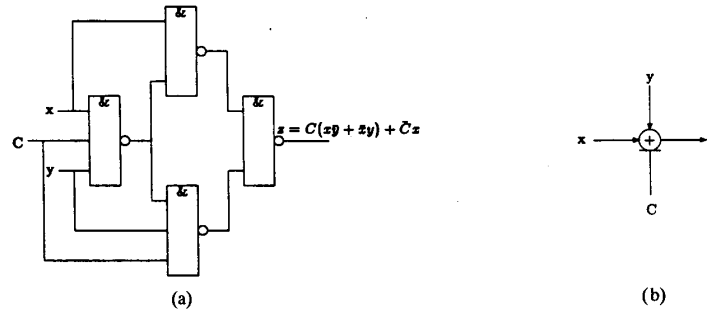


Fig. 4. (a) The expanded XOR gate. (b) Its symbol.

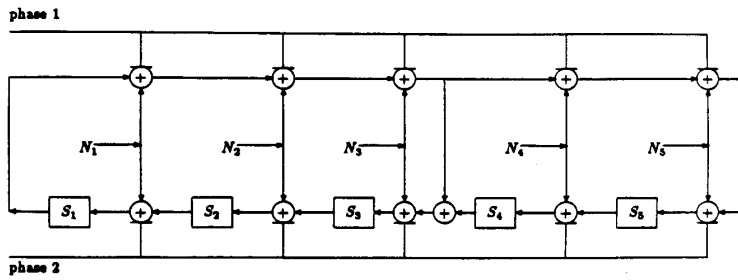


Fig. 5. The PSA design with expanded XOR gates.

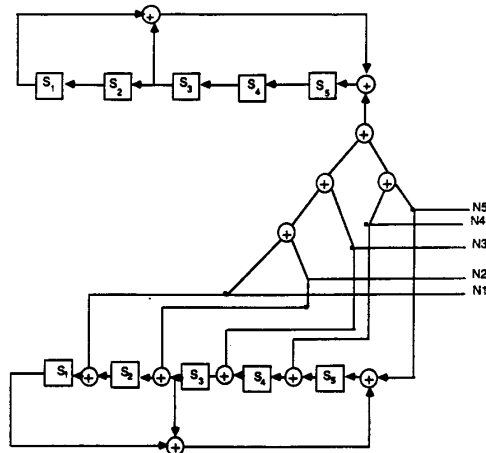


Fig. 6. A PSA-SF-SI/PSA-SF-MI-based design to eliminate aliasing double errors.

- [2] L.-T. Wang and E.J. McCluskey, "Condensed linear feedback shift register (LFSR) testing—A pseudoexhaustive test technique," *IEEE Trans. Comput.*, vol. C-35, pp. 367–370, Apr. 1986.
- [3] K.K. Saluja and M. Karpovsky, "Testing computer hardware through data compression in space and time," in *Proc. Int. Test Conf.*, Oct. 1983, pp. 83–88.
- [4] S.Z. Hassan, D.J. Lu, and E.J. McCluskey, "Parallel signature analyzers—Detection capability and extensions," in *Proc. COMPCON Spring '83*, Feb. 1983, pp. 440–445.
- [5] B. Jin and Y. Min, "Alias rate of parallel signature analyzers," in *Proc. Forth Nat. Conf. CAD*, Zhang Zhou, China, 1986.
- [6] S.Z. Hassan and E.J. McCluskey, "Increased fault coverage through multiple signatures," in *Proc. 14th Fault-Tolerant Comput. Symp.*, Kissimmee, FL, June 1984.
- [7] A. Raghemi-Azar and P.C. Maxwell, "Suitable polynomial for aliasing reduction by test vector reversal in signature analyzers," *Electron. Lett.*, vol. 22, p. 958, June 1986.
- [8] T.W. Williams, W. Daehn, M. Gruetzner, and C.W. Starke, "Aliasing errors in signature analysis registers," *IEEE Design Test*, Apr. 1987.
- [9] S.K. Gupta and D.K. Pradhan, "A new framework for designing and analyzing BIST technique: Computation of exact aliasing probability," in *Proc. ITC*, Sept. 1988, pp. 329–342.
- [10] J.E. Smith, "Measures of the effectiveness of fault signature analysis," *IEEE Trans. Comput.*, vol. C-29, pp. 510–514, June 1980.
- [11] L. Shen and S.Y.H. Su, "Parallel signature analyzers with external XOR gates," in *Proc. 1986 Nat. Test Conf.*, Beijing, China, 1986.
- [12] Y. Min, Y.K. Malaiya, and B.P. Jin, "Enhancing detection capability of signature analyzers," Tech. Rep. FTC-86-AD, Dep. Comput. Sci., Colorado State Univ., Dec. 1986.
- [13] P.C. Maxwell, "Comparative analysis of different implementations of multiple-input signature analyzers," *IEEE Trans. Comput.*, vol. 37, pp. 1411–1414, 1988.
- [14] T. Sridhar, D.S. Ho, T.J. Powell, and S.M. Thatte, "Analysis and simulation of parallel signature analyzers," in *Proc. IEEE Int. Test Conf.*, Nov. 1982, pp. 656–661.
- [15] K. Iwasaki, N. Yamaaguchi, and T. Nishimuki, "Analysis and proposal of signature circuit for LSI testing," in *Proc. ITC*, Sept. 1987, pp. 517–522.
- [16] A. Ivanov and V. Agarwal, "On a fast method to monitor the behavior of signature analysis registers," in *Proc. ITC*, Sept. 1987, pp. 645–655.

Coping with Erroneous Information while Sorting

K. B. Lakshmanan, B. Ravikumar, and K. Ganesan

Abstract— In this correspondence, we study the problem of sorting n distinct elements in ascending sequence according to a total order, using comparison queries which receive "yes" or "no" answers, but as many as e of which may be erroneous. In a half-lie version, all "yes" answers are guaranteed to be correct and the errors are confined to "no" answers. We show that the comparison query complexity of the sorting problem for this case is $\Omega(n \log n + e)$ and then demonstrate an asymptotically optimal algorithm. In a full-lie version, however, both "yes" and "no" answers can be false. We show that the comparison query complexity of the sorting problem for this case is $\Omega(n \log n + en)$ and then demonstrate an algorithm which generalizes the familiar binary insertion sort technique. This algorithm is asymptotically optimal if $e = O(n)$.

Index Terms—Adversary strategy, analysis of algorithms, binary insertion sort, comparisons, errors, fault tolerance, intermittent failure, lies, misinformation, noise, sorting, sorting networks

I. INTRODUCTION

The problem of *sorting* we are interested in can be stated as follows. Given a set $X = \{x_1, x_2, \dots, x_n\}$ of n distinct elements, arrange them in ascending sequence according to a specified total order. The elementary operation we will focus on for complexity analysis is the *binary comparison* of the form "Is $x_i < x_j$?" for $i, j \in \{1, 2, \dots, n\}$, with two possible outcomes "yes" and "no." It is well-known that any sorting algorithm requires at least $\lceil \log n! \rceil = \Omega(n \log n)$ comparisons in the worst case.¹ In this correspondence, we consider a situation where we may receive *erroneous information* in response to some comparisons, due to hardware intermittent failure, noise in communication channel, deliberate misinformation, etc. [1]. By an erroneous response we mean that the outcome of the comparison is "no" when factually it should be "yes," and vice versa.

In order to present the adversary-based lower bound arguments we will use later, it is convenient to restate the problem in the form of a *two-person game* between a questioner and a responder. The questioner has a set $X = \{x_1, x_2, \dots, x_n\}$ of n arbitrary elements from a universal set U on which a total order exists, but is known only to the responder. The questioner is required to sort the elements of the set X according to this total order with the help of the responder, seeking responses to binary comparison queries of the form "Is $x_i < x_j$?" for some $i, j \in \{1, 2, \dots, n\}$. Now, erroneous responses to comparison queries arise if the responder is allowed to *lie*.

It is intuitively clear that the questioner can make no progress in sorting if the responder is allowed to lie in an unconstrained way. Hence, we assume that the responder can lie at most e times, where e can be a function of n but is known to both the questioner and the

Manuscript received October 20, 1988; revised June 22, 1989. This work was supported in part by the Natural Sciences and Engineering Research Council of Canada under Grant A9194.

K. B. Lakshmanan was with the Department of Computer Science, Concordia University, Montreal, P.Q. H3G 1M8 Canada. He is now with the Department of Computer Science, State University of New York, Brockport, NY 14420.

B. Ravikumar is with the Department of Computer Science, University of Rhode Island, Kingston, RI 02881.

K. Ganesan is with the Department of Computer Science, Florida Atlantic University, Boca Raton, FL 33431.

IEEE Log Number 9100080.

¹All logarithms in this correspondence are to base 2.