TPG Applications using LFSR

Mohammad Feroz Khan 1*, Devunoori Sandeep 2, Mohd Khaja Yakoob Hussaini 3, Vaddepally Ashok 4, Lisbeth Priyadharshini 5

1,2,3,4 B.Tech, Department of ECE, Kakatiya Institute of Technology and Science, Warangal-015, India.
5 Assistant Professor, Department of ECE, Kakatiya Institute of Technology & Science, Warangal-015, India.

1. Introduction

The development of modern microelectronics industry allows us to implement complex digital systems on a single chip. To design such a system is not a trivial task anymore because the increasing complexity rises a whole set of different problems such as size, speed and power consumption of the chip. Complex VLSI testing problems such as BIST technique has been extensively studied and widely used nowadays. Various approaches introduce an embedded test structure into the circuit under test (CUT) to make testing easier and better [4]. As we know BIST, the test patterns are generated and applied to the CUT by on chip hardware, minimizing hardware overhead which is a major concern of BIST implementation. Unlike stored pattern BIST, which requires high hardware overhead due to memory devices required to store precomputed test patterns, pseudo random BIST, whereas test patterns are generated by pseudorandom pattern generators such as LFSR. LFSR are the logic circuits used to create pseudorandom binary sequences (PRBSs) [10]. An LFSR circuit consists of set of M registers and feedback taps that determine the sequence of states that the LFSR transitions through. The feedback taps are described by a modulo-2 polynomial. A primitive polynomial generates a maximal length of m-sequence, where the LFSR transitions through $2^m-1$ states before repeating, there is a single unused LFSR state. The PRBS is the binary output of the LFSR. The binary sequence is described as pseudorandom, as the sequence is deterministic, yet it has the correlation properties of a random sequence. LFSRs and PRBSs are widely used in digital systems.

In general, power consumption of a system in normal mode is less than in testing modes, the four reasons for the increase of power during testing mode are:

- High switching activity due to nature of test patterns.
- Parallel activation of internal cores during test.
- Power consumed by extra design –for-test-circuitry.
- Low correlation among test vectors. [1][2][8]

The rest of the paper is as follows. In section II previous work related to TPG are discussed. In section III TPG architecture is explained using LFSR. In section IV Algorithm is explained for the proposed model. Section V explains the applications of TPG. In section VI simulation...
results are carried out. In section VII the conclusion is summarized.

TPG has been used previously for BIST [7] and for System on chip architecture (SOC) [9] effectively using low power linear feedback shift register (LP-LFSR). However, there are some major drawbacks for this BIST technique whose architecture is based on the LFSR. One is that the BIST circuit introduces more switching activities in the circuit under test during test than that during normal operation. This can cause excessive power dissipation and results in delay penalty into the design, which are responsible for mismatch of data while transferring from one place to another, hence secure transferring, may not be easier. The switching activity in the circuit can be significantly higher during BIST than during its normal operation. Excessive switching activity during test can cause several problems. To lower the power dissipation in test mode, many techniques have been proposed to reduce the switching activities of test pattern. TPG using LP-LFSR for braun array multiplier is designed for different sizes [3], the LFSR is commonly used as a TPG in low overhead BIST. This is due to the fact that an LFSR can be built with little area overhead and used not only as a TPG, which provides high fault coverage for a large class of circuits, but also as an output response analyzer. Hence in LFSR based TPG [3][7][9] a low power LFSR architecture and high fault coverage designs is proposed for BIST architecture. Hence for the proposed applications below we can use these proposed LP-LFSR’s so that we can also achieve low power consumptions and the efficient designs with optimized characteristics for efficient performance of the security applications.

3. TPG applications using LFSR

TPG means generating sequences which are predictable and these sequences are called pseudo random binary sequences as the same sequences repeats after N elements, unlike real random sequence [3]. As in figure 1 the implementation of this can be done by using LFSR, which is an n-bit shift register which pseudo-randomly scrolls between $2^n-1$ values, but does it very quickly because there is minimal combinational logic involved. Once it reaches its final state, it will traverse the sequence exactly as before. LFSR mainly works on shift registers and feedback function (tapping).

A shift register is a device whose identifying function is to shift its contents into adjacent positions within the register or, in the case of the position on the end, out of the register. The position on the other end is left empty unless some new content is shifted into the register.

Two uses for a shift register are:
- Convert between parallel and serial data.
- Delay a serial bit stream.

The feedback is done so as to make the system more stable and free from errors. Specific taps are taken from the tapping points and then by using the XOR operation on them they are feedback into the registers [5].

4. Algorithm for Proposed Model

The algorithm for TPG is as follows:
- Consider a 4 - bit LFSR with polynomial $1 + x^3 + x^4$ in which feedback action is done for tap3 and tap4 using synchronous D flip flops.
- Gray codes are generated using counter inputs.
- In the next stage the output from the counter is fed to the NOR gate, when all the bits of the counter output are zero then NOR gate produces logic 1 as output and for remaining bits it produces logic 0.
- In the next stage, the outputs from the NOR gate initializes the seed to LFSR, when input is high to LFSR it retains the value which is initialized to 0001.
- When input is low to LFSR it operates according to the feedback action and shifting occurs.
- In the next stage the LFSR outputs are Exclusive-ORed and test pattern generation is obtained.
- The same process can be repeated for inverse gray codes and respective TPG is obtained.

The operation of proposed method is as illustrated as followed in Table I.
Table 1. Operation of proposed method

<table>
<thead>
<tr>
<th>Binary bits</th>
<th>Gray codes(g1)</th>
<th>Inverse gray codes(g2)</th>
<th>LFSR(y)</th>
<th>y ⊕ g1</th>
<th>y ⊕ g2</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td>0000</td>
<td>0000</td>
<td>0001</td>
<td>0001</td>
<td>0001</td>
</tr>
<tr>
<td>0001</td>
<td>0001</td>
<td>1110</td>
<td>0001</td>
<td>0000</td>
<td>1111</td>
</tr>
<tr>
<td>0010</td>
<td>0011</td>
<td>0011</td>
<td>1000</td>
<td>1011</td>
<td>1011</td>
</tr>
<tr>
<td>0011</td>
<td>0010</td>
<td>1101</td>
<td>0100</td>
<td>0110</td>
<td>1001</td>
</tr>
<tr>
<td>0100</td>
<td>0110</td>
<td>0110</td>
<td>0010</td>
<td>0100</td>
<td>0100</td>
</tr>
<tr>
<td>0101</td>
<td>0111</td>
<td>1000</td>
<td>1001</td>
<td>1110</td>
<td>0001</td>
</tr>
<tr>
<td>0110</td>
<td>0101</td>
<td>0101</td>
<td>1100</td>
<td>1001</td>
<td>1001</td>
</tr>
<tr>
<td>0111</td>
<td>0100</td>
<td>1011</td>
<td>0110</td>
<td>0010</td>
<td>1101</td>
</tr>
<tr>
<td>1000</td>
<td>1100</td>
<td>1100</td>
<td>1011</td>
<td>0111</td>
<td>0111</td>
</tr>
<tr>
<td>1001</td>
<td>1101</td>
<td>0010</td>
<td>0101</td>
<td>1000</td>
<td>0111</td>
</tr>
<tr>
<td>1010</td>
<td>1111</td>
<td>1111</td>
<td>1010</td>
<td>0101</td>
<td>0101</td>
</tr>
<tr>
<td>1011</td>
<td>1110</td>
<td>0001</td>
<td>1101</td>
<td>0011</td>
<td>1100</td>
</tr>
<tr>
<td>1100</td>
<td>1010</td>
<td>1010</td>
<td>1110</td>
<td>0100</td>
<td>0100</td>
</tr>
<tr>
<td>1101</td>
<td>1011</td>
<td>0100</td>
<td>1111</td>
<td>0100</td>
<td>1011</td>
</tr>
<tr>
<td>1110</td>
<td>1001</td>
<td>1001</td>
<td>0111</td>
<td>1110</td>
<td>1110</td>
</tr>
<tr>
<td>1111</td>
<td>1000</td>
<td>0111</td>
<td>0011</td>
<td>1011</td>
<td>0100</td>
</tr>
</tbody>
</table>

5. Applications

5.1. PN Sequence Generation

A Pseudo Random Binary Sequence Generator as in figure 2 actually consists of a LFSR which is a sequential shift register with combinational logic that causes it to pseudo-randomly cycle through a sequence of binary values.

![Figure 2: PN Sequence generation](image)

Let us consider LFSR with polynomial $1 + x + x^3$ with sequence the tapping action is done and the result produced is fed back.

For n-bit shift register the length of PN sequence produced is $2^n-1$ which is illustrated below in table 2.

Table 2. PN Sequence outputs

<table>
<thead>
<tr>
<th>States</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>111</td>
<td>1</td>
</tr>
<tr>
<td>011</td>
<td>1</td>
</tr>
<tr>
<td>101</td>
<td>1</td>
</tr>
<tr>
<td>010</td>
<td>0</td>
</tr>
<tr>
<td>001</td>
<td>1</td>
</tr>
<tr>
<td>100</td>
<td>0</td>
</tr>
<tr>
<td>110</td>
<td>0</td>
</tr>
<tr>
<td>111</td>
<td>1</td>
</tr>
</tbody>
</table>

5.2. Security

The test pattern generation can be implemented using different encoding techniques. Let us consider gray and
inverse gray codes, the output are calculated by EX-ORing the seed generated from LFSR for the polynomial $1 + x^2 + x^4$. Let the outputs be F1, F2 respectively.

$$y_1 1 1 8 4 A 5 2 1 8 4 A 5 2 1 8 4$$

$$F1 1 0 B 6 C 2 7 5 4 9 5 B 8 A 1 6$$

$$F2 1 F B 9 C D 7 A 4 6 5 4 8 5 1 3$$

5.3. Data Compression

Data compression is mainly based on LFSR reseeding, reseeding leads to repetition of similar bits which leads to reduction of test data storage [6].

Consider the above polynomial the output of the LFSR is as follows

$$y 1 1 8 4 A 5 2 1 8 4 A 5 2 1 8 4$$

6. Simulation Results

The TPG is obtained with external LFSR using gray and inverse gray codes for the polynomial $1 + x^2 + x^4$, and also the PN sequences are generated at LFSR. The simulation results for gray and inverse gray codes are illustrated in figure 5 and figure 6 respectively as below.

Let us consider the value at the counter k(12) be 1100 as per the proposed algorithm the gray code g(12) generated is 1010 and in the next step of operation LFSR output y(12) will be 0010 is exored with gray code to get the final result x(12) equal to 1000 i.e 1010 exor 0010.

The explanation of the simulation results given for the polynomial $1 + x^2 + x^4$ holds good in a similar manner for other polynomials.

If considered the simulation results of TPG for the polynomial $1 + x^3 + x^4$ using gray and inverse gray codes are illustrated in figure 7 and figure 8 respectively as below.

The simulation results and power report is synthesized using Xilinx Version 13.4, Spartan3E. The RTL view of the proposed model is given in figure 9.

The process can also be extended to N bits using different polynomials we get different TPG codes.
7. Conclusion

The test pattern generator model has been proposed using LFSR which can be used for wireless communication applications. To generate maximum range of binary digits taping is done to LFSR for different polynomials to get different TPG codes. From the simulation results it can be summarized that proposed method of TPG can be used efficiently in security transmission of codes and with low power consumption and also can be used for applications like Data compression, PN sequence generation.

8. Acknowledgment

We are thankful to R. Sridevi, Associate Professor, Department of ECE, KITS Warangal for her continuous support and advice on our work.

References