Design and Synthesis of Low Power Weighted Random Pattern Generator Considering Peak Power Reduction

Xiaodong Zhang and Kaushik Roy
Electrical & Computer Engineering
Purdue University, West Lafayette, IN 47907

Abstract

In order to meet the power and reliability constraints, it is important to reduce average power and peak power during test. In this paper we propose a Low Power Automatic Test Pattern Generator (LPATPG), which can be used during on-line testing of large circuits requiring low power dissipation. The LPATPG can be implemented by linear cellular automata (CA) with appropriate external weighting logic. While the average power is reduced by finding the optimal signal activities (probabilities of signal switching) at the primary inputs, the peak power is reduced by finding the best initial conditions in the CA cells. Results on ISCAS benchmark circuits show that average power reduction of up to 79.7% , peak power reduction of up to 39.2% and energy reduction of up to 84.4% can be achieved (compared to linear cellular automata) while achieving high fault coverage.

Keywords

BIST Synthesis, Low Power, Testing, Peak Power, Weighted Random Pattern Generator, Cellular Automata, Low Power BIST.

I. Introduction

As part of our present day life-styles, the portable consumer electronic products, ranging from simple wristwatches to complex laptop computers, have brought the issues of low power VLSI circuit design and testing to the forefront. The prohibitive costs for extra cooling devices, fans and expensive packaging systems impose a strong limit on the average power dissipation. In order to meet the power and reliability constraints, it is very important to reduce the average power dissipation during test [1].

Besides the average power, the peak power consumption is another critical issue during the circuit design and test, since it determines the thermal and electrical limits of components, the system packaging requirements, and heat sinks dimensions [10]. When dealing with high-density systems such as modern ASICs and MCMs, in order to perform a non-destructive test, we have to satisfy all the power constraints defined in the design phase. Therefore, the peak power consumption during test must be kept under a well defined threshold. Excessive peak power during test can cause several problems. First, this leads to an increased maximum current flow in the circuit under test (CUT) which in turn requires a more expensive package as a consequence. Moreover, with increased peak currents, electromigration becomes likely thus affecting the reliability of the system.

The average power during random test will decrease, if we reduce the signal activities (probability of switching) at the primary inputs. In this paper, we propose a Low Power BIST technique to minimize the test length and the average power during random testing, with high fault coverage. The conventional weighted random pattern testing technique (WRP) only reduces the test length [3] by optimizing signal probablities (probability that a signal is logic ONE) or weights of the inputs. Conversely, in Low Power BIST, if we apply a random sequence conforming to the optimal probabilities and activities to the CUT (circuit under test), then both the average power and the test length will be reduced significantly [1]. Because BIST is generally used during regular operation of a circuit or during power on, Low Power BIST can be used efficiently in on-line testing for battery operated applications. Low-power BIST is a concern for companies like Lucent Technologies [11]. Since generation of optimal probabilities or activities can be area expensive, we restrict our

This research is supported in part by DARPA and by Lucent Technologies

search for the optimal weights to a small number of probabilities and activities. The corresponding low power ATP (Automatic Test Pattern) generator can be designed and implemented by linear cellular automata (CA) with appropriate external weighting logic.

While average power can be reduced by optimizing the signal activities, the peak power cannot be reduced in such a fashion, since it is not directly related to the average signal activities. However, peak power can be reduced during the design phase of the low power ATP generator. It can be noted that the required test length is usually negligibly small compared to the maximum cycle length of a CA. During random test, only a small segment of the maximum cycle length is applied to the CUT, hence, it is possible to find an optimal segment which excludes those vectors with high instantaneous power. The optimal initial conditions of the CA cells will lead to the optimal vector segment within the maximum cycle length.

The rest of the paper is organized as follows: Section II introduces the formal definitions of signal probabilities and signal activities. Section III shows how to design a low power ATP generator. Section IV describes a genetic algorithm based optimization technique to generate weight sets and initial conditions optimized for high testability, low average power and low peak power. Experimental results for a number of benchmark circuits are presented in Section V. Finally, the conclusions are given in Section VI.

II. Preliminaries and Definitions

Definition 1 (**Signal Probability**) Signal probability $P(k)$ of a node k of a circuit is the probability that node k is logical ONE.

In this paper, we assume the probability $P(x(t))$ of signal $x(t)$ does not change with time, and is denoted as $P(x)$.

Definition 2 (**Primary Input Probability Set**) For a circuit with n primary inputs, a probability set S_P assigned to its inputs is a vector $(P(x_1), P(x_2), ..., P(x_n))$, where $P(x_i) \in [0,1]$ is the probability that input x_i is a logic ONE.

Definition 3 (**Signal Activity**) The *activity* $A(x)$ of a signal $x(t)$ is defined as $\lim_{\tau\to\infty} \frac{m_x(\tau)}{\tau}$, equals to the expected value of $\frac{m_x(\tau)}{\tau}$. The variable m_x is the number of switching of $x(t)$ in the time interval $(-\tau/2, \tau/2]$. The normalized activity $a(x)$ is defined as $A(x)$ divided by clock frequency f, and is the probability of the signal to switch within a clock period.

We assume the normalized activity $a(x(t))$ does not change with time, and is denoted as $a(x)$ in this paper. $a(x)$ is given as follows:

$$a(x) = P(\{x(t-T)\bar{x}(t)\} \cup \{\bar{x}(t-T)x(t)\})$$

where $x(t-T)\bar{x}(t)$ denotes a switching transition from 1 to 0, while $\bar{x}(t-T)x(t)$ denotes a switching transition from 0 to 1 (T is the clock period). We assume, without loss of generality, that the signal switches at the rising edge of the clock cycle. Since the two events $\{x(t-T)\bar{x}(t)\}$ and $\{\bar{x}(t-T)x(t)\}$ are mutually exclusive, we also have:

$$a(x) = P(x(t-T)\bar{x}(t)) + P(\bar{x}(t-T)x(t)) \quad (1)$$

Definition 4 (**Activity Set**) For a circuit with n primary inputs, an activity set S_A assigned to its inputs is a vector $(a(x_1), a(x_2), ..., a(x_n))$, were $a(x_i)$ is the activity of i-th primary input x_i.

III. Design of Low Power ATP Generator

A. Optimal Probability-Activity Set

In order to reduce the average power during test, we have to optimize the probability-activity set. The probability-activity set is optimal if both the average power and the test length are minimum for a certain fault coverage.

After we find the optimal Probability-Activity set, we need to design a circuit which will generate test vectors that conforms to the optimal probabilities and activities. However, the realization of signal activities is not straight-forward, hence we need to transform the signal activities into some conditional probabilities, which can be easily implemented by linear cellular automata with external weighting logic.

B. Using Conditional Probabilities to Realize Probability-Activity set

Now let us consider how to transform signal probability and activity into some conditional probabilities. Let us denote the optimal probability and activity at a certain primary input x as $P(x)$ and $a(x)$. The realization of $P(x)$ and $a(x)$ can be considered in two steps: first, we generate the random sequence according to some conditional probabilities; then, we derives the relationship between the conditional probabilities and $P(x)$, $a(x)$.

Let us denote random pattern applied to the primary input x as $V_0, V_1, V_2, V_3, ...$ The first bit V_0 can be randomly generated. The subsequent bits $(V_1, V_2, V_3, ...)$ can be generated according to some conditional probabilities $P(x(t)|x(t-T))$ and $P(x(t)|\bar{x}(t-T))$: if $V_j (j = 0, 1, 2, 3, ...)$ is ONE, then V_{j+1} is generated according to the conditional probability $P(x(t)|x(t-T))$; on the other hand, if V_j is ZERO, then V_{j+1} is generated according to $P(x(t)|\bar{x}(t-T))$. Now let us determine the relationship between $P(x(t)|x(t-T))$, $P(x(t)|\bar{x}(t-T))$ and $P(x)$, $a(x)$.

First, let us consider the event $\{x(t-T)\}$, which can also be viewed as a union of two mutually exclusive events:

$$\{x(t-T)\} = \{x(t-T)x(t)\} \cup \{x(t-T)\bar{x}(t)\}$$

Therefore we have:

$$P(x(t-T)) = P(x(t-T)x(t)) + P(x(t-T)\bar{x}(t)) \tag{2}$$

Similarly we also have:

$$P(x(t)) = P(x(t-T)x(t)) + P(\bar{x}(t-T)x(t)). \tag{3}$$

However, since $P(x(t))$ is invariant with respect to time,

$$P(x(t-T)) = P(x(t)) = P(x), \tag{4}$$

Therefore, equating equations 2 and 3 we obtain

$$P(x(t-T)\bar{x}(t)) = P(\bar{x}(t-T)x(t)) \tag{5}$$

From equation(1), we have:

$$a(x) = P(x(t-T)\bar{x}(t)) + P(\bar{x}(t-T)x(t)) \tag{6}$$

Hence,

$$P(x(t-T)\bar{x}(t)) = P(\bar{x}(t-T)x(t)) = a(x)/2 \tag{7}$$

From equation(7), we can obtain the conditional probabilities $P(x(t)|x(t-T))$ and $P(x(t)|\bar{x}(t-T))$ which yield the desired probability-activity set. They are given as follows:

$$P(x(t)|x(t-T)) = [P(x) - a(x)/2]/P(x), \text{ and} \tag{8}$$

$$P(x(t)|\bar{x}(t-T)) = [a(x)/2]/[1 - P(x)] \tag{9}$$

In summary, the realization of $P(x)$ and $a(x)$ can be done in two steps. First, we determine the corresponding $P(x(t)|x(t-T))$ and $P(x(t)|\bar{x}(t-T))$, then we generate the random sequence according to the conditional probabilities.

C. Implementation of Conditional Probabilities by Cellular Automata with External Weighting Logic

Now let us consider how to design a circuit that realizes the optimal conditional probabilities. The structure of the circuit is shown in Fig. 1. First of all, we use one-dimensional linear cellular automata (with CA rules as described in [8]) to generate two equiprobable random sequences. Then we use some external weighting logic to produce sequences conforming to the desired conditional probabilities. In Fig. 1, the weighting logic below the cellular array 1 are designed such that the signal probabilities of A_i conform to the conditional

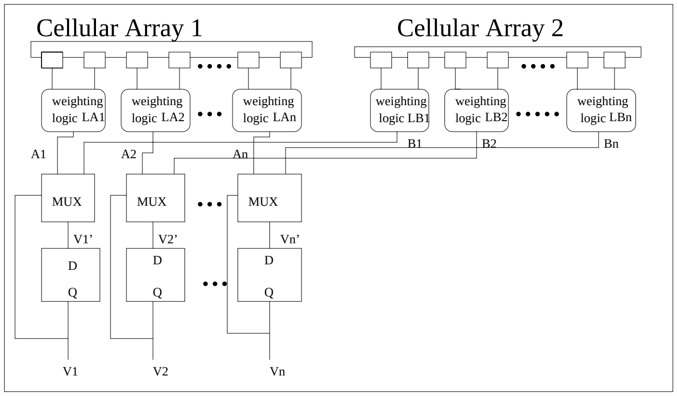

Fig. 1. Low power ATP generator with linear CA

probabilities $P(x_i(t)|x_i(t-T))(i = 1, 2, ..., n)$, and the weighting logic under the cellular array 2 are designed such that the signal probabilities of B_i conform to the conditional probability $P(x_i(t)|\bar{x}_i(t-T))$. The final random sequence that applies to the circuit under test contains $V_1, V_2, ..., V_n$, and they are output bits of some flip-flop's. The present value of V_i equals the value of $V_i'(i = 1, 2, ..., n)$ at the preceding clock cycle. At the control end of the MUX, V_i selects either A_i or B_i to produce the bit V_i' depending on the value of V_i. If V_i is logic ONE, then it chooses A_i, otherwise it chooses B_i. Hence the output sequence of MUX $(V_1', V_2', ..., V_n')$ conforms to the desired probability and activity, which is given as follows:

$$P(x) = \frac{P(x(t)|\bar{x}(t-T))}{P(x(t)|\bar{x}(t-T)) + [1 - P(x(t)|x(t-T))]} \tag{10}$$

$$a(x) = \frac{P(x(t)|\bar{x}(t-T))[1 - P(x(t)|x(t-T))]}{P(x(t)|\bar{x}(t-T)) + [1 - P(x(t)|x(t-T))]} \tag{11}$$

D. Design of Weighting Logic in Low Power ATPG

D.1 Restriction of Conditional Probabilities

The weighting logic can be expensive since the optimal conditional probabilities are arbitrary, and the required external logic may be very large for some conditional probabilities. For instance, to realize a conditional probability of $P = 0.35$ (Fig. 2), we have to use:

- 1 AND gate to yield $P1 = 0.25$;
- 3 AND gates to yield $P2 = 0.0625$;
- 4 AND gates to yield $P3 = 0.0312$;
- Finally, 2 OR gates to yield $P = P1 + P2 + P3 = 0.3438$.

In summary, we need 10 gates to realize one conditional probability. Therefore, in this paper, we will restrict the conditional probabilities to a small set of numbers. Since the signal activity will be small during low power testing, the conditional probability $P(x(t)|\bar{x}(t-T))$ will be also small, and we restrict it to a set of numbers smaller than 1/2, such as $\{1/2, 3/8, 1/4, 1/8, 1/16, 1/32\}$. On the other hand, the conditional probability $P(x(t)|x(t-T))$ will be large if the signal activity is small, hence we restrict it to a set of numbers larger than 1/2, such as $\{1/2, 5/8, 3/4, 7/8, 15/16, 31/32\}$. In this approach, every conditional probability can be implemented by 1 to 4 gates (Fig. 3), and the implementation of all the 6 conditional probabilities only require 6 logic gates, making it area and power efficient.

D.2 Weighting Logic Block in Low Power ATPG

The weighting logic block is shown in Fig. 3. The upper weighting logic block LA_i in Fig. 3 realizes the conditional probabilities $P(x_i(t)|\bar{x}_i(t-T))$. The output bits of the upper weighting logic block are $C_j(j = 1, 2, 3, 4, 5, 6)$. C_1 is just an output of a CA cell, hence $P(C_1) = 1/2$. C_2 is generated in two stages: first, an intermediate bit with probability $P = 3/4$ is generated by an OR gate, next, by applying the intermediate

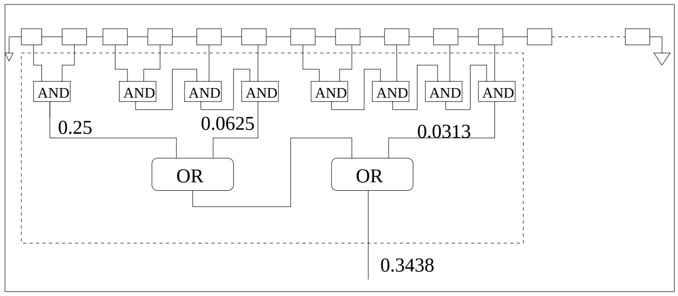

Fig. 2. Weighting logic to realize P=0.35

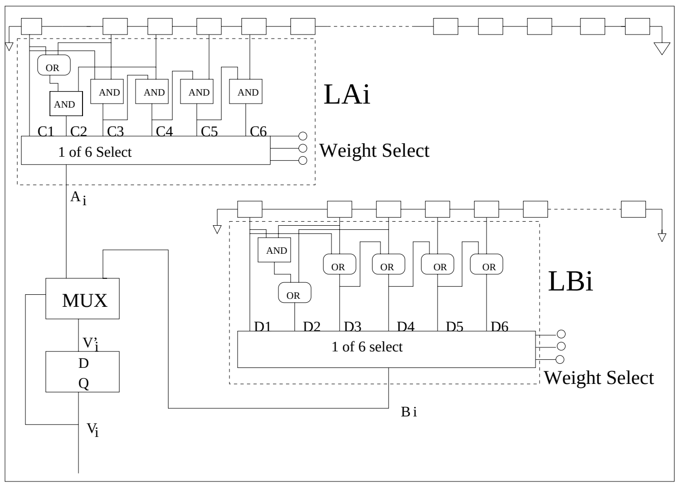

Fig. 3. Programmable low power ATP generator with linear CA

bit and another equiprobable bit to an AND gate, C_2 is produced. Hence $P(C_2) = (3/4)\ (1/2) = 3/8$. The other four output bits $C_j's(j = 3,4,5,6)$ are generated by one to four AND gates, respectively. In summary, $C_j's(j = 1,2,3,4,5,6)$ realize the conditional probabilities $\{1/2, 3/8, 1/4, 1/8, 1/16, 1/32\}$. For a certain circuit block under test, we can determine the best conditional probability $P(x_i(t)|\bar{x}_i(t-T))$ among $\{1/2, 3/8, 1/4, 1/8, 1/16, 1/32\}$, and use weight select to choose the proper C_j, producing the output bit A_i which conforms to optimal $P(x_i(t)|\bar{x}_i(t-T))$. The lower weighting logic block LB_i in Fig. 3 implements the conditional probability of $P(x_i(t)|x_i(t-T))$. The output bits of the lower weighting block are $D_j(j = 1,2,3,4,5,6)$. D_1 is just an output of a CA cell, hence $P(D_1) = 1/2$. D_2 is generated in two stages: first, an intermediate bit with probability $P = 1/4$ is generated by an AND gate, next, by applying the intermediate bit with another equiprobable bit to an OR gate, D_2 is produced. Hence $P(D_2) = 1-(1-1/4)\ (1-1/2) = 5/8$. The other four output bits $D_j's(j = 3,4,5,6)$ are generated with one to four OR gates, respectively. In summary, $D_j's(j = 1,2,3,4,5,6)$ realize the conditional probabilities $\{1/2, 5/8, 3/4, 7/8, 15/16, 31/32\}$. Once we determine the optimal $P(x_i(t)|x_i(t-T))$ among $\{1/2, 5/8, 3/4, 7/8, 15/16, 31/32\}$, we can choose some D_j to produce the correct output bit B_i with appropriate weight select signal.

IV. Synthesis of Low Power Automatic Test Pattern Generator

We have already shown the basic circuit structure of a Low Power Automatic Test Pattern Generator (LPATPG). The synthesis of LPATPG for a circuit under test requires the conditional probabilities (or the weight select signals) to be optimized for minimum average power, and the initial conditions of the CA cells to be optimized for minimum peak power.

A. Optimization of Weights for Low Average Power

The search for the optimal weights (conditional probabilities) requires estimation of fault coverage and test length, and accurate estimation of average power dissipation.

TABLE I

COMPARISON BETWEEN LINEAR CA AND LOW POWER CA

Cir- -cuit	# of faults detected /(total#)	Linear CA			Low Power CA					
		length0	power0	peak0	length	power	peak	energy reduct	power reduct	peak reduct
C432	460/(512)	765	110.9	192	508	25.9	168	84.4%	76.6%	12.5%
C499	1215/(1262)	605	281.8	370	588	69.2	225	76.11%	75.42%	39.2%
C880	880/(912)	739	156.7	263	554	38.27	211	81.7%	75.6%	20%
C1908	1375/(1575)	615	347	500	583	70.5	361	80.7%	79.7%	27.8%
C1355	1529/(1606)	612	293.2	380	568	75.5	252	76.1%	74.3%	22.7%
C2670	1900/(2280)	648	481.6	676	601	117.5	519	77.4%	75.6%	23.2%
C3540	2730/(2959)	830	673.2	896	592	175.4	658	81.4%	74.0%	26.6%
C5315	4724/(4826)	657	1228.1	657	574	538.2	574	61.7%	56.2%	12.6%
C6288	7616/(7618)	119	1711	2111	98	1150	1840	44.61%	32.7%	12.8%
C7552	6187/(6744)	1175	1931.1	2459	567	846.7	2264	78.84%	56.2%	8%
AVG								74%	68%	21%

It is expensive to use fault simulation during optimization to estimate fault coverage and test length. In this paper, we use an approach based on single path sensitization [3]. A test pattern sensitizes a single path from a node x to a primary output o, if there is exactly one path from x to o, in which the logical value at each node depends on the value at x. Once node x is single path sensitized, the stuck-at-0 fault at x can be detected if x is at logic ONE, and stuck-at-1 fault at x can be detected if x is at logic ZERO.

A.1 Exact Calculation of Energy and Power of Cellular Automata

After we estimate the minimum test length (denoted as $length_{test}$) to achieve a certain fault coverage by single path sensitization, we can determine the number of switching at every node of the circuit during the test mode by applying the weighted random sequences to the circuit under test.

The energy consumed per switching at a node k is $\frac{1}{2}C(k)V_{DD}^2$, where $C(k)$ is the output capacitance at node k, and V_{DD} is the power supply voltage. $C(k)$ is approximately proportional to the fanout at node k.

Let us denote the test length to achieve a certain fault coverage as $length_{test}$, and denote the clock period as T. If the signal at node k switches $m(k)$ times during test, then the energy consumed at node k during the test is $\frac{1}{2}m(k)fanout(k)C_{avg}(k)V_{DD}^2$. Hence the energy consumed by the CUT during test is given as follows:

$$Energy = \sum_{k \in allnodes} \frac{1}{2} m(k) fanout(k) C_{avg}(k) V_{DD}^2 \tag{12}$$

The average power is the average energy per clock period, and is given as follows:

$$Power = \frac{Energy}{length_{test}\, T} \tag{13}$$

A.2 Optimization of Weights

We need to determine the optimal weights for high fault coverage with minimum power. Genetic Algorithm (GA) [7] is well-suited to this problem because of the complex nature of the optimization. The cost function for Genetic Algorithm is the energy dissipation. Since energy is a product of test length and average power, minimizing energy will minimize both test length and average power. We encode the conditional probabilities $P(x_i(t)|x_i(t-T))$ and $P(x_i(t)|\bar{x}_i(t-T))(i = 1, 2, ...n)$ as a binary string. Since the conditional probabilities determine the weight selects, the binary string represents the structure of LPATPG. The best binary string represents the optimal LPATPG. The population size is typically set to 100. For each binary string, we can decode the corresponding conditional probabilities, and determine the weight selects. Using single path sensitization, we can estimate the minimum test length for a certain fault coverage. Then we can calculate the energy consumed during the application of the test sequence. The energy is then assigned to the binary string as its associated fitness.

The population is first initialized with random strings. The fitness function is then calculated. The evolutionary processes include selection, 1-point crossover, 2-point crossover, uniform crossover, multi-point crossover, mutations and inversions. Mutation probability is set to 0.01, and crossover probability is set to 0.25 [7]. Since selection is biased toward more highly fit individuals, the average fitness is expected to increase from one generation to the next. The best individual is always saved since it may appear in any generation.

B. Optimization of Initial Conditions for Low Peak Power

After we determine the optimal weights (conditional probabilities) for minimum average power, we separately search the best initial conditions of the CA (cellular automata) cells for minimum peak instantaneous power during the test.

Let us denote the required test length for a certain fault coverage as L. L is usually negligible, compared to the maximum cycle length L_{max} of cellular automata or LFSR. For example, for a linear cellular automata with 36 cells, the maximum cycle length is 6.87×10^{10}, and this is much longer than a typical test sequence with several thousand vectors. If we divide the maximum cycle into $\frac{L_{max}}{L_0}$ equal segments, where each segment contains L_0 vectors, then the peak instantaneous power in different segments can be quite different. Hence it is important to search the optimal segments for minimum peak power. Let us denote the first vector of the optimal segment as V_0. If we set the initial conditions of CA cells to V_0, then during random testing, the optimal segment that follows V_0 is applied to the circuit under test, and the peak power during test is reduced, compared to the case of random initial conditions.

We search the optimal initial conditions while keeping the same conditional probabilities. Genetic algorithm (GA) is employed once again for optimization, where the parameters to be optimized are the initial conditions, and the cost function for optimization is the peak instantaneous power during the minimum test length for a certain fault coverage. The population of GA is first initialized with random strings. The cost function associated with each string is determined as follows: First, each string can be interpreted as the initial conditions of the CA cells. Next, given the initial conditions and the weighting logic (conditional probabilities), the test sequence can be calculated. Then the minimum test length can be estimated by single path sensitization. Finally, after the calculation of instantaneous power for every time instance during the minimum test length, the peak power among all the instantaneous power can be extracted as the cost function. As the genetic algorithm evolves, the initial conditions are optimized. The best binary string is always saved since it may appear in any generation.

V. Experimental Results

A tool **CASYN** to design and synthesize the LPATPG has been implemented in C under the Berkeley SIS environment[6]. **CASYN** uses genetic algorithm to determine proper weight select signals and the best initial conditions for LPATPG. It takes a circuit description file and a parameter $num_{generation}$ as inputs, where the circuit description file is of Berkeley BLIF format. The parameter $num_{generation}$ will determine how many generations the genetic algorithm will evolve. The outputs of **CASYN** are the optimal weight selects and the best initial conditions.

After using **CASYN** to obtain the optimal weight selects and best initial conditions for LPATPG, we use fault simulation to verify our results. The fault simulator is HITEC/PROOFS from UIUC, which supports gate types of AND, NAND, OR, NOR, INVERTER and BUFFER [5]. Consequently, all the benchmark circuits are technology mapped to circuits consisting of only AND, NAND, OR, NOR, INVERTER and BUFFER. The verifying procedure is as follows: First, we use the LPATPG with the optimal weight selects to generate random patterns; then we apply the random patterns to the HITEC/PROOFS fault simulator and determine the actual minimum test length for a certain fault coverage. Finally, we calculate the energy

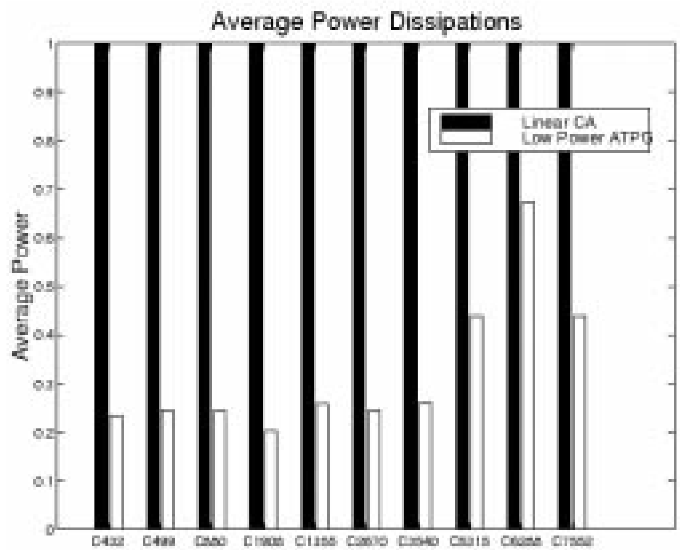

Fig. 4. Average Power Dissipation for Benchmark Circuits

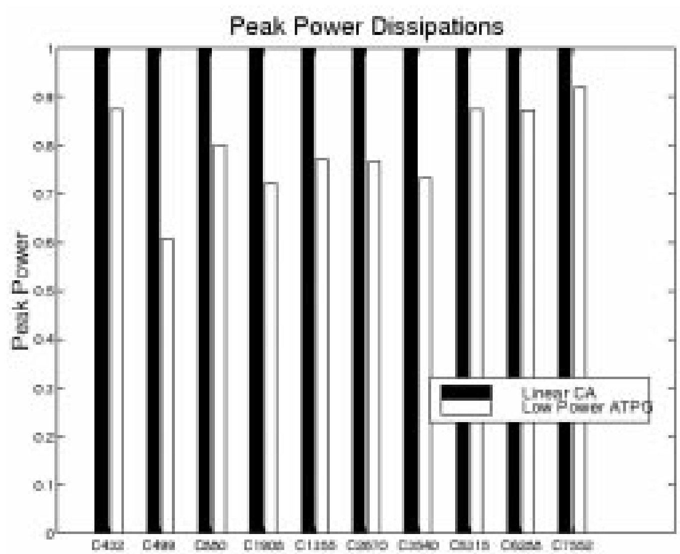

Fig. 5. Peak Power Dissipation for Benchmark Circuits

and power consumed for that test length by equation (12) and equation (13). The energy, average power and peak power are compared to the case of equiprobable random testing by linear cellular automata.

The average power reduction using LPATPG is illustrated in Fig. 4. The vertical axis of the plot is the normalized average power (all powers are normalized to the equiprobable case). For each benchmark circuit, the two bars represent the average power dissipation using linear cellular automata as equi-probable random pattern generator (left) and the power when we use the low power ATPG (right). The reductions of peak power and energy are illustrated in Fig. 5 and Fig. 6, respectively.

Finally, the summary of our experimental results are given in Table I. In Table I, "power0", "peak0" and "length0" refer to the average power, peak power and minimum test length obtained by linear cellular automata, while "power", "peak" and "length" represents the average power, peak power and test length achieved by LPATPG. The last row gives the average values of power/energy reductions over ten benchmark circuits.

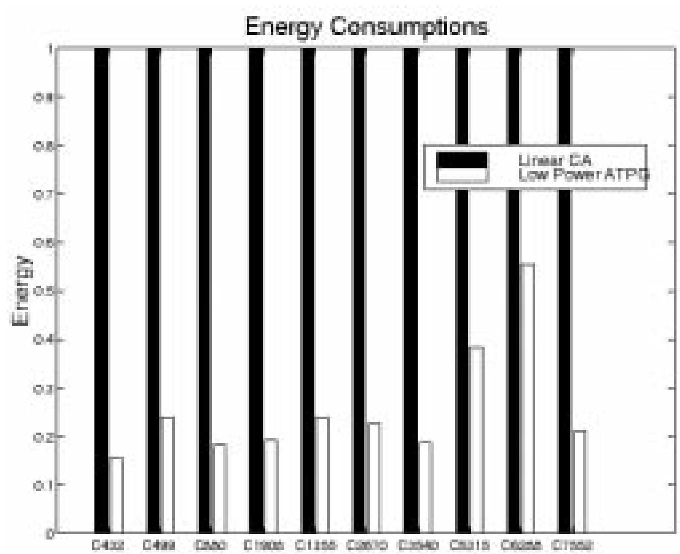

Fig. 6. Energy Consumption for Benchmark Circuits

VI. Conclusions

Low power BIST is important for battery operated systems for on-line testing strategies. In this paper we present a design and synthesis tool **CASYN** which determine the optimal structure of LPATPG for high fault coverage with minimum average power and minimum peak power. Our results on ISCAS benchmark circuits show that power reduction of up to 79.7%, peak power reduction of up to 39.2% and energy reduction of up to 84.4% can be achieved by using LPATPG, compared to linear cellular automata.

Acknowledgments

This research was supported in part by Lucent (670-1285-4016) and DARPA (F33615-95-C-1625).

References

[1] X. Zhang, K. Roy, "POWERTEST: A Tool for Energy Conscious Weighted Random Pattern Testing", *The 12th International Conference on VLSI Design.*

[2] M. A. Miranda, Carlos A. Lopez-Barrio, " Generation of Optimized Single Distributions of Weights for Random Built-In Self-Test " *International Test Conference 1993*, Dec. 1993, pp. 1023.

[3] H. Wunderlich, "Protest: A Tool for Probabilistic Testability Analysis" *22nd Design Automation Conference* PP. 204-211.

[4] J.A. Waicukauski, etc. " A Method for Generating Weighted random Test Patterns" *IBM Journal Research Develop.* Vol 33, No.2, March 1989, pp. 149-160.

[5] T. M. Niermann, W. Cheng, and J. H. Patel, "PROOFS: A fast, memory efficient sequential circuit fault simulator", *Proceedings of the ACM/IEEE Design Automa* pp. 535-540, June 1990.

[6] E.M. Sentovich, K.J. Singh, etc. "SIS: A system for Sequential Circuit Synthesis", Memorandum No. UCB/ERL M92/41.

[7] D.E. Goldberg, "Genetic Algorithms in search, Optimization, and Machine Learning, Reading", 1989 Addison-Wesley.

[8] K. Cattel, S. Zhang, "Minimal Cost One-Dimensional Linear Hybrid Cellular Automata of Degree Through 500", *Journal of Electronic Testing: Theory and Applications*, 6, 255-258 (1995).

[9] Hortensius et al., "Cellular Automata based Pseudo-random Number Generators for Built-In-Self-Test", *IEEE Trans. CAD, vol. 8, no. 8, August 1989*, pp. 842-859.

[10] M. Pedram, "Power Minimization in IC Design: Principles and Applications", *ACM Transaction on Design Automation of Electronic Systems*, Vol. 1, No. 1, 1996, pp. 3-56.

[11] Dr. Sudipta Bhawmik, Lucent Technolgies, "private communications".