Low-Cost On-Line Test for Digital Filters *

Ismet Bayraktaroglu
Computer Science & Engineering Department
University of California, San Diego
La Jolla, CA 92093
ibayrakt@cs.ucsd.edu

Alex Orailoglu
Computer Science & Engineering Department
University of California, San Diego
La Jolla, CA 92093
alex@cs.ucsd.edu

Abstract

A low-cost on-line test scheme for digital filters is proposed. The scheme uses an invariant of the digital filter, the frequency response at specific points, in order to detect possible malfunctioning of the circuit. The analysis performed indicates that 100% fault secureness is possible, if certain design constraints are followed.

1. Introduction

Digital Signal Processing has been used in a wide variety of applications over the last two decades. While certain types of applications do not require on-line testing capabilities, critical applications such as military, medical and long lifecycle applications, such as communication satellites, require concurrent fault detection. The latency tolerable for on-line fault detection is application-dependent. While medical applications do not tolerate any delay in fault detection, satellite applications may tolerate limited latency.

Additionally, applications that do not require on-line test may utilize low-cost concurrent fault detection for quality improvement purposes. For example, in the case of a multimedia application, a fault may deteriorate the quality of the result only, without necessarily resulting in a catastrophic failure. Detection of this class of faults improves service quality and consumer satisfaction for high quality manufacturers.

Filters constitute the main building blocks of signal processing applications. Digital filters nowadays are utilized in place of analog ones whenever possible, since they are easier to design, introduce less noise, and fit well with today's powerful digital designs.

In this study, we propose a low-cost on-line test solution for digital filters, which reduces the cost of on-line test hardware by trading off some latency to fault detection. There exist various alternative techniques, such as duplication, error coding, or redundant residue systems. However, none of them is capable of producing a similarly low-cost, moderate latency solution.

On-line test techniques require a decision-making scheme to determine whether the output of a circuit is correct or not. The simplest but possibly the most expensive solution is duplication, wherein the functionality of the circuit is duplicated through either structural or algorithmic duplication. A cheaper solution is the use of error codes [12]. In the case of parity codes, for example, the parity of the outputs is compared to the expected parity, derived from the parity of the inputs.

The method proposed in this study utilizes a functional invariance of a digital filter, and is therefore comparable to algorithm-based fault detection [7]. Furthermore, the proposed on-line monitoring of the inputs and outputs of the circuit is comparable to input vector monitoring techniques [11]. In contrast to algorithm-based techniques which utilize certain relationships between the input and output space at a given time instance, the proposed scheme utilizes an input/output relation, which holds only if a wide time range is considered. Use of such a technique introduces latency to the fault detection process. However, the latency is reduced if the effect of a fault to the output is large in magnitude, because the time required to determine the violation of the property is appreciably shorter than the time required to determine that the property holds. The faults with low-magnitude effect, on the other hand, are detected in a longer time period. In contrast to our approach, input vector monitoring techniques, independent of the fault effect, have consistently higher latency in fault detection. Furthermore, the scheme we propose has appreciably lower area overhead than comparable methods.

In this study, digital filters are investigated in order to prove the effectiveness of the proposed technique that relies on a time-extended invariance. The results indicate that near 100% fault secureness is possible within a reasonable latency with low area overhead if certain design constraints

*This work is being supported by Hughes Space and Communications and the UC Micro Program.

are followed. The method is applied to both high and low-pass filters and is proven to be effective. Furthermore, the hardware added for on-line test may also be shared with a low-cost off-line BIST solution which utilizes arithmetic pattern generators and response compactors [10].

We illustrate in section 2 examples of previously developed off-line and on-line test schemes, while section 3 motivates the rationale and the application areas for the proposed scheme. In section 4, the determination of the proposed invariant is outlined. This discussion is further extended in section 5 to include how the invariant is utilized in the study. Section 6 provides results and associated explanatory remarks. Furthermore, concluding remarks and possible extensions of the applicable arenas of the proposed invariant-based approach are provided in section 7.

2. Previous Work

Off-line test schemes developed for digital filters outweigh comparable schemes developed for on-line test. Examples of off-line test schemes can be found in [2, 4, 6]. In [2], a short sequence of functional test patterns is proposed, providing fault coverages in excess of 90%; high functional coverage is proven through analysis and high structural coverage is substantiated through fault simulations. The work in [4] develops built-in self-test techniques capable of 100% fault coverage through RT and gate level optimizations that eliminate redundant and random-pattern resistant faults. In [6] a low-cost, parameterizable BIST solution for digital filters is developed through utilization of arithmetic pattern generators.

The work in the area of on-line test of digital filters is concentrated in three categories: use of redundant residue number systems [3, 9], algorithm-based approaches [7], and use of simplified functionality [1]. Use of redundant residue number systems is only applicable to implementations that already make use of residue number systems, typically utilized to obtain high performance. In [1], linear functions are utilized as a checking circuitry for nonlinear functions, reducing the area overhead below the level of duplication. However, the technique is only applied to nonlinear applications. Finally, the algorithm-based approaches come closest to the technique proposed in this paper. In [7], *Parseval's theorem* [8] is utilized for the FFT algorithm so as to achieve concurrent detection of faults. A similar approach is also utilized in [7] for QR factorization. However, the proposed technique is applicable only when there exists a time-independent relation between the input and the output, thus limiting its applicability.

A general approach developed in [5] utilizes the idle clock cycles of the functional units for on-line BIST. A low area and performance overhead is claimed; the proposed method is inapplicable for high performance DSP applications though, as the hardware units are typically fully utilized. A comparable approach, input vector monitoring, is proposed in [11]. The method selectively analyzes the outputs of the circuit to attain a complete input space coverage. In order to provide a level of on-line fault coverage, the approach relies on a predetermined set of BIST vectors being matched in an input stream. The considerably high number of vectors and related probabilistic nature of the proposed algorithm force inordinately high latencies while the inability to guarantee uninterrupted execution of sequences of vectors limits its applicability to combinational circuits only.

3. Motivation

Not only are most of the techniques previously developed for on-line test of digital filters limited, but additionally, they either require high area overhead or are solely applicable to high performance filters that utilize a residue number system. The algorithm-based approaches can furthermore be applied only when a time-independent relation exists between the input and the output. To surpass such limitations, we introduce in this study a low-cost technique that is applicable even when such invariant relations are not time independent. Even though the relation depends on time, by accepting a small inaccuracy, the time range of the relation can be reduced. While introducing inaccuracy results in delays in fault detection, the technique can still be used as a low-cost, moderate latency on-line test method, if fault masking is prevented. An alternative approach that would reduce the time dependency of the input/output relation would need to rely on reversing the filter's effect on the input, more specifically by introducing an inverse filter at the same cost as that of duplication.

The relation between the inputs and the outputs of a filter being utilized in this work ($\sum y_n = \sum_{k=0}^{M} h_k \sum x_n$) is examined in detail in section 4. The relation only holds if the summation range for x and y extends from $-\infty$ to $+\infty$. Limiting the summation range from time 0 to current time imposes a tolerance to the relation which decreases gradually as the number of patterns increases.

4 Invariant Determination

The functionality of an FIR filter is defined by the following equation.

$$y[n] = \sum_{k=0}^{M} h_k x[n-k] \tag{1}$$

In this equation, the h_k's constitute the coefficients that govern the output characteristics of the filter. M represents the

order of the filter; as M gets larger, the frequency response gets better at the expense of increased cost of implementation. Calculation of the frequency response of a filter can be performed by taking the *Fourier* transform of the transfer function. Even though calculation of the frequency response requires complex operations, its value at certain frequencies can be derived easily. We utilize the frequency response of the filter at $w = 0$ ($F(w)|_{w=0}$) in order to take advantage of the associated computational simplicity at that point. Furthermore, the unidirectional summation nature of equation 2 enables possible elimination of fault masking effects.

$$F(w)|_{w=0} = \sum_{k=0}^{M} h_k \tag{2}$$

The response of a filter at $w = 0$ corresponds to the DC gain of the filter, which is equal to the sum of the filter coefficients. In order to be able to derive equation 2, we start by taking the infinite summation of both sides of equation 1. Then we change the order of summation and eliminate k from $x[n-k]$ since the summation is over an infinite range. We conclude by dividing both sides of the equation by $\sum_{n=-\infty}^{+\infty} x[n]$ to arrive at equation 3.

$$\begin{aligned} \sum_{n=-\infty}^{+\infty} y[n] &= \sum_{n=-\infty}^{+\infty} \sum_{k=0}^{M} h_k x[n-k] \\ &= \left(\sum_{k=0}^{M} h[k] \right) \sum_{n=-\infty}^{+\infty} x[n] \\ \sum_{k=0}^{M} h[k] &= \frac{\sum_{n=-\infty}^{+\infty} y[n]}{\sum_{n=-\infty}^{+\infty} x[n]} \end{aligned} \tag{3}$$

The ratio of the sum of the outputs to the sum of the inputs is always equal to the sum of the filter coefficients, $\sum_{k=0}^{M} h[k]$. However, for a relation to be applicable in an on-line test environment, it should be determinable from a limited set of input/output values. The same relation derived over a limited time range yields:

$$\begin{aligned} \sum_{n=0}^{N} y[n] &= \sum_{k=0}^{M} h_k \sum_{n=0}^{N} x[n-k] \\ &= \sum_{k=0}^{M} h_k \sum_{n=0}^{N} x[n] + Tolerance \end{aligned} \tag{4}$$

where

$$Tolerance = \sum_{k=0}^{M} h_k \left(\sum_{n=-k}^{-1} x[n] - \sum_{n=N-k+1}^{N} x[n] \right) \tag{5}$$

and

$$|Tolerance| \leq 2 \sum_{k=1}^{M} k h_k x_{max}$$

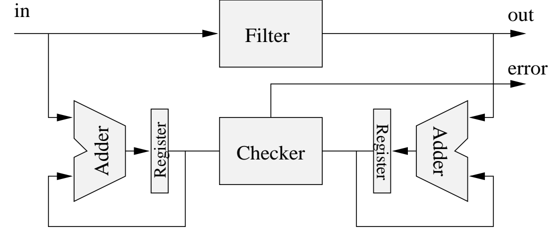

Figure 1. On-line test: Filter

where x_{max} denotes the maximum input signal magnitude. $Tolerance$ depends only on filter coefficients and the maximum input magnitude. As can be seen in equation 4, an increase in the number of inputs results in a diminution of the effective impact of tolerance on invariance uncertainty. In the next section, we propose an implementation of an on-line test scheme for digital filters which has a low cost and a reasonable latency in fault detection.

5. Method

Figure 2 illustrates the transposed direct form implementation of a Finite Impulse Response (FIR) filter. The proposed method is first applied to a 13 tap low-pass filter. A fractional number system of the form 1.11 (1 for the sign, and 11 bits for the fraction) is chosen for the input. The output is truncated to be in the form of 1.13. The filter coefficients are converted to canonical-signed-digit form and the constant multipliers are implemented as a number of shift-add operations.

The relation derived in the previous section is utilized as in the following form:

$$\left| \sum_{k=0}^{M} h_k \sum_{n=0}^{N} x_n - \sum_{n=0}^{N} y_n \right| \leq Tolerance \tag{6}$$

This form of the relation is applicable to filters with arbitrary frequency response. As shown in figure 1, an adder and a register is introduced both to the input and output of the circuit in order to calculate $\sum_{n=0}^{N} x_n$ and $\sum_{n=0}^{N} y_n$. The accumulated input is multiplied with $\sum_{k=0}^{M} h_k$ and the accumulated output is subtracted. The absolute value of the difference is compared subsequently with the off-line computed tolerance. If the result is not within the acceptable range, an error is reported. While equation 6 establishes a sufficient theoretical basis for this technique, its applicability to real-life designs needs to be undertaken through an analysis of its fault detection capabilities. In such an analysis, we need to be sensitive to the fault masking effects.

The scheme checks correctness of the filter output by observing the ratio of the accumulated outputs to accumulated inputs, which converges to the sum of the filter coefficients

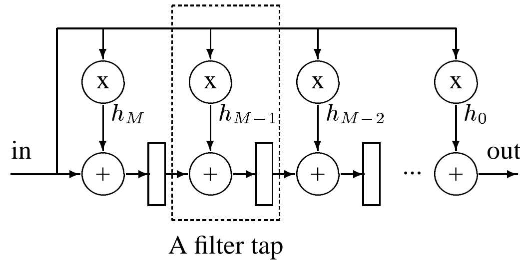

Figure 2. Transposed Direct Form

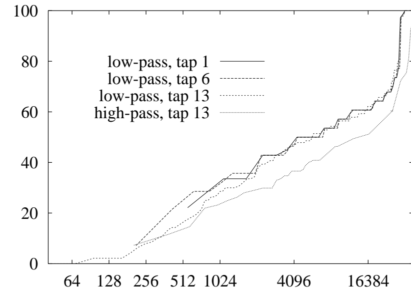

Figure 3. Number of patterns versus fault coverage

when there exists no erroneous output. If a fault produces an erroneous output, effects of this fault will be accumulated at the output accumulator; as the accumulated effect of the erroneous behavior exceeds the tolerance determined by equation 5, the fault will be detected. It can be argued that the effect of the same fault can mask itself; such a event can occur if the fault effect is not consistently in one direction. Fault masking effects are discussed further in section 6.

In addition to fault masking effects, a couple of additional points need to be considered in order to expose the efficacy of the method. We start with a discussion of the underlying adders. The effect of a fault at the nth bit position of an adder to the output will be 2^{n-B}, where B is the output width of the filter. The tolerance of the filter used in this study is around 2^3, as determined by equation 5. Therefore, in order for a fault at the nth bit position to be detected, it has to be activated (and observed at the output) 2^{B+3-n} times; the faults at the sign-bit position will be detected after 8 activations and the faults at the least significant bit will be activated after 2^{17} patterns. Assuming that the least significant bits are activated at a 50% rate, a fault at the least significant bit will require 2^{18} patterns to be detected. This wide divergence in the number of patterns required is correlated with the magnitude of the fault effect. In addition to the positive correlation with fault effect magnitude, the likelihood of fault activation also impacts detection latency. If a fault is unlikely to be activated, it will not be rapidly detected. Transient faults are unlikely to be detected by our scheme unless they are persistent or show fault effects of high magnitude.

The theoretical results indicate that the detection time of a fault depends on the magnitude of the effect of the fault and the activation probability of the fault. However, determination of the precise activation probability of a fault is not possible without specific knowledge of input patterns, since in an on-line application no prior knowledge of the input patterns exists. Therefore, we instead utilize simulation-based techniques with random patterns in order to verify our scheme. Fault simulations are performed in this study in order to determine the fault secureness level. The setup in Figure 1 is utilized during fault simulations by assuming that only the *error* output is observable. The *error* output is always precisely $zero$ for a fault-free circuit. Detection of a fault during fault simulations indicates that the fault causes the *error* output to attain the value of one, guaranteeing that the fault will be detected while the circuit is operating. Undetected faults, on the other hand, may not degrade the fault secureness of the scheme, since it may be the case that they have never been activated by these random patterns. Therefore, we also perform fault simulations in which all outputs are assumed to be observable. The ratio of the fault coverage of the former simulations to the latter indicates the actual fault secureness level.

Fault simulation, in general, is a computationally complex process. Additionally, in our case, the limited output observability and the sequential loops introduced by the accumulators cause the fault effect to remain in the circuit for a long time prior to its detection and thus increase the complexity of the fault simulations. In addition to the complexity, the high number of patterns involved increases simulation time. Therefore, a deterministic fault sampling technique is utilized in order to guarantee coverage of faults at each bit position and reduce the complexity of fault simulations. Fault simulations are performed on three adders selected at the leftmost, middle, and the rightmost position of the filter taps. The consequent simulation results are reported in the following section.

6. Results

The filters in this study are implemented in the transposed direct form, shown in Figure 2. The constant multipliers are implemented as a series of shift-add operations, after the coefficients are converted into the canonical-signed-digit format. Due to the implementation style of the filter, no resource sharing is possible, since each functional unit is used every clock cycle.

Two 13-tap filters, one high-pass and one-low pass, are

	Bit positions in the adder													
Patterns	13	12	11	10	9	8	7	6	5	4	3	2	1	0
128	4	2												
512	4	8	10	4										
1014	8	10	10	10	6	1								
4096	8	10	10	10	10	10	6	1						
16384	8	10	10	10	10	10	10	10	6	1				
Total	8	10	10	10	10	10	10	10	10	10	10	10	10	10

Table 1. Detected fault locations

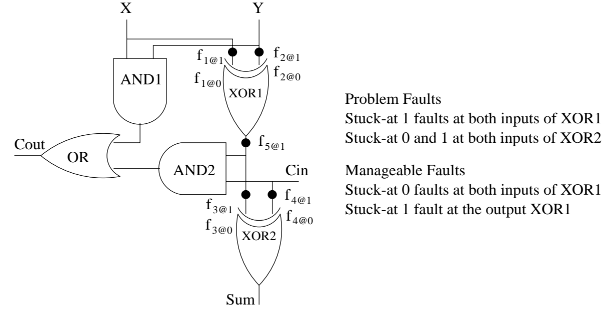

Figure 4. A typical full-adder implementation

implemented and simulations are performed on both filters. The filters are optimized and mapped to the LSI_10k library by a commercial logic synthesis tool. The fault simulations are also performed using the same tool. Figure 3 shows the results of fault simulation performed on both filters. The results indicate that fault detection increases as expected up to the 80% range. At that point, there is a sudden jump in fault coverage figures. A possible explanation for the jump is that the activation probability of the faults at the low bits is quite high. However, further detailed analysis indicates that an additional reason for this unexpected jump consists of the numerical inaccuracies caused by truncation errors. Truncation errors, in addition to helping reduce the latency for the faults at the low order bits, may cause false alarms. The absolute tolerance approach being utilized in the reported study improves fault simulation speed; the applicability of the approach is additionally improved as the absolute tolerance method enables handling high-pass and band-pass filters and varying phase inputs. The use of a relative tolerance[1],while capable of solving the problems caused by truncation errors, forces disabling of on-line error checking for a number of patterns, increasing both fault simulation complexity and fault detection latency. A combination of the two schemes, on the other hand, both improves the fault simulation speed and possibly eliminates false alarms.

Figure 3 shows the percentage of faults detected at particular pattern counts only. This figure itself is not sufficient to support the hypothesis that the faults at the high bits are detected in a shorter time that the ones at the low bits. In order to further differentiate among high versus low bit faults, analysis of the bit positions of the faults caught at analogous pattern counts needs to be undertaken. We report on this analysis in table 1, which supports the magnitude-based differential latency aspect, outlined in section 5.

The impact of the scheme on performance is negligible since the scheme solely observes the inputs and outputs of the circuit. The area overhead, on the other hand, is not negligible, particularly for smaller designs. For example, the area overhead for the experimental benchmarks is around 25%. Nonetheless, for larger filters the same area overhead will constitute an appreciably smaller percentage of the design. In a larger design, such as a 60 tap filter, the area overhead would be only 5% since it is independent of the filter structure. Additionally, the input and output accumulators (constituting 50% of the total overhead) can be utilized as off-line BIST hardware.

[1] $\left(\sum h_k - \%T\right)\sum_0^N x_n \leq \sum_0^N y_n \leq \left(\sum h_k + \%T\right)\sum_0^N x_n$

6.1. Validation of Unidirectional Fault Effect

In this section, we examine the effects of the faults in order to be able to prove that they are not being masked while being accumulated. We start by an examination of the effect of the faults from the RT level. At the RT level, without resource sharing, the linearity of the filter structure guarantees that unidirectional fault effects at the block level result in unidirectional output effects. In other words, the RT components are required to have unidirectional fault effects. Ripple-carry adders are utilized in this work. A fault at the ith bit full-adder effects the output of the adder by 2^i or 2^{i+1} depending on whether the fault is on the carry output or not. If the fault is a stuck-at-0 fault, then the output is always less then or equal to the correct output. If, on the other hand, the fault is a stuck-at-1 fault, then the faulty output will always be greater than or equal to the fault-free output. The same conclusion is reached upon considering register faults. However, these conclusions are only valid for the faults at the inputs and outputs of the full-adders and of the flip-flops. The commercial tool utilized introduces faults only at the inputs and the outputs of these cells, since they are defined as primitive cells in the LSI_10k library. Under this fault model, the proposed scheme produces 100% fault secureness.

An idealized analysis of the effects of the faults internal to the primitive cells is futher undertaken in this study. A simple-full adder implementation, as illustrated in Figure 4, is utilized for this purpose. Fault simulation is performed in order to record the outputs of this full adder circuit for all possible faults. The simulation results indicate that out of 26 faults simulated, 9 violate the unidirectional fault effect requirement. The simulation results for these faults are

Inputs	Output	f_{10}	f_{20}	f_{51}	f_{11}	f_{21}	f_{30}	f_{40}	f_{31}	f_{41}
000	0	0	0	1	1	1	0	0	1	1
001	1	1	1	2	2	2	1	0	0	1
010	1	1	0	1	0	1	0	1	1	0
011	2	2	1	2	1	2	3	3	2	2
100	1	0	1	1	1	0	0	1	1	0
101	2	1	2	2	2	1	3	3	2	2
110	2	3	3	3	2	2	2	2	3	3
111	3	2	2	2	3	3	3	2	2	3

Table 2. Full-adder outputs for certain faults

reported in table 2. Three of these faults possess unequal numbers of patterns that shift the output in divergent directions. The remaining six have equal number of shifts in either direction; those are identified as "problem faults" in figure 4. Assuming equiprobable input streams at the inputs of these adders[2], the six faults would have decreased the fault coverage levels to 80%. However, NAND/NAND implementation of the XOR gates in the depicted full-adder cell reduces the number of faults that violate the unidirectional effect property to three, one of them still being $f_{5@1}$. Further studies examining both the feasibility and cost of alternative adder implementations displaying solely unidirectional fault effects can be undertaken.

7. Conclusion

On-line test is becoming essential as we rapidly move towards sub-micron technologies. Simple techniques developed for on-line test, while providing zero latency solution, necessitate intolerable area overheads. High latency solutions, on the other hand, are unacceptable for most applications. We propose a simple method capable of providing a moderate latency, low-cost solution. While the approach is applied to digital filters in this study, it can be analogously applied to linear, pipelined datapaths.

In this paper, we propose for on-line digital filter test a functional property that is invariant over a wide time range. When considered in a shorter time range for actual on-line execution, the invariance leads to tolerance effects that need to be considered during analysis. Selection of a proper tolerance guarantees that no false alarms exist.

In order to verify the proposed scheme, a test setup is created in order to evaluate the proposed scheme. Intelligent fault sampling schemes are utilized in order to be able to deal with the inordinate time complexity of the problem. Though initial fault simulations indicate very high fault coverages, further analysis of the schemes indicates that fault masking can degrade fault coverage. In order to eliminate the fault masking problem, the primitive components of the circuits need to be modified such that the fault effects at the output are always in the same direction.

An interesting future extension consists of applying the proposed invariance-based approach to linear pipelined datapaths. In such datapaths, the relation between the input and the output is time dependent unless a number of successive inputs in excess of the pipeline depth are stored. For these circuits, the time average behavior of the circuit may be investigated as a low-cost on-line test solution. Achieving similarly strong results in such alternative domains may help establish the proposed invariance-based technique as an alternative to existing on-line test techniques, especially for cost-critical applications.

[2]The likelihood of the equiprobability assumption wanes as embedding levels for the adders increase.

References

[1] A. Chatterjee and R. K. Roy. Concurrent error detection in nonlinear digital circuits with applications to adaptive filters. In *IEEE International Conference on Computer Design*, pages 606–609, 1993.

[2] C. Counil and G. Cambon. A functional BIST approach for FIR digital filters. In *VLSI Test Symposium*, pages 90–94, 1992.

[3] M. H. Etzel and W. K. Jenkins. Redundant residue number systems for error detection and correction in digital filters. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 28(5):538–545, October 1980.

[4] L. Goodby and A. Orailoğlu. Towards 100% testable FIR digital filters. In *International Test Conference*, pages 394–402, Oct. 1995.

[5] R. Karri and N. Mukherjee. Versatile BIST: An integrated approach to on-line/off-line BIST. In *International Test Conference*, pages 910–917, October 1998.

[6] N. Mukherjee, J. Rajski, and J. Tyszer. Parameterizable testing scheme for FIR filters. In *International Test Conference*, pages 894–902, October 1997.

[7] A. L. Narasimha Reddy and P. Banerjee. Algorithm-based fault detection for signal processing applications. *IEEE Transactions on Computers*, 39(10):1304–1308, October 1990.

[8] A. V. Oppenheim and R. W. Schafer. *Discrete-Time Signal Processing*. Prentice Hall, 1989.

[9] G. A. Orton, L. E. Peppard, and S. E. Tavares. New fault tolerant techniques for residue number systems. *IEEE Transactions on Computers*, 41(11):1453–1464, November 1992.

[10] J. Rajski and J. Tyszer. *Arithmetic built-in self-test for embedded systems*. Prentice Hall, 1998.

[11] I. Voyiatzis, A. Paschalis, D. Nikolos, and C. Halatsis. R_CBIST: An effective RAM-based input vector monitoring concurrent BIST technique. In *International Test Conference*, pages 918–925, October 1998.

[12] J. F. Wakerly. *Error Detecting Codes, Self-Checking Circuits and Applications*. North-Holland, 1978.