An Efficient BIST scheme for High-Speed Adders

D. G. Nikolos[1], D. Nikolos[1,2], H. T. Vergos[1,2] & C. Efstathiou[3]
[1]Computer Engineering and Informatics Dept., University of Partas, 26500 Patras, Greece
[2]Computer Technology Institute, 3 Kolokotroni Str., 26221 Patras, Greece
[3]Informatics Dept., TEI of Athens, Ag. Spyridonos St., 12210 Egaleo, Athens, Greece
e:mail: nikolos@ceid.upatras.gr, nikolosd@cti.gr, vergos@cti.gr & cefsta@teiath.gr

Abstract

In this paper we present a new pseudorandom BIST scheme for high-speed adders. Under this scheme an adder is simultaneously used as a test pattern generator and as a response compactor during its own testing. The main advantages of the proposed scheme, compared to prior methods, are minimal performance penalty, small hardware overhead and the benefits of at-speed testing.

1. Introduction

Built-In Self-Test (BIST) [1] is an effective approach for testing integrated circuits (ICs) that reduces the need for external testing, since the circuit and its tester are implemented in the same chip, enabling the circuit to test itself. The main components of a BIST scheme are: the Test Pattern Generator (TPG), the Test Response Compactor (TRC) and the BIST controller.

The primary parameters that must be considered when developing a BIST methodology are the following [2]: *Fault Coverage*: the fraction of modeled faults that can be exposed by the patterns produced by test generation and detected by test response compaction. Safety-critical applications require very high fault coverage, typically 100% of the modeled faults. *Test set size*: the number of test patterns produced by the TPG. This parameter is linked to fault coverage. Generally, large test sets imply high fault coverage. Besides, large test sets detect more unmodeled faults. *Hardware Overhead*: the extra hardware needed for BIST. In most applications, high hardware overhead is not acceptable because of its impact on circuit size, packaging, power consumption, and cost. *Performance Penalty*: the impact on the operation speed of the circuit, such as increase in critical path delay, due to the inclusion of BIST hardware. This type of overhead is crucial especially in high-speed applications.

In this paper we consider the single stuck-at fault model. Although this fault model is not a very accurate representation of production defects, it has been shown [3] that most real defects are detected by the at-speed application of a test set derived under the single stuck-at fault model. The smaller the frequency at which the test vectors are applied, the larger the number of defects that escape testing [3].

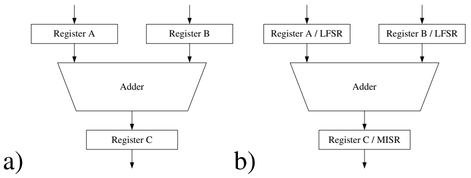


Figure 1. (a) Typical Adder (b) LFSR / MISR Based BIST Scheme

In this paper we consider BIST for high-speed adders. In high-speed circuits the test set size is not a very significant parameter, provided that it does not grow beyond acceptable sizes. On the other hand, in high-speed circuits the performance penalty is of critical importance. Minimal, if zero is impossible, performance penalty is required. The hardware overhead is also a significant parameter. Because of the high integration capabilities of the current manufacturing technologies the misconception that hardware is cheap has been spread. Given that a 20% increase of the chip area implies a 50% increase of cost [4], we perceive that hardware overhead reduction remains a significant target. The already known BIST schemes that can be applied to high-speed CLA and parallel prefix adders have several disadvantages. Figures 1 (a) and (b) respectively present an adder and the classical BIST scheme according to which the input and output registers are modified to function in test mode as LFSRs and a MISR respectively. The required modifications of the output register so as to function as a parallel in - parallel out register in normal mode and as a MISR in test mode, cause, as we will show in the last section, a significant performance penalty which is unacceptable in the case of the high-speed adders. The required modifications also impose large hardware overhead. An interesting BIST scheme (see Figure 2) for CLA adders was proposed in [2]. The great advantage of this scheme is that fault simulations

Proceedings of the 9th IEEE International On-Line Testing Symposium (IOLTS'03)
0-7695-1968-7/03 $17.00 © 2003 IEEE

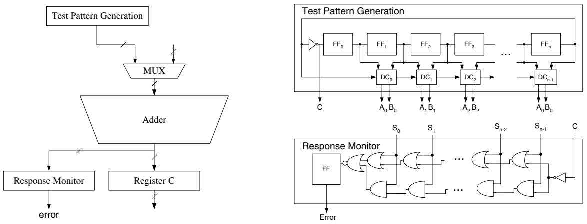


Figure 2. BIST Scheme for CLA Adders in [2]

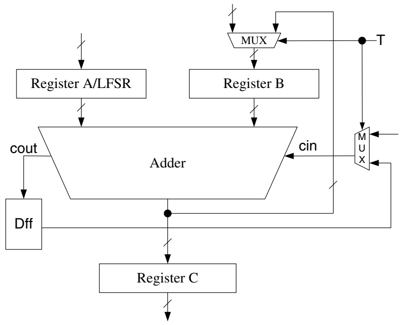


Figure 3. A Very Low Hardware Overhead BIST Scheme

i	Modulo $2^3 - 1$ adder			Stored Carry Bit adder				
	A_i	B_i	Out	A_i	B_i	Cin	Out	$Cout$
1	010	000	010	010	000	0	010	0
2	110	010	001	110	010	0	000	1
3	001	001	010	001	000	1	010	0
4	101	010	111	101	010	0	111	0
5	100	111	100	100	111	0	011	1
6	111	100	100	111	011	1	011	1
7	011	100	111	011	011	1	111	0
8	010	111	010	010	111	0	001	1
9	110	010	001	110	001	1	000	1
10	001	001	010	001	000	1	010	0

Table 1. Sequences Generated by a Modulo $2^n - 1$ Adder and a Stored Carry Bit Adder

are not required. However this scheme imposes large hardware overhead, equal to 40.1%, 36.9% and 35.8% for 8, 16 and 32-bit adders respectively [2]. Another problem of this scheme is that it is not suitable for at-speed testing. Modifications are required to enable at-speed testing. For example registers should be added between the multiplexers and the inputs of the adder. The required modifications increase the hardware overhead further.

In this paper we will present a minimal performance penalty BIST scheme with significantly smaller area overhead than the above-mentioned schemes. It is known that an accumulator can be used efficiently as a TPG [5] as well as a TRC [6][7]. In this paper we show that an adder in test mode can be used as an accumulator, which behaves simultaneously as a TPG and a TRC during its own testing. Two BIST schemes will be presented. The first proposed scheme has in logic level the lowest hardware overhead and offers 100% fault coverage in nearly all cases. The second, sacrificing slightly larger hardware overhead, achieves complete fault coverage with shorter test sequences than the first scheme.

2. Low Hardware Overhead BIST Schemes

Figure 3 presents a BIST scheme with very low hardware overhead, hereafter called VLHO BIST scheme. VLHO BIST scheme cannot be used for at-speed testing; it is given as a vehicle for our investigation. In the last subsection we will introduce a scheme suitable for at-speed testing. Since the accumulator with stored carry bit adder (rotate carry adder) [6] exhibits smaller aliasing than that of the classical accumulator, a flip-flop and a multiplexer have been inserted to enable the adder to behave as a stored carry bit accumulator in test mode. In test mode, at every clock cycle, the adder receives a test vector consisting of two parts; part A is the output of the LFSR while part B is the sum of all the previously received input vectors. At the end of testing register C contains the signature.

In the following we consider: (a) the maximum length of the test sequence generated at the inputs of the adder, (b) the pre-compaction fault coverage that can be achieved and (c) the post-compaction fault coverage (PCFC).

2.1. Test Sequence Length

Both stored carry bit adders and modulo $2^n - 1$ adders perform the operation of adding the produced carry output as a carry input. In modulo $2^n - 1$ addition, the addition of the carry output as a carry input is performed during the same clock cycle the addition of the operands is performed, while in stored carry bit addition the carry output is added as carry input during the next clock cycle. During an accumulation where each output is added as one of the inputs of the next cycle, the output of the stored carry bit adder will be either equal or smaller by one than the corresponding output of the modulo $2^n - 1$ adder depending on the carry produced by the last addition. The relationship between the two output sequences is illustrated in Table 1. We will use the sequence generated by a modulo $2^n - 1$ adder in order to study the sequence generated by the stored carry bit adder.

We consider that the LFSR in Figure 3 is characterized by a primitive polynomial of degree n. It is well known that

$$\left||x+y|_{2^n-1} + z\right|_{2^n-1} = |x+y+z|_{2^n-1} \ , \qquad (1)$$

where $|a|_m = a \bmod m$.

Let A_i and B_i denote the LFSR state and the contents of the output register of the modulo $2^n - 1$ adder at clock cycle i respectively. Since the LFSR implements a primitive polynomial, during the first $2^n - 1$ clock cycles it will generate all $2^n - 1$ distinct non-zero states. Therefore, the first $2^n - 1$ input vectors of the adder will be distinct. After the first $2^n - 1$ cycles the LFSR generates the same sequence from the beginning. This means that,

$$A_{2^n-1+k} = A_k, \text{ for } k = 0 \ldots 2^n - 1 \tag{2}$$

The content of output register C at clock cycle k is:

$$B_k = \left| \sum_{i=1}^{k-1} A_i \right|_{2^n-1}$$

Considering the $2^n - 1$ distinct LFSR states as decimal numbers we get the first $2^n - 1$ terms of an arithmetic series with common difference equal to one. Taking into account the sum of the first $2^n - 1$ terms of an arithmetic series with common difference one, as well as relation (1) we have:

$$B_{2^n} = \left| \frac{(1 + 2^n - 1) \cdot (2^n - 1)}{2} \right|_{2^n-1} = 0$$

In 1's complement arithmetic zero has two representations, the all 0s and the all 1s representation. The all 0s representation is produced only when all 0s vectors are added and since our LFSR never generates the all 0s vector, B_{2^n} has the all 1s representation.

Let B'_i and $Cout_i$ be the output and the carry out of the stored carry bit adder at clock cycle i respectively. There are two possibilities for the values of B'_{2^n} and $Cout_{2^n}$: either B'_{2^n} is $2^n - 1$ and $Cout_{2^n}$ is 0 or B'_{2^n} is $2^n - 2$ and $Cout_{2^n}$ is 1. In both cases the result of the next addition in the stored carry bit adder is $A_{2^n} + B'_{2^n} + Cout_{2^n} = A_1 + 2^n - 1$, which means that $B'_{2^n+1} = A_1 - 1$ and $Cout_{2^n+1} = 1$. Since $B'_1 = 0$ we have that $B'_2 = B'_1 + A_1 = A_1$, therefore we have that $B'_{2^n+1} \neq B'_2$.

The next addition result will be $A_{2^n+1} + B'_{2^n+1} + Cout_{2^n+1} = A_2 + A_1 - 1 + 1 = A_1 + A_2$. Thus, $B'_{2^n+2} = |A_1 + A_2|_{2^n}$ and $Cout_{2^n+2}$ is the carry out of the addition $A_1 + A_2$. Since during the third clock cycle A_1 and A_2 are added, $B'_3 = B'_{2^n+2}$ and $Cout_{2^n+2} = Cout_3$. From relation (2) we have that $A_3 = A_{2^n+2}$. Considering the circuit of Figure 3 as an FSM with its internal state determined by the contents of LFSR A, Register C and the flip-flop storing the carry out we can see that each state explicitly defines the next one. Since the $2^n + 2$ state is the same with the 3^{rd} state, the machine enters a loop after $2^n + 2$ cycles and it will never enter a state that has not been entered before. This means that the stored carry bit adder receives only $2^n + 1$ different test vectors when the VLHO BIST scheme is used.

2.2. Pre-Compaction Fault Coverage

In this paper CLA adders and Kogge-Stone, Ladner-Fischer and Han-Carlson parallel prefix adders are considered. For each of these adders and for sizes equal to 8, 16 and 32, pre-compaction fault simulation was performed. The practical result of the test sequence length that we have derived above is that we can stop the pre-compaction fault simulation after the application of $2^n + 1$ input vectors even if 100% fault coverage has not been achieved.

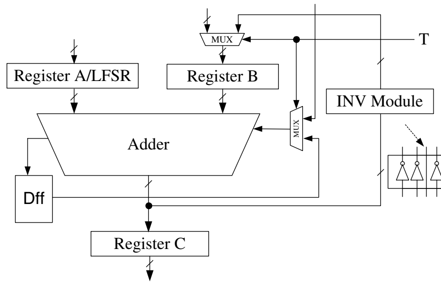


Figure 4. Partially Inverted Feedback BIST Scheme

The attained fault coverage and the test sequence lengths depend heavily on the chosen primitive polynomial. Polynomials with many terms give better results than polynomials with a small number of terms. Polynomials with terms spread over the range of the polynomial give also better results than polynomials with the same number of terms concentrated in a region.

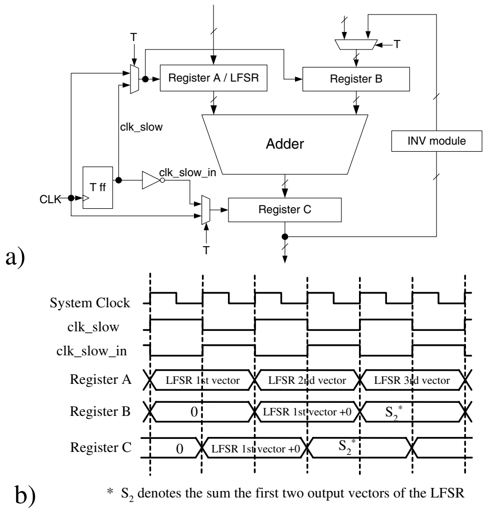
2.3. Post-Compaction Fault Coverage (PCFC)

By performing fault simulations we found out that 100% PCFC is achieved in most cases but with test sequence lengths significantly longer than those required by the LFSR / MISR based technique. In some cases the test sequence length exceeds the maximum test sequence length ($2^n + 1$ test vectors). This could not happen in a case where the TPG is a circuit distinct from the CUT. However in our case the CUT is also used as a TPG and a TRC. This means that the appearance of an error due to a fault, at the output of the adder influences the subsequent test vectors.

Experimental results are given in Table 3 and will be analyzed in the next section. The derived test sequences lengths may cause a test application time problem in the case of very large adders. To overcome this problem, a new BIST scheme, hereafter called Partially Inverted Feedback (PIF) BIST scheme will be presented.

2.4. Partially Inverted Feedback (PIF) BIST Scheme

In the PIF BIST scheme (see Figure 4) we can see that in test mode some of the bits of the output of the adder are inverted before being applied as a part of the next test vector to the adder. The selection of the bits to be inverted is a design parameter. In our experiments this selection was made in a random manner.



* S_2 denotes the sum the first two output vectors of the LFSR

Figure 5. a) Partially Inverted Feedback BIST Scheme b) Corresponding Waveforms

PIF BIST scheme requires slightly larger hardware overhead than the VLHO BIST scheme, but it achieves complete fault coverage with short test sequences. As we will show in the next section, using the PIF BIST scheme 100% PCFC is achieved with test sequence lengths similar to those of the LFSR / MISR based BIST scheme.

Figure 5 shows how the proposed PIF BIST scheme can be used to exploit the benefits of at speed testing. In at-speed testing, successive test patterns are applied to the circuit under test at the system clock rate with no interruption. In the scheme of Figure 5, a test vector is applied every other clock cycle, however, since the response of the adder is captured just one clock cycle after each test vector's application, the scheme detects all the defects detected in the case of at-speed testing. Removing from the block diagram of Figure 5 the block "INV Module" we get the corresponding VLHO BIST scheme.

3. Evaluation and Comparisons

In this section we compare the various BIST techniques using the metrics discussed earlier: performance penalty, hardware overhead, PCFC and test sequence lengths.

In order to get realistic estimates of the overhead imposed on a buffered adder, by the aforementioned BIST schemes, we modeled all of them in HDL for adders of 8, 16 or 32 bits wide. We present our results for two distinct adder architectures, namely, a two level CLA architecture and a parallel-prefix architecture with a Kogge - Stone prefix tree for the carry computation. Since an adder is usually embedded in a larger circuit, its inputs and outputs are not accessible by the primary inputs and outputs of the chip. For applying an external testing in such embedded circuits one usually relies on scan paths. For comparison reasons a scan path scheme has also been implemented. Each design was mapped in the UMC-VST 25 implementation technology ($0.25\mu m$, 1.8 / $3.3V$ interface, up to 5 metal layers for interconnects), using the Synopsys® Design Compiler tool. We optimized each adder recursively for speed, until the tool was unable to produce a faster design. Subsequent steps of area recovery followed. The adders were then characterized as "untouchable" blocks that were used for every scheme. The DFT logic of each technique was then optimized, considering it as a single flat design, using recursive timing optimization passes and a last area recovery step. The results gathered are listed in Table 2.

	Delay		Area	
	Value	Dif	Value	% inc
8-bit CLA				
Buf Adder	1.50		13614	
Scan Path	1.63	0.13	15293	12.33
LFSR/MISR	1.87	0.37	17724	30.19
VLHO	1.60	0.10	17371	27.60
PIF	1.60	0.10	16108	18.32
16-bit CLA				
Buf Adder	1.70		26470	
Scan Path	1.84	0.14	29168	10.04
LFSR/MISR	2.07	0.37	32798	23.91
VLHO	1.81	0.11	29792	12.55
PIF	1.81	0.11	29962	13.19
32-bit CLA				
Buf Adder	1.89		57665	
Scan Path	2.02	0.13	63289	9.75
LFSR/MISR	2.21	0.32	70256	21.83
VLHO	1.99	0.10	63930	10.86
PIF	1.99	0.10	63669	10.41
8-bit Kogge-Stone				
Buf Adder	1.42		16617	
Scan Path	1.55	0.13	18187	9.45
LFSR/MISR	1.81	0.39	20843	25.43
VLHO	1.52	0.10	19596	17.93
PIF	1.52	0.10	19170	15.36
16-bit Kogge-Stone				
Buf Adder	1.58		41369	
Scan Path	1.71	0.13	44265	7.00
LFSR/MISR	1.93	0.35	48548	17.35
VLHO	1.72	0.14	46100	11.44
PIF	1.72	0.14	46208	11.70
32-bit Kogge-Stone				
Buf Adder	1.74		88004	
Scan Path	1.87	0.13	92823	5.48
LFSR/MISR	2.07	0.33	100306	13.98
VLHO	1.85	0.11	95382	8.38
PIF	1.85	0.11	95136	8.10

Table 2. Area-Time Comparisons

From Table 2 we can see that the performance penalty of the proposed BIST schemes is roughly three times smaller than that imposed by the classical LFSR / MISR based BIST scheme and in most cases even smaller than that of the scan path scheme. We can also see that the hardware overhead of the proposed BIST schemes against the classical LFSR / MISR based one is significantly smaller. When $n = 8$, in some cases the test length is larger than the LFSR cycle

	LFSR / MISR		VLHO		PIF	
	Vectors	%cov	Vectors	%cov	Vectors	%cov
8-bit						
CLA	48	100	386	100	70	100
KS	31	100	198	100	49	100
LF	28	100	103	100	39	100
HC	31	100	94	100	44	100
16-bit						
KS	219	100	2732	100	513	100
LF	90	100	1696	100	66	100
HC	218	100	929	100	220	100
32-bit						
KS	193307	100	491120	99.77	185100	100
LF	9078	100	98434	100	7167	100
HC	193307	100	179231	100	163788	100

Table 3. Test Set Sizes and PCFC

	LFSR / MISR Based		VLHO		PIF	
	Avg	Dev	Avg	Dev	Avg	Dev
CLA	191	85	669	328	200	81
KS	112	52	295	213	127	52
LF	82	44	840	549	94	45
HC	95	46	348	248	104	48

Table 4. Average Number of Vectors - Deviations

length. A comparator that compares the LFSR state with the stored one is therefore not enough for detecting the last vector in this case and a larger control unit is required. As we can see from Tables 2 and 3 in those cases the hardware overhead is larger.

PCFC and test sequence lengths are derived using CLA and parallel prefix adders with a carry computation accordingly to Kogge-Stone, Ladner-Fischer and Han-Carlson. For collecting our results, 40 LFSR seeds were randomly generated for every circuit and technique. In the case of the PIF BIST scheme, 40 combinations of seeds and position of bits to be inverted were also randomly generated. For each of these cases we used 4 LFSR primitive polynomials. For the 160 different cases that we produced, the sequence applied to the circuit was generated, and pre-compaction fault coverage was calculated. For the best 4 attained results out of the 160 simulated pre-compaction results, the PCFC was calculated. The best post-compaction result out of these 4 is shown in Table 3.

From Table 3 we can see that the test sequence lengths of the PIF scheme are similar to those of the classical LFSR / MISR based scheme. However, in order to exploit the benefits of at-speed testing we have to use the BIST scheme of Figure 5. Then the test application time is equal to two times the test sequence length. The test application time remains extremely small. For example, for testing a 32-bit Kogge-Stone adder with critical path $1.85ns$ (Table 2) we need 185100 test vectors (Table 3) and only $185100 \cdot 2 \cdot 1.85ns = 684870ns \approx 685\mu s$ are required.

We remind that the design parameters in the classical LFSR / MISR based BIST scheme are the primitive polynomials of the LFSRs A and B as well as their seeds. In the case of PIF BIST scheme the parameters are the primitive polynomial of the LFSR A, its seed and the position of the inverted bits. In order to have an estimation of the effort required to find the parameters for a good solution (100% fault coverage and relatively short test sequence) for the LFSR / MISR, PIF and VLHO BIST schemes we performed a series of experiments.

For the LFSR / MISR based and VLHO BIST schemes we calculated the post-compaction fault-coverage for 4 polynomials and 40 seeds, that is 160 cases for each one of the 8-bit adders. For the PIF BIST scheme we calculated the post-compaction fault-coverage for 4 polynomials and 40 combinations of position of bits to be inverted and seeds, that is 160 cases for each one of the 8-bit adders as well. In all these cases 100% PCFC was achieved, and the minimum number of vectors, the average number of vectors and the standard deviation are shown in Table 4.

From Table 4 we can see that the average number of vectors, as well as the value of the standard deviation for the PIF BIST scheme are very close to the corresponding values of the classical LFSR / MISR based BIST scheme. Therefore, we conclude that the effort required for reaching a good solution is similar for both schemes. Since the effort required for the VLHO scheme is significantly larger, and taking into account the results of Table 2 we conclude that PIF is the most suitable BIST scheme for high-speed adders.

4. Conclusions

A new BIST scheme suitable for high-speed adders is proposed. The proposed scheme causes impressively smaller performance penalty and imposes significantly smaller hardware overhead than the LFSR / MISR based scheme while requiring similar test sequence lengths to achieve 100% fault coverage.

References

[1] M. Abramovici, M. A. Breuer, and A. D. Friedman. *Digital Systems Testing and Testable Design*. CSP, 1990.

[2] H. Al-Asaad et. al. Scalable test generators for high-speed datapath circuits. *JETTA*, 12(1-2):111–125, Feb. 1998.

[3] E. McCluskey and C.-W. Tseng. Stuck-at fault tests vs actual defects. In *Proc. of ITC*, pages 336–343, Oct. 2000.

[4] M. G. Walker. Modeling the wiring of deep submicron ICs. *IEEE Spectrum*, 37(3):65–71, Mar. 2000.

[5] A. P. Stroele. BIST pattern generators using addition and subtraction operators. *JETTA*, 11(1):69–80, Aug. 1997.

[6] J. Rajski and J. Tyszer. Test responses compaction in accumulators with rotated carry adders. *IEEE Trans. CAD*, 12(4):531–539, Apr. 1993.

[7] J. Rajski and J. Tyszer. Accumulator-based compaction of test responses. *IEEE Trans. Comp.*, 42(6):643–650, Jun. 1993.