Low Power Pattern Generation for BIST Architecture

N. Ahmed, M. H. Tehranipour, M. Nourani
Center for Integrated Circuits & Systems
The Univ. of Texas at Dallas
Richardson, TX 75083
{nxa018600,mht021000,nourani}@utdallas.edu

***Abstract*— A new low power test pattern generator using a linear feedback shift register (LFSR), called LP-TPG, is presented to reduce the average and peak power of a circuit during test. The correlation between the test patterns generated by LP-TPG is more than conventional LFSR. LP-TPG inserts intermediate patterns between the random patterns. The goal of having intermediate patterns is to reduce the transitional activities of primary inputs which eventually reduces the switching activities inside the circuit under test, and hence, power consumption. The random nature of the test patterns are kept intact. The area overhead of the additional components to the LFSR is negligible compared to the large circuit sizes. The experimental results are shown for ISCAS85 benchmarks, confirming up to 63% and 27% reduction in average and peak power, respectively.**

I. INTRODUCTION

Today's system-on-chips (SoCs) design and test confront several problems, especially power dissipation. Generally, power dissipation of a system in test mode is more than in normal mode [1]. This is because a significant correlation exists between consecutive vectors applied during the circuit's normal mode of operation, whereas this may not be necessarily true for applied test vectors in the test mode. Reduced correlation between the consecutive test vectors increases the switching activity and eventually the power dissipation in the circuit. The second reason of increasing the power dissipation during test is because the test engineers may test cores in parallel to reduce the test application time. This extra power (average or peak) can cause problems such as instantaneous power surge that causes circuit damage, difficulty in performance verification and decreased overall product yield and cost. Low power test application has become important in today's VLSI design and test.

Built-In Self-Test (BIST) has emerged as a promising solution to the VLSI testing problems. BIST is a DFT methodology aimed at detecting faulty components in a system by incorporating the test logic on chip. BIST is well known for its numerous advantages such as improved testability, at-speed testing and reduced need for automatic test equipment (ATE). In BIST, a linear feedback shift register (LFSR) generates test patterns and a multiple input shift register (MISR) compacts test responses. Test vectors applied to a circuit under test at nominal operating frequency may have more average and/or peak power dissipation than those in normal mode. The reason is that the random nature of patterns reduces the correlation between the pseudorandom patterns generated by LFSR compared to normal functional vectors. It results in more switchings and power dissipation in test mode.

Several techniques have been reported to address the low power BIST problem. The technique proposed in [1] consists of a distributed BIST control scheme that simplifies the BIST execution of complex ICs, especially during higher test activity levels. This approach can schedule the execution of every BIST element to keep the power dissipation under the specified limit. A BIST strategy called dual-speed LFSR [2] is proposed to reduce the circuit's overall switching activities. Having two different speed LFSRs, the proposed strategy applies some test patterns using low-speed LFSR by connecting to some inputs that have elevated transition densities. The low-power test pattern generator presented in [3] is based on cellular automata, reducing the test power in combinational circuits. Another low-power test pattern generator based on a modified LFSR is proposed in [4]. This scheme reduces the power in circuit under test (CUT) in general and clock tree in particular. A low power BIST for data path architecture built around multiplier-accumulator pairs is proposed in [5].

Modifying the LFSR, by adding weights to tune the pseudorandom vectors for various probabilities, decreases energy consumption and increases fault coverage [6]. A low power random pattern generation technique to reduce signal activities in the scan chain is proposed in [7]. In this technique, an LFSR generates equally probable random patterns. The technique generates a high correlation between neighboring bits in the scan chain, reducing the number of transitions and, thus, the average power. The authors in [8] proposed a method to select an LFSR's seed to reduce the lowest energy consumption using a simulated-annealing algorithm. Test vector inhibiting techniques [9] [10] [11] filter out some non-detecting subsequences of a pseudorandom test set generated by an LFSR. These architectures apply the minimum number of test vectors required to attain the desired fault coverage.

This paper presents a new test pattern generator for low power BIST. The proposed technique increases the correlation between test patterns. The original patterns are generated by an LFSR and our proposed technique generates and inserts intermediate patterns between each pair patterns to reduce the primary inputs (PIs) activities which eventually reduces the switching activities inside the circuit under test, and hence power consumption. Adding intermediate test patterns does not prolong the overall test length for a target fault coverage, and hence the test application time is still the same. We embed our proposed technique into an LFSR to create our LP-TPG. We will show that both the average and peak powers are significantly reduced in LP-TPG compared to a conventional LFSR.

The rest of the paper is organized as follows. Section II describes the proposed technique for low power test pattern generation. In Section III, we propose our new low power LFSR (LP-TPG). The experimental results are discussed in Section IV. Finally, the concluding remarks are in Section V.

II. Low Power Test Generation

The basic idea behind low power BIST is to reduce the PI activities. In this section, we propose a new test pattern generation technique which generates three intermediate test patterns between each two consecutive random patterns generated by a conventional LFSR. The proposed test pattern generation method does not decrease the random nature of the test patterns. Our technique reduces the PI's activities and eventually switching activities in the circuit under test. Let us assume that T^i and T^{i+1} are two consecutive test patterns generated by a pseudorandom pattern generator (e.g. a conventional LFSR). Suppose the two vectors are T^i=$\{t_1^i, t_2^i, \ldots, t_n^i\}$ and T^{i+1}=$\{t_1^{i+1}, t_2^{i+1}, \ldots, t_n^{i+1}\}$, where n is the number of bits in the test patterns which is equal to the number of PIs in the circuit under test. Assume that T^{k1}, T^{k2} and T^{k3} are the intermediate patterns between T^i and T^{i+1}. T^{k2} is generated as

$$T^{k2} = \{t_1^i, \ldots, t_{\frac{n}{2}}^i, t_{\frac{n}{2}+1}^{i+1}, \ldots, t_n^{i+1}\}$$

As seen, T^{k2} is generated using one half of each of the two random patterns T^i and T^{i+1}. T^{k2} is also a random pattern because it is generated using two random patterns. The other two intermediate test patterns are generated using T^{k2}. T^{k1} is generated between T^i and T^{k2} and T^{k3} is generated between T^{k2} and T^{i+1}. For example, T^{k1} is obtained by:

$$t_j^{k1} = \begin{cases} t_j^i & \text{if } t_j^i = t_j^{k2} \\ R & \text{if } t_j^i \neq t_j^{k2} \end{cases}$$

where $j \in \{1,2,...,n\}$ and R is a random bit. This method of generating T^{k1} and T^{k3} is called *R-Injection*. As seen, if two corresponding bits in T^i and T^{k2} (t_j^i and t_j^{k2}) are the same, the same bit is positioned in the corresponding bit of T^{k1} (t_j^{k1}), otherwise a random bit (R) is positioned. R can come from an output of the random generator. In this method, sum of PI's activities between T^i and T^{k1} ($N_{trans}^{i,k1}$), T^{k1} and T^{k2} ($N_{trans}^{k1,k2}$), T^{k2} and T^{k3} ($N_{trans}^{k2,k3}$) and T^{k3} and T^{i+1} ($N_{trans}^{k3,(i+1)}$) are equal to the activities between T^i and T^{i+1} ($N_{trans}^{i,(i+1)}$) or briefly:

$$N_{trans}^{i,k1} + N_{trans}^{k1,k2} + N_{trans}^{k2,k3} + N_{trans}^{k3,(i+1)} = N_{trans}^{i,(i+1)}$$

$$\sum_{j=1}^{n} |t_j^i - t_j^{k1}| + \sum_{j=1}^{n} |t_j^{k1} - t_j^{k2}| + \sum_{j=1}^{n} |t_j^{k2} - t_j^{k3}| + \sum_{j=1}^{n} |t_j^{k3} - t_j^{i+1}| = \sum_{j=1}^{n} |t_j^i - t_j^{i+1}|$$

Figure 1 shows an example of generation of intermediate test patterns between T^i and T^{i+1} assuming R=0. As shown, first and second halves of T^{k2} are equal to T^i and T^{i+1}, respectively. T^{k1} and T^{k2} are generated using R-Injection (R=0 injected in the shaded bits of T^{k1} and T^{k2}). As shown, $\sum_{j=1}^{n} |t_j^i - t_j^{i+1}|$=10 while $\sum_{j=1}^{n} |t_j^i - t_j^{k1}|$=2, $\sum_{j=1}^{n} |t_j^{k1} - t_j^{k2}|$=1, $\sum_{j=1}^{n} |t_j^{k2} - t_j^{k3}|$=4 and $\sum_{j=1}^{n} |t_j^{k3} - t_j^{i+1}|$=3, respectively. This reduction of PI's activities, reduces the switching activity inside the circuit and eventually power consumption. Note that having three intermediate

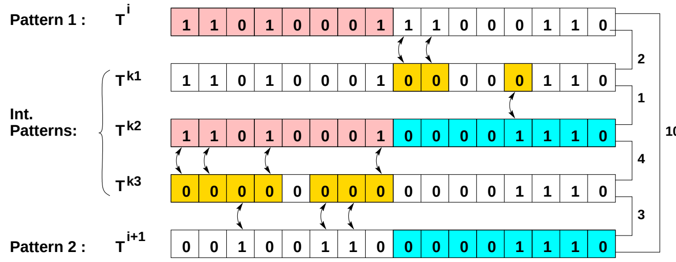


Figure 1. An example for intermediate pattern generation.

patterns between each two consecutive patterns may seem to prolong the test session by a factor of 3. However, empirically many of the intermediate patterns can do as good as conventional LFSR patterns in terms of fault detection. In fact in Section IV we show that for some benchmarks the overall number of test patterns to hit a fault coverage target may be even slightly reduced.

III. LP-TPG

We design the proposed technique into an LFSR architecture to create LP-TPG. Figure 2 shows LP-TPG with added circuitry to generate intermediate test patterns. The LFSR used in LP-TPG is an external-XOR LFSR. As shown, R-injection circuit taps the present state (T^i pattern) and the next state (T^{i+1} pattern) of the LFSR. As shown, R-injection circuit is included one AND, one OR and one 2×1 MUX. When t_j^i and t_j^{i+1} are equal, both AND and OR gates generate the same bit and regardless of R, that bit is transferred to the MUX output. When they are not equal, random bit R is sent to the output.

The LP-TPG is activated by two non-overlapping enable signals (en_1 and en_2). Each enable signal activates one half of the LFSR. In other words, when en_1en_2=10, first half of the LFSR is active and the second half is in idle mode. The second half is active when en_1en_2=01. The shaded flip flop between $n/2$th and $n/2+1$th flip flops is used to store the $n/2$th bit of the LFSR when en_1en_2=10 and that bit is used for the second half when en_1en_2=01. MUX selects either the injection bit or the exact bit in the LFSR. One small finite state machine (FSM) controls the pattern generation process as follows:

Step 1: en_1en_2=10, sel_1sel_2=11. The first half of the LFSR is active and the second half is in idle mode. Selecting sel_1sel_2=11, both halves of the LFSR are sent to the outputs (O_1 to O_n). In this case, T^i is generated.

Step 2: en_1en_2=00, sel_1sel_2=10. Both halves of the LFSR are in the idle mode. The first half of the LFSR is sent to the outputs (O_1 to $O_{n/2}$), but the injector circuit outputs are sent to the outputs ($O_{\frac{n}{2}+1}$ to O_n). T^{k1} is generated.

Step 3: en_1en_2=01, sel_1sel_2=11. The second half of the LFSR is active and the first half of the LFSR is in idle mode. Both halves are transferred to the outputs (O_1 to O_n) and T^{k2} is generated.

Step 4: en_1en_2=00, sel_1sel_2=01. Both halves of the LFSR are in the idle mode. From the first half the injector outputs are sent to the outputs of LP-TPG (O_1 to $O_{n/2}$) and the second half sends the exact bits in the LFSR to the outputs ($O_{\frac{n}{2}+1}$ to O_n) to generate T^{k3}.

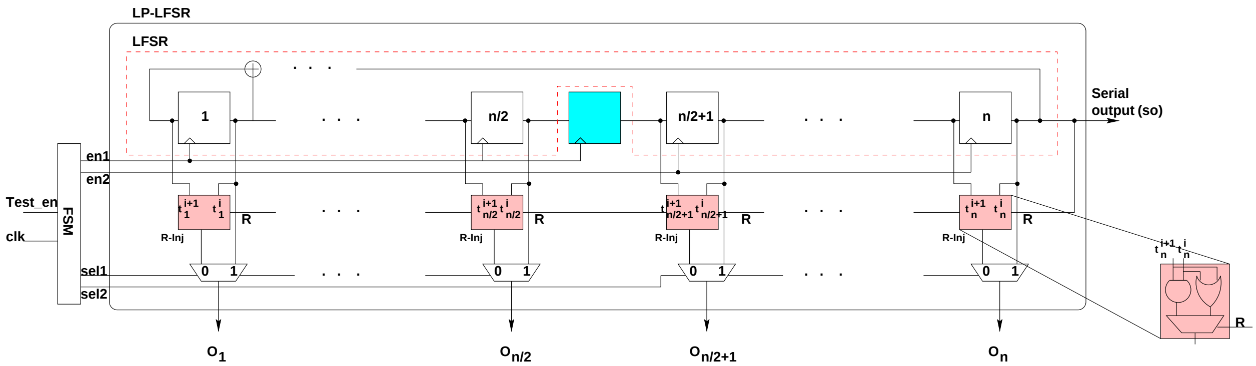
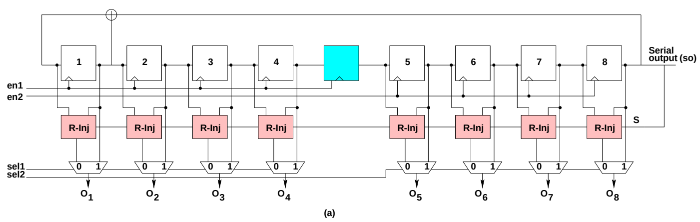


Figure 2. Our proposed low power LFSR (LP-TPG).



# clk	pattern	en1en2	sel1sel2	LP-LFSR	
1	T^1	1 0	1 1	1010	1011
2	T^{k1}	0 0	1 0	1010	1111 (R=1)
3	T^{k2}	0 1	1 1	1010	0101
4	T^{k3}	0 0	0 1	1111 (R=1)	0101
5	T^2	1 0	1 1	0101	0101
…	…	…	…	…	…

(b)

Figure 3. An example of LP-TPG using a 8-bit LFSR.

Step 5: The process continues by going through Step 1 to generate T^{i+1}.

The LP-TPG with R-injection keeps the random nature of the test patterns intact. The FSM controls the test pattern generation through steps 1 to 4 and it is independent of the LFSR size and polynomial. *clk* and *test_en* are the inputs of the FSM. When *test_en*=1, FSM starts with step 1 by setting en_1en_2=10 and sel_1sel_2=11. It continues the process by going through step 1 to step 4. One pattern is generated in each clock cycle. The size of the FSM is very small and fixed as reported in Section IV.

Figure 3(a) shows an example of the LP-TPG using an 8-bit LFSR with polynomial x^8+x+1 and seed=01001011. Figure 3(b) shows the output of the LP-TPG for R=1. As shown, between two consecutive patterns T^1 and T^2, three intermediate patterns are generated. As shown, $N^{1,2}_{trans}$=7, but $N^{1,k1}_{trans}$=1, $N^{k1,k2}_{trans}$=2, $N^{k2,k3}_{trans}$=2 and $N^{k3,2}_{trans}$=2, respectively. As clearly seen, the number of transitions between the test patterns are significantly reduced, which eventually reduces the overall peak and average powers during test.

IV. Experimental Results

A. Power Analysis and Simulation Setup

In this section, we present the experimental results of the proposed LP-TPG on the ISCAS benchmarks. The power estimation results are shown based on weighted switching activity (WSA) [14]. WSA for zero delay between test patterns T^i and T^{i+1} is given by:

$$WSA_{\{T^i,T^{i+1}\}} = \sum_{p=1}^{N_{gate}} F_p \cdot (g_{pi} \oplus g_{p(i+1)})$$

where F_p is the number of fanouts of gate p, g_{pi} and $g_{p(i+1)}$ are the output of a gate p after applying T^i and T^{i+1}, respectively. Increase or decrease in switching activities (proportional to power consumption) will be shown by ΔWSA_{avg} and ΔWSA_{peak}. They are computed with respect to random case, i.e. $|\Delta WSA_{avg}| = |\frac{WSA_{avg}(Random)-WSA_{avg}(LP-TPG)}{WSA_{avg}(Random)}|$.

In our experiments we design an LFSR with polynomial x^n+x+1. Table I shows the specifications of the ISCAS benchmarks, i.e., the number of PI's, PO's and fault coverage (FC) from random patterns. The test patterns are generated using an LFSR written in C++ program. The number of test patterns to generate the fault coverage is obtained by TetraMax [13]. It stops when it generates the desired FC and the output is the number of required test patterns for that FC. After determining the number of required test patterns, the test data set is applied to the C++ program to obtain average and peak WSA. For this purpose, we use the benchmarks' gate level description in Verilog. This process is performed for all two test data sets, conventional LFSR and LP-TPG.

FSM can be a part of BIST controller used in the circuit to control the test process. The size of the FSM is fixed and 46 equivalent gates based on the implementation using Synopsys Design Analyzer [13]. For an 8-bit LFSR with polynomial x^8+x+1, the LFSR and LP-TPG sizes are 61 and 137 equivalent gates, respectively. The LP-TPG costs more than LFSR, but as of the test area overhead is negligible compared to the size of the a large core under test.

TABLE I
ISCAS benchmarks specifications.

Circuit	*PI*	*PO*	*FC%*
c432	36	7	98
c499	41	32	98
c880	60	26	97
c1355	41	32	98
c2670	233	140	84
c3540	50	22	94
c5315	178	123	99

TABLE II

EXPERIMENTAL RESULTS FOR TWO DIFFERENT TEST GENERATION METHODS USING ZERO DELAY.

Circuit	*LFSR*			*LP-TPG*		
	N_p	WSA_{avg}	WSA_{peak}	N_p	WSA_{avg}	WSA_{peak}
c432	330	94	153	365	36	134
c499	454	128	203	450	97	174
c880	445	167	288	436	89	268
c1355	899	305	421	889	177	305
c2670	2289	761	1107	2368	324	916
c3540	950	889	1278	1308	390	1073
c5315	725	1646	2196	962	603	1633

TABLE III

CHANGES IN TEST LENGTH, AVERAGE AND PEAK WSA (COMPARED TO LFSR) FOR ZERO DELAY.

Circuit	*LP-TPG*		
	ΔN_p%	ΔWSA_{avg}%	ΔWSA_{peak}%
c432	-10.6	61.7	12.4
c499	1.0	24.2	14.2
c880	2.1	46.7	6.9
c1355	1.1	41.9	27.5
c2670	-3.4	57.4	17.2
c3540	-16.6	56.1	16.0
c5315	-34.5	63.3	24.4
Avg.	-8.7	50.2	16.9

B. Simulation Results

Table II shows the experimental results for the benchmarks considering zero delay for the gates. For fair evaluation of performance in terms of fault detection, the number of patterns N_p is obtained to achieve the same fault coverage (FC) for each of the two cases of patterns (conventional LFSR and LP-TPG). For example, to achieve fault coverage of 98% for c432 benchmark, 330 and 365 patterns are required for LFSR and LP-TPG, respectively. For LP-TPG, N_p is either less or slightly more compared to the LFSR. WSA_{avg} and WSA_{peak} for LP-TPG method is significantly less than LFSR.

Table III shows the change, compared to the LFSRcase, in test length (ΔN_g), average WSA (ΔWSA_{avg}) and peak WSA (ΔWSA_{peak}) for LP-TPG technique for zero delay. The last row shows the average reduction of each column in the table. LP-TPG shows higher number of test patterns for a target fault coverage compared to LFSR, but significantly reduction in average and peak WSA.

The instantaneous power (between two consecutive patterns) can put a lot of stress on circuits and thus is a matter of concern. Our LP-TPG significantly lowers the chance of instantaneous power violations. As an example, we measured the instantaneous WSA for each pair of test vectors generated using conventional LFSR and LP-TPG for c432 benchmark. The parameter WSA_{thr} represents the instantaneous switching activity limit defined by circuit designer. For this particular benchmark WSA_{thr}=120 (as reported in [12]). The test patterns generated by LP-TPG cross this limit much less frequently than LFSR patterns. That is 1% (4 out of 365 patterns) of LP-TPG patterns and 12% (43 out of 365 patterns) of LFSR patterns violate the limit. The more violation of such a limit the more chance to damage the circuit.

V. CONCLUSION

This paper presented a new low-power LFSR to reduce the average and peak power of a circuit during the test mode. By increasing the correlation between the test pattern, the switching activity in the circuit under test and eventually the power consumption is reduced. Additional intermediate test patterns inserted between the original random patterns reduces the PIs activity, average and peak power, but do not affect on fault coverage. The experimental results indicate up to 63% and 27% reduction in average and peak power, respectively, while it achieves the same fault coverage. LP-TPG also reduces the instantaneous power violation compared to conventional LFSR.

ACKNOWLEDGEMENTS

This work was supported in part by the National Science Foundation CAREER Award #CCR-0130513.

REFERENCES

[1] Y. Zorian, "A Distributed BIST Control Scheme for Complex VLSI Devices," in Proc. *VLSI Test Symp. (VTS'93)*, pp. 4-9, 1993.

[2] S. Wang and S. Gupta, "DS-LFSR: A New BIST TPG for Low Heat Dissipation," in Proc. *Int. Test Conf. (ITC'97)*, pp. 848-857, 1997.

[3] F. Corno, M. Rebaudengo, M. Reorda, G. Squillero and M. Violante, "Low Power BIST via Non-Linear Hybrid Cellular Automata," in Proc. *VLSI Test Symp. (VTS'00)*, pp. 29-34, 2000.

[4] P. Girard, L. Guiller, C. Landrault, S. Pravossoudovitch, H. -J. Wunderlich, "A modifi ed Clock Scheme for a Low Power BIST Test Pattern Generator," in Proc. *VLSI Test Symp. (VTS'01)*, pp. 306-311, 2001.

[5] D. Gizopoulos et. al., "Low Power/Energy BIST Scheme for Datapaths," in Proc. *VLSI Test Symp. (VTS'00)*, pp. 23-28, 2000.

[6] X. Zhang, K. Roy and S. Bhawmik, "POWERTEST: A Tool for Energy Conscious Weighted Random Pattern Testing," in Proc. *Int. Conf. VLI Design*, pp. 416-422, 1999.

[7] S. Wang and S. Gupta, "LT-RTPG: A New Test-Per-Scan BIST TPG for Low Heat Dissipation," in Proc. *Int. Test Conf. (ITC'99)*, pp. 85-94, 1999.

[8] P. Girard et. al., "Low Energy BIST Design: Impact of the LFSR TPG Parameters on the Weighted Switching Activity," in Proc. *Int. Symp. on Circuits and Systems (ISCAS'99)*, pp. , 1999.

[9] P. Girard, et. al., "A Test Vector Inhibiting Technique for Low Energy BIST Dsign," in Proc. *VLSI Test Symp. (VTS'99)*, pp. 407-412, 1999.

[10] S. Manich, et. al., "Low Power BIST by fi ltering Non-Detecting Vectors," in Proc. *European Test Workshop (ETW'99)*, pp. 165-170, 1999.

[11] F. Corno, M. Rebaudengo, M. Sonza Reorda and M. Violante, "A New BIST Architecture for Low Power Circuits," in Proc. *European Test Workshop (ETW'99)*, pp. 160-164, 1999.

[12] X. Zhang and K. Roy, "Peak Power Reduction in Low Power BIST," in Proc. *Int. Symp. on Quality Elect. Design (ISQED'00)*, pp. 425-432, 2000.

[13] Synopsys Inc., "User Manuals for SYNOPSYS Toolset Version 2002.05," Synopsys, Inc., 2002.

[14] S. Manich and J. Figueras, "Sensitivity of the Worst Case Dynamic Power Estimation on Delay and Filtering Models," in Proc. *PATMOS Workshop*, 1997.