

# LOW POWER TEST PATTERN GENERATION FOR BIST APPLICATIONS

Karthikkumar.B<sup>1</sup>, Anusha.T<sup>2</sup>, Dhanagopal.R<sup>3</sup>

<sup>1</sup>ME (Applied Electronics), ECE, Jayaram College of Engineering and Technology, Trichy, India

<sup>2</sup>ME (Applied Electronics), ECE, Jayaram College of Engineering and Technology, Trichy, India

<sup>3</sup>Assistant Professor, ECE, Jayaram College of Engineering and Technology, Trichy, India

## Abstract

This paper proposes a novel test pattern generator (TPG) for built-in self-test. Our method generates multiple single input change (MSIC) vectors in a pattern, i.e., each vector applied to a scan chain is an SIC vector. A reconfigurable Johnson counter and a scalable SIC counter are developed to generate a class of minimum transition sequences. The proposed TPG is flexible to both the test-per-clock and the test-per-scan schemes. A theory is also developed to represent and analyze the sequences and to extract a class of MSIC sequences. Analysis results show that the produced MSIC sequences have the favorable features of uniform distribution and low input transition density. Simulation results with ISCAS benchmarks demonstrate that MSIC can save test power and impose no more than 7.5% overhead for a scan design. It also achieves the target fault coverage without increasing the test length.

**Keywords**—Built-in self-test (BIST), low power, single-input change (SIC), test pattern generator (TPG)

\*\*\*

## 1. INTRODUCTION

BUILT-IN SELF-TEST (BIST) techniques can effectively reduce the difficulty and complexity of VLSI testing, by introducing on-chip test hardware into the circuit-under-test (CUT). In conventional BIST architectures, the linear feedback Shift register (LFSR) is commonly used in the test pattern Generators (tpgs) and output response analyzers. A major Drawback of these architectures is that the pseudorandom Patterns generated by the LFSR lead to significantly high switching activities in the CUT [1], which can cause excessive Power dissipation. They can also damage the circuit and reduce Product yield and lifetime [2], [3]. In addition, the LFSR Usually needs to generate very long pseudorandom sequences In order to achieve the target fault coverage in nanometer technology.

### 1.1 Prior Work

Several advanced BIST techniques have been studied and applied. The first class is the LFSR tuning. Girard et al. analyzed the impact of an LFSR's polynomial and seed selection on the CUT's switching activity, and proposed a method to select the LFSR seed for energy reduction [4].

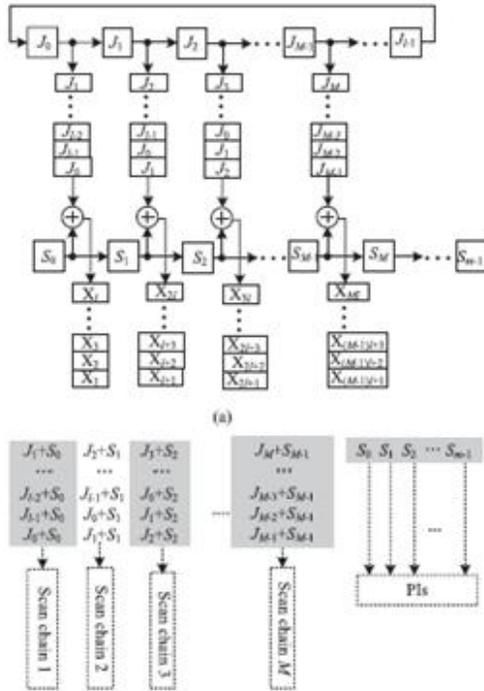
The second class is low-power TPGs. One approach is to design low-transition TPGs. Wang and Gupta used two LFSRs of different speeds to control those inputs that have elevated transition densities [5]. Corno et al. provided a low power TPG based on the cellular automata to reduce the test power in combinational circuits [6]. Another approach focuses on modifying LFSRs. The scheme in [7] reduces the power in the

CUT in general and clock tree in particular. In [8], a low power BIST for data path architecture is proposed, which is circuit dependent. However, this dependency implies that no detecting subsequences must be determined for each circuit test sequence. Bonhomme et al. [9] used a clock gating technique where two no overlapping clocks control the odd and even scan cells of the scan chain so that the shift power dissipation is reduced by a factor of two. The ring generator [10] can generate a single-input change (SIC) sequence which can effectively reduce test power. The third approach aims to reduce the dynamic power dissipation during scan shift through gating of the outputs of a portion of the scan cells. Bhunia et al. [11] inserted blocking logic into the stimulus path of the scan flip-flops to prevent the propagation of the scan ripple effect to logic gates. The need for transistors insertion, however, makes it difficult to use with standard cell libraries that do not have power-gated cells. In [12], the efficient selection of the most suitable subset of scan cells for gating along with their gating values is studied.

The third class makes use of the prevention of pseudorandom patterns that do not have new fault detecting abilities [13]–[15]. These architectures apply the minimum number of test vectors required to attain the target fault coverage and therefore reduce the power. However, these methods have high area overhead, need to be customized for the CUT, and start with a specific seed. Gerstendorfer et al. also proposed to filter out no detecting patterns using gate-based blocking logics [16], which, however, add significant delay in the signal propagation path from the scan flip-flop to logic.

Several low-power approaches have also been proposed for Scan-based BIST. The architecture in [17] modifies scan-path Structures, and lets the CUT inputs remain unchanged during a shift operation. Using multiple scan chains with many scan enable (SE) inputs to activate one scan chain at a time, the TPG proposed in [18] can reduce average power consumption during scan-based tests and the peak power in the CUT. In [19], a pseudorandom BIST scheme was proposed to reduce switching activities in scan chains. Other approaches include LT-LFSR [20], a low-transition random TPG [21], and the weighted LFSR [22]. The TPG in [20] can reduce the transitions in the scan inputs by assigning the same value to most neighboring bits in the scan chain. In [21], power reduction is achieved by increasing the correlation between consecutive test patterns. The weighted LFSR in [22] decreases energy consumption and increases fault coverage by adding weights to tune the pseudorandom vectors for various probabilities.

**2. PROPOSED MSIC-TPG SCHEME**



**Fig 1** Symbolic Representation of an MSIC Pattern

This section develops a TPG scheme that can convert an SIC vector to unique low transition vectors for multiple scan chains. First, the SIC vector is decompressed to its multiple codeword's. Meanwhile, the generated codeword's will bit-XOR with a same seed vector in turn. Hence, a test pattern with similar test vectors will be applied to all scan chains. The Proposed MSIC-TPG consists of an SIC generator, a seed generator, an XOR gate network, and a clock and control block.

**2.1 Test Pattern Generation Method**

Assume there are  $m$  primary inputs (PIs) and  $M$  scan chains in a full scan design, and each scan chain has  $l$  scan cells. Fig. 1(a) shows the symbolic simulation for one generated pattern. The vector generated by an  $m$ -bit LFSR with the primitive polynomial can be expressed as  $S(t) = S_0(t)S_1(t)S_2(t), \dots, S_{m-1}(t)$  (hereinafter referred to as the seed), and the vector generated by an  $l$ -bit Johnson counter can be expressed as  $J(t) = J_0(t)J_1(t)J_2(t), \dots, J_{l-1}(t)$ . In the first clock cycle,  $J = J_0 J_1 J_2, \dots, J_{l-1}$  will bit-XOR with  $S = S_0S_1S_2, \dots, S_{m-1}$ , and the results  $X_1X_{l+1}X_{2l+1}, \dots, X_{(M-1)l+1}$  will be shifted into  $M$  scan chains, respectively. In the second clock cycle,  $J = J_0 J_1 J_2, \dots, J_{l-1}$  will be circularly shifted as  $J = J_{l-1} J_0 J_1, \dots, J_{l-2}$ , which will also bit-XOR with the seed  $S = S_0S_1S_2, \dots, S_{m-1}$ . The resulting  $X_2X_{l+2}X_{2l+2}, \dots, X_{(M-1)l+2}$  will be shifted into  $M$  scan chains, respectively. After  $l$  clocks, each scan chain will be fully loaded with a unique Johnson codeword, and seed  $S_0S_1S_2, \dots, S_{m-1}$  will be applied to  $m$  PIs. Since the circular Johnson counter can generate  $l$  unique Johnson code words through circular shifting a Johnson vector, the circular Johnson counter and XOR gates in Fig. 1 actually constitute a linear sequential decompressor

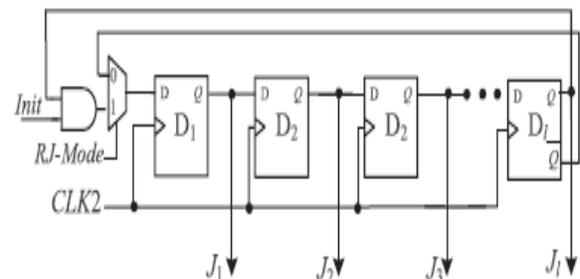
**2.2 Reconfigurable Johnson Counter**

According to the different scenarios of scan length, this paper develops two kinds of SIC generators to generate Johnson vectors and Johnson code words, i.e., the reconfigurable Johnson counter and the scalable SIC counter.

For a short scan length, we develop a reconfigurable Johnson counter to generate an SIC sequence in time domain.

As shown in Fig. 2(a), it can operate in three modes.

- 1) Initialization: When RJ\_Mode is set to 1 and Init is set to logic 0, the reconfigurable Johnson counter will be initialized to all zero states by clocking CLK2 more than 1 times.
- 2) Circular shift register mode: When RJ\_Mode and Init are set to logic 1, each stage of the Johnson counter will output a Johnson codeword by clocking CLK2 1 times.
- 3) Normal mode: When RJ\_Mode is set to logic 0, the reconfigurable Johnson counter will generate 2l unique SIC vectors by clocking CLK2 2l times.



**Fig.2** Reconfigurable Johnson Counter

### 2.3 Scalable SIC Counter

When the maximal scan chain length  $l$  is much larger than the scan chain number  $M$ , we develop an SIC counter named the “scalable SIC counter.” As shown in Fig. 2(b), it contains a  $k$ -bit adder clocked by the rising SE signal, a  $k$ -bit subtractor clocked by test clock CLK2, an  $M$ -bit shift register clocked by test clock CLK2, and  $k$  multiplexers. The value of  $k$  is the integer of  $\log_2(l - M)$ . The waveforms of the scalable SIC counter are shown in Fig. 2(c). The  $k$ -bit adder is clocked by the falling SE signal, and generates a new count that is the number of 1s (0s) to fill into the shift register. As shown in Fig. 2(b), it can operate in three modes

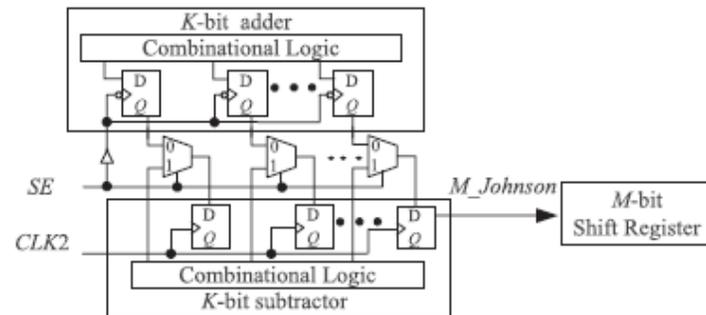


Fig 3 Scalable SIC Counter

- 1) If  $SE = 0$ , the count from the adder is stored to the  $k$ -bit subtractor. During  $SE = 1$ , the contents of the  $k$ -bit subtractor will be decreased from the stored count to all zeros gradually.
- 2) If  $SE = 1$  and the contents of the  $k$ -bit subtractor are not all zeros,  $M$ -Johnson will be kept at logic 1 (0).
- 3) Otherwise, it will be kept at logic 0 (1). Thus, the needed 1s (0s) will be shifted into the  $M$ -bit shift register by clocking CLK2 1 times, and unique Johnson codeword’s will be applied into different scan chains.

For example, after full-scan design, ISCAS’89 s13207 has 10 scan chains whose maximum scan length is 64. To implement a scalable SIC counter as shown in Fig. 2(b), it only needs 6 D-type flip-flops (DFFs) for the adder, 6 DFFs for the subtractor, 10 DFFs for a 10-bit shift register for 10 scan chains, 6 multiplexers, and additional 19 combinational logic gates. The equivalent gates are 204 in total. For a 64-bit Johnson counter, it needs 64 DFFs, which are about 428 equivalent gates. The overhead of a MSIC-TPG can thus be effectively decreased by using the scalable SIC counter..

### 2.4 MSIC-TPGs for Test-per-Clock Schemes

The MSIC-TPG for test-per-clock schemes is illustrated in Fig. 3(a). The CUT’s PIs  $X_1 - X_{mn}$  are arranged as an  $n \times m$  SRAM-like grid structure. Each grid has a two-input XOR gate whose inputs are tapped from a seed output and an output of the Johnson counter. The outputs of the XOR gates are applied to the CUT’s PIs. A seed generator is an  $m$ -stage Conventional

LFSR, and operates at low frequency CLK1. The test procedure is as follows.

- 1) The seed generator generates a new seed by clocking CLK1 one time.
- 2) The Johnson counter generates a new vector by clocking CLK2 one time.
- 3) Repeat 2 until  $2l$  Johnson vectors are generated.
- 4) Repeat 1–3 until the expected fault coverage or test length is achieved.

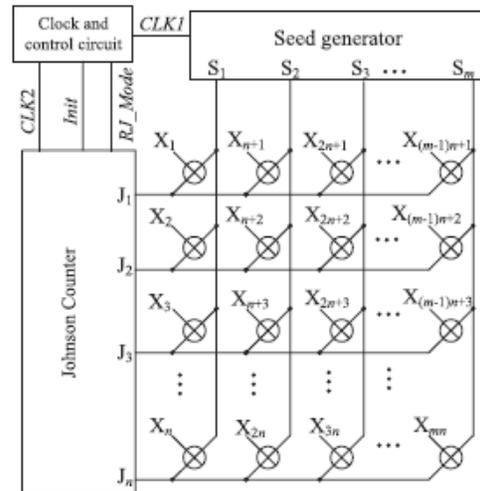


Fig 4 MSIC-TPGs test-per-clock

### 2.4 MSIC-TPGs for Test-per-Scan Schemes

The MSIC-TPG for test-per-scan schemes is illustrated in Fig. 3(b). The stage of the SIC generator is the same as the maximum scan length, and the width of a seed generator is not smaller than the scan chain number. The inputs of the XOR gates come from the seed generator and the SIC counter, and their outputs are applied to  $M$  scan chains, respectively. The outputs of the seed generator and XOR gates are applied to the CUT’s PIs, respectively. The test procedure is as follows.

- 1) The seed circuit generates a new seed by clocking CLK1 one time.
- 2)  $RJ\_Mode$  is set to “0”. The reconfigurable Johnson counter will operate in the Johnson counter mode and generate a Johnson vector by clocking CLK2 one time.
- 3) After a new Johnson vector is generated,  $RJ\_Mode$  and  $Init$  are set to 1. The reconfigurable Johnson counter operates as a circular shift register, and generates 1 codewords by clocking CLK2 1 times. Then, a capture operation is inserted.
- 4) Repeat 2–3 until  $2l$  Johnson vectors are generated.
- 5) Repeat 1–4 until the expected fault coverage or test length is achieved.

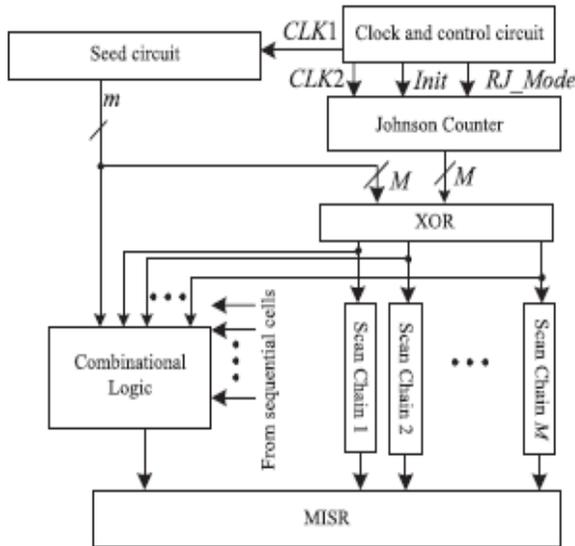


Fig 5 MSIC-TPGs Test-Per-Scan schemes

### 3. PRINCIPLE OF MSIC SEQUENCES

The main objective of the proposed algorithm is to reduce the switching activity. In order to reduce the hardware overhead, the linear relations are selected with consecutive vectors or within a pattern, which can generate a sequence with a sequential decompressed, facilitating hardware implementation. Another requirement is that the MSIC sequence should not contain any repeated test patterns, because repeated patterns could prolong the test time and reduce test efficiency [5]. Finally, uniformly distributed patterns are desired to reduce the test length (number of patterns required to achieve a target fault coverage) [21]. This section aims to extract a class of test sequences that meets these requirements.

#### 3.1 Example

Suppose there are four scan chains whose lengths are 8 in a full-scan design. The 8-bit Johnson codeword's and 4-bit (or more bits) seeds can be used to generate an MSIC sequence. A MSIC sequence with seed 01101 is

For  $(25 - 1)$  unique seeds, there are 31 such blocks. Therefore, there are  $16 \times 31 = 496$  unique patterns in this MSIC sequence. Also, the transition between patterns  $X(t)$  and  $X(t+1)$  is one for every scan chain, and transitions for scan chain 2 to scan in "100111" are two per clock. Seed "01101" is kept unchanged for patterns  $X(1)$  to  $X(16)$ .

### 4. PERFORMANCE ANALYSIS

To analyze the performance of the proposed MSIC-TPG, Experiments on ISCAS'85 benchmarks and standard full-scan Designs of ISCAS'89 benchmarks are conducted. The performance Simulations are carried out with the Synopsys Design Analyzer and Prime Power. Fault simulations are

carried Out with Synopsys Tetramax. Syntheses are carried out with Nangate library based on 45-nm typical technology. The testFrequency is 100 mhz, and the power supply voltage is 1.1 V. The test application method is test-per-clock for ISCAS'85 Benchmarks and test-per-scan for ISCAS'89 benchmarks. The Number of scan chains is 10 for s13207 and s15850, and 20 For s38417, s35932 and s38584.

#### 4.1 Fault Coverage Comparison

The fault coverage's for ISCAS benchmarks with the MSIC-TPG, the conventional LFSR, and the methods in [7] and [21]. In this table, the columns labeled SFC, TFC, and TL refer to the stuck-at fault coverage, transition fault coverage, and test length, respectively. The TFC values of this table correspond to launch-on-shift test patterns. In order to achieve fair comparisons, for ISCAS'89 benchmarks, our TLs are chosen to be the same as those in [7]. For the first three ISCAS'85 benchmarks, our TLs are chosen according to [21]. The last two ISCAS'85 benchmarks are selected to have similar fault coverage's as [7].

Compared with the conventional LFSR, the MSIC-TPG achieves similar stuck-at fault coverage for ISCAS'85 benchmarks and higher stuck-at fault coverage for ISCAS'89 benchmarks except for s38417. It also achieves higher transition fault coverage for both ISCAS'85 and ISCAS'89 benchmarks except for s35392. Compared to the LT-LFSR method in [21], it can be seen from the ISCAS'85 benchmarks that, with the same TL, the MSIC-TPG achieves similar test efficiency.

Compared to the TPG in [7], the results with ISCAS'89 benchmarks show that, with the same TL, the MSIC-TPG has higher efficiency, except for s38417. Note that, for ISCAS'89 benchmarks, the results of our method and those of [21] in the table should not be compared directly because of the different TLs.

#### 4.2 Average and Peak Power Reduction

The total and peak power reductions of CUTs with the MSIC-TPG and with the LFSR-TPG, where columns labeled "Ptot" and "Ppeak" refer to the CUT's total power and peak power with the MSIC-TPG, respectively. Columns labeled "\_Ptot" and "\_Ppeak" refer to the percentages of different methods' total power reduction and peak power reduction with respect to the LFSR method, respectively. The results of [25] are also included for comparison. For ISCAS'85 benchmarks, the MSIC-TPG saves 25%–50.0% total power and 15.6%–32.6% peak power against the conventional LFSR. All total power reductions and peak power reductions are higher than those with variable-length ring counter in [25].

For ISCAS'89 benchmarks, the MSIC-TPG can save 21.38%–34.52% total power and 3%–32.4% peak power against the conventional LFSR. The total power reductions and peak power

reductions are not higher than those with the variable-length ring counter in [25]. One reason for this is that the power simulation results for the MSIC-TPG are with 45-nm technology, whereas results in [25] are with 0.25- $\mu$ m technology. Experimental results show that leak power contributes about 10% to total power with 45-nm technology, but contributes little to total power with 0.18- or 0.25- $\mu$ m technology.

**5. RESULTS**

In this paper BIST will be used to generate the TEST PATTERN GENERATION. To evaluate the proposed model, the LFSR is used to measure power and area in this model we are used Johnson counter, scalable SIC counter.

**5.1 Simulation Procedure**

**5.1.1 Simulate Seed Generator**

Step 1: Open the ModelSim SE 6.3f software and use the browse button in the project location box to specify the location of the directory that you created for the project. ModelSim uses a working library to contain the information on the design in progress; in the Default Library Name field we used the name work. Click OK.

Step 2: Go to the Library window, select the pattern and right click the pattern to simulate the program.

Step 3: Again go to the Library window, select the pattern and right click that to add the wave. If enable the clock 2 we will get test pattern output.

Step 4: Generate the seed generator value, enable the seed generator value 01101 and then get output of seed generator. It will be change until 8 bit and single input changes.

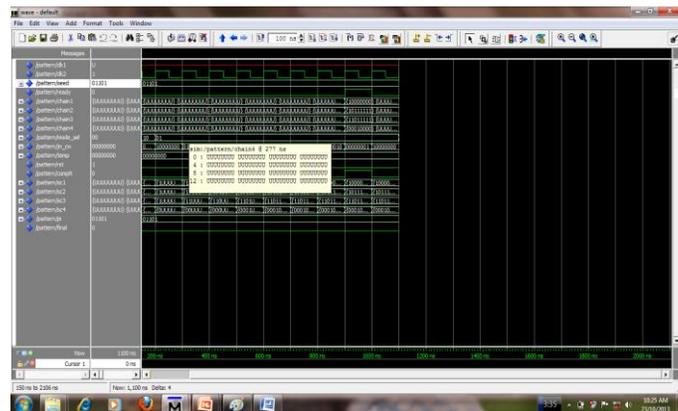


Fig 6 Output for pattern method

**5.1.2 Find Fault Occurrence**

Step 1: Go to the Library, select the Test Main. right click the test main, to write the program and simulate.

Step 2: Opened simulated window add to wave. if fault value is '0' means fault is not enabled. If fault value '1' means fault enabled and will be get different output

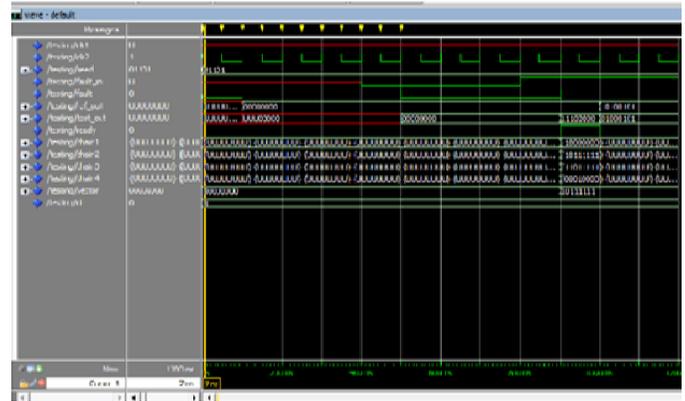
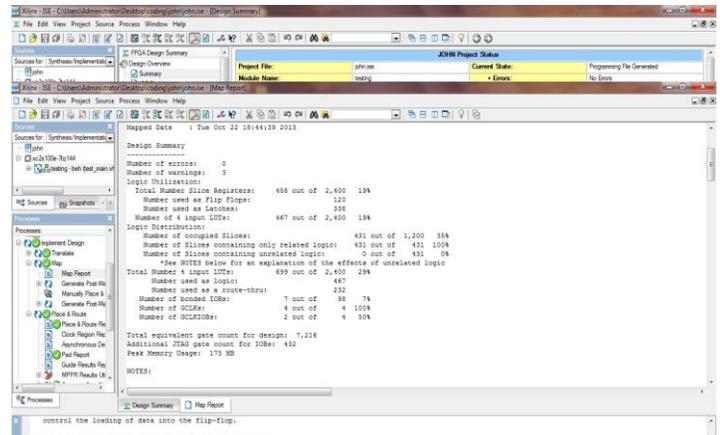


Fig 7 Output for testing method

**5.1.3 Simulate Power and Area**

Step 1: open the Xilinx ISE 13.2 software.

Step 2: open the new file, to write the program for power and area and simulate



**Design summary**

Step 3: Finally get the output of total equivalent gate count design is 7.216

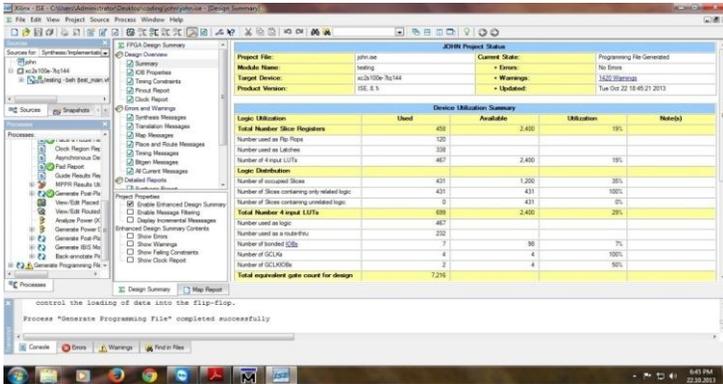


Fig 8 Output of power and area

## 6. CONCLUSIONS

This paper has proposed a low-power test pattern generation method that could be easily implemented by hardware. It also developed a theory to express a sequence generated by linear Sequential architectures, and extracted a class of SIC sequences named MSIC. Analysis results showed that an MSIC sequence had the favorable features of uniform distribution, low input Transition density, and low dependency relationship between the test length and the TPG’s initial states. Combined with the proposed reconfigurable Johnson counter or scalable SIC counter, the MSIC-TPG can be easily implemented, and is flexible to test-per-clock schemes and test-per-scan schemes. For a test-per-clock scheme, the MSIC-TPG applies SIC sequences to the cut with the SRAM like grid. For a test-per-scan scheme, the MSIC-TPG converts and sic vector to low Transition vectors for all scan chains. Experimental results and analysis results demonstrate that the MSIC-TPG is scalable to scan length, and has negligible impact on the test overhead.

## REFERENCES

[1]. Feng Liang, Luwen Zhang, Shaochong Lei, Guohe Zhang, Kaile Gao, and Bin Liang” Test Patterns of Multiple SIC Vectors: Theory and Application in BIST Schemes” IEEE transactions on very large scale integration (vlsi) systems, vol. 21, no. 4, april 2013

[2]. A. Abu-Issa and S. Quigley, “Bit-swapping LFSR and scan-chain ordering: A novel technique for peak- and average-power reduction in scan-based BIST,” IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol. 28, no. 5, pp. 755–759, May 2009.

[3]. Y. Bonhomme, P. Girard, L. Guiller, C. Landrault, and S. Pravossoudovitch, “A gated clock scheme for low power scan testing of logic ICs or embedded cores,” in Proc. 10th Asian Test Symp., Nov. 2001, pp. 253–258.