

Effective LFSR Reseeding Technique for Achieving Reduced Test Pattern

¹S. Saravanan and ²Har Narayan Upadhyay

¹School of Computing (SOC),

²School of Electrical and Electronics Engineering (SEEE), SASTRA University,
Thanjavur-613401, India

Abstract: Aim of this study is to focus on reducing test pattern with effective Linear Feedback Shift Register (LFSR) reseeding. Test data volume of modern devices for testing increases rapidly corresponding to the size and complexity of the Systems-on-Chip (SoC). LFSR is a good pseudorandom pattern generator, which generates all possible test vectors with the help of the tap sequence. It can achieve high fault coverage by reducing correlation between the test vectors. Reseeding is a powerful method for reducing the test data volume and storage. This study presents a new LFSR reseeding technique for efficient reduction of test pattern. A new encoding technique is proposed in this study which is used to reduce the size of the test data. Size of the test data can be reduced by LFSR clock which is inactive for several clock cycles after the input seed is given. When the clock goes to inactive state, a rotate right shift operation is done on the seed to get all the remaining possible values. After getting all the possible values for that seed a new seed is given by making the clock active. Test data volume is reduced by storing the data only when the clock is active. With in the reduced clocks, rest of all the remaining test vectors was derived. A special Control logic is used to make the clock active as well as inactive. Experimental results are targeted to ISCAS89 benchmark circuits.

Keywords: Clock active and inactive, control logic, encoding, linear feedback shift register reseeding,

INTRODUCTION

Today's challenging problem on System-on-chips (SoCs) design and test focuses on huge test data and its corresponding storage space. The test data volume increases rapidly for the modern devices in order to manufacture test pattern for them. This is because exponential increase of transistors for the chip and also by the exposure of modern technology causes many physical and analytical defects. This may be the use of new test data to test its functionality.

Linear Feedback Shift Registers (LFSR) can generate good pseudorandom patterns. Clock is the only signal which is required to generate those test patterns. When the LFSR is clocked after giving the input seed to the LFSR, it will generate all the possible test patterns for the input seed for each clock. Note that the seed value can be anything except all 0's, which produces all the test patterns as 0. The LFSR generates it patterns according to its feedback polynomial. The LFSR has the behaviour of achieving high-fault coverage by reducing correlation among the test patterns.

In practical, the user can design an LFSR for many bits for getting large number of test patterns. But, there are some practical restrictions while designing an LFSR with higher seed width.

For example, almost 5 min can be taken to generate over 4 billion test patterns by a 32-bit maximal-length LFSR at a rate of 16-MHz clock (John, 1996). It also uses more space to store all the patterns. So it is necessary to reduce the test data volume by using compression technique.

Reseeding is a very powerful method for reducing test data. Most of the test data reduction is mainly concentrating on LFSR reseeding. The basic idea in LFSR reseeding is to generate deterministic test cubes by expanding seeds. A seed is an initial state of the LFSR that is expanded by running the LFSR in autonomous mode. Since typically only 1-5% of the bits in a test vector are specified, most bits in a test cube do not need to be considered when a seed is computed because they are don't care bits. Therefore, the size of a seed is much smaller than the size of a test vector.

LITERATURE REVIEW

Many LFSR reseeding techniques has been proposed. In DS-LFSR (Seongmoon and Sandeep, 2002), a low power LFSR is proposed to reduce the switching activity and also to achieve high fault coverage by using dual clock. A low transition and less data storage LFSR reseeding technology is proposed in

(Lee and Touba, 2004). Average and peak power of the circuit can be reduced by reducing the number of transitions as proposed in LT-LFSR (Tehranipoor *et al.*, 2005). A multiple-output low-power LFSR (Rajendra *et al.*, 2006) is proposed by getting the output of several clock cycles at once by reducing the flip-flop's activity factors. Another LFSR reseeding scheme is proposed in Myung-Hoon *et al.* (2007), which prevent all transitions in the non-transitional block and reduce the number of specified bits. In Jinkyu and Touba (2007), a new scheme of LFSR reseeding is proposed to reduce the number of transitions and also to reduce the test data storage. In Mayank and Soundra Pandian (2009), the Total Hamming Distance (THD) is reduced by making the clock inactive for a period when the same pattern is repeated and also by reordering the test vectors. Power optimization of LFSR is done in Balwinder *et al.* (2009), by increasing the correlation between the successive bits.

In this study, we present a new LFSR reseeding scheme to reduce the number of clock cycles used to produce the test vectors. Also we reduced the test data storage in the design. A control FSM is used to reduce the clock cycle by making the clock inactive for several outputs. From the output where the clock is active, the remaining vectors are derived. We are going to store only the outputs when the clock is in active position. So the test data storage is reduced.

PROPOSED METHODOLOGY

In the proposed methodology, the outputs of the LFSR were reordered to reduce the test data storage by reducing clock signal input. The proposed methodology is shown in Fig. 1. The input seed is taken and is given to LFSR. The LFSR starts to give the output for each and every clock cycle, which is given to Circuit Under Test (CUT). If the reset signal is forced high, then it resets the LFSR. If it is low, it started to produce the output. The output of the LFSR is given to both CUT and also to control FSM, which includes a counter, control circuit and a rotate right shifter. The LFSR output is given to both counter and control circuit. Counter counts the number of outputs produced by the LFSR. If the counter value reaches the period, then it resets the LFSR.

The main function of the control circuit is to check the output of LFSR and gives a control signal to both LFSR and inbuilt rotate right shifter. According to the control signal the LFSR either enables or disables its function by making the clock active or inactive. The control signal also controls the number of shifts of the rotate right shifter. The control signal disables the function of the LFSR, while shifting is going on. The control signal enables the function of LFSR, if the shifting is done or no need of any shifting. The shifted output is given to CUT. By using control FSM,

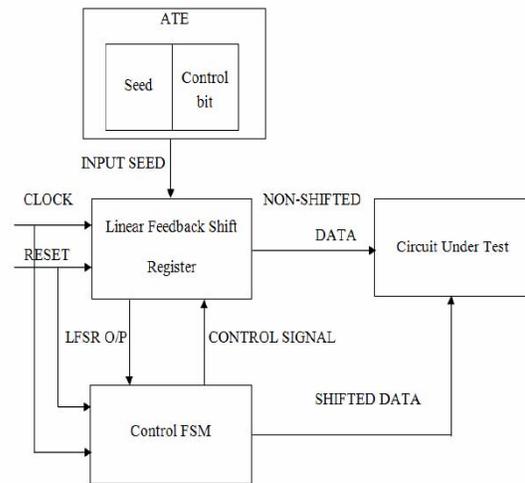


Fig. 1: Block diagram

the LFSR functionality is reduced. The control FSM checks the LFSR output with the look-up table values and starts to shift the value in order to derive the remaining values of the LFSR. During the shift, the LFSR is in disabled condition. LFSR is enabled, when there is no shift is required or the shifting is finished, in order to produce the remaining outputs of the LFSR. By this way we can reduce the functionality of LFSR.

For example, the input seed of the LFSR is taken as 3. If the reset is active high or the input seed is "000", then it reset the LFSR. If the input seed is other than "000", the LFSR starts shifting the given seed for each clock cycle. Let the given input seed is "001". The LFSR starts to produce its output starting from "001". Then the output "001" is given to the control FSM, which includes a counter, control circuit and a rotate right shifter. The output of the LFSR is fed to both counter and control circuit. Counter is used to count the number of outputs at LFSR and resets when the count is 7 (Input seed length, $n = 3$ (i.e., feedback polynomial = $X^3 + X^2 + 1$ and period = $(2^n - 1) = 7$)). The control circuit provides control for enabling and disabling the LFSR and also controls the number of shifts of the shifter. The look-up table for 3-bit input seed is given in Table 1.

- Input seed length, $n = 3$, (i.e., feedback polynomial = $X^3 + X^2 + 1$ (from Table 2) and Period = $(2^n - 1)$)
- The given input seed = "001"
- The possible LFSR outcomes = {001, 100, 010, 101, 110, 111, 011}.

Initially the counter value is 0. The first output of the LFSR is "001". This value is then given to control FSM. Now the counter is incremented by 1.

Table 1: Truth table for no of shifts

Input	# Shifts	Control
001	0	1
100	1	1
010	2	1
101	0	1
110	1	1
011	2	1
111	0	0

Table 2: Polynomial value

Bits N	Feedback polynomial (maximal length)	Period (2n-1)
2	x^2+x+1	3
3	x^3+x^2+1	7
4	x^4+x^3+1	15
5	x^5+x^3+1	31
6	x^6+x^5+1	63
7	x^7+x^6+1	127
8	$x^8+x^6+x^5+x^4+1$	255
9	x^9+x^5+1	511
10	$x^{10}+x^7+1$	1023
11	$x^{11}+x^9+1$	2047
12	$x^{12}+x^{11}+x^{10}+x^4+1$	4095
13	$x^{13}+x^{12}+x^{11}+x^8+1$	8191
14	$x^{14}+x^{13}+x^{12}+x^2+1$	16383
15	$x^{15}+x^{14}+1$	32767
16	$x^{16}+x^{14}+x^{13}+x^{11}+1$	65535
17	$x^{17}+x^{14}+1$	131071
18	$x^{18}+x^{11}+1$	262143
19	$x^{19}+x^{18}+x^{17}+x^{14}+1$	524287

Also the control circuit checks the value with the LUT. From the table we can clearly know that, we can able to derive the value “100” by rotating “001” to one bit right and “010” by rotating “001” to two bit right. This is achieved by the control signal given to rotate right shifter.

While during the shift, a control signal is given to LFSR to stop its function, because we are deriving the value “100” and “010” from “001”. So there is no need for the working of LFSR during that time. The signal produced by the control circuit during the shift is ‘1’ (which disables LFSR for next 3 clock cycles, at 1st cycle no shift is done and shift is made for 2nd and 3rd clock cycle).

After “010” is derived, LFSR is enabled and produce the next output “101”. Now “101” is given to control FSM and then “110” and “011” was derived by the same process. If the LFSR output is “111”, then the control signal produced is ‘0’, where no shift is done and also enables the LFSR to produce the next output. These outputs of the LFSR are fed to the circuit under test block. For each and every process the counter is incremented. If the counter value = 7, then the LFSR and counter resets. In this method the LFSR is enabled

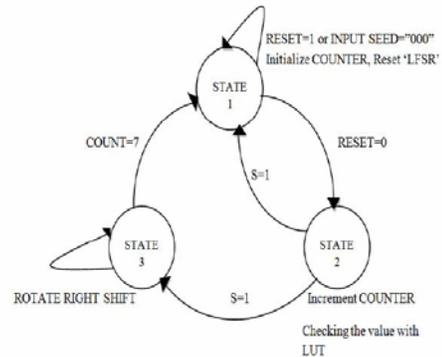


Fig. 2: Proposed state diagram

for 3 clock cycles alone. From the outputs the remaining values are derived by shifting those values. So working of LFSR is reduced and can able to produce the output of the LFSR with reduced clock cycles. In the above example, we can able to get all the 7 values with three clock cycles itself.

Table 2 shows feedback polynomial of common LFSR methods. It also shows its corresponding polynomial weight. The state diagram is shown in Fig. 2. The State 1 is the initializing state to initialize the counter and LFSR. State 2 is the control circuit to produce the control signal ‘S = 0/1’. State 3 is the rotate right shifter. When RESET signal is high, or the input seed is “000”, then both LFSR and counter is initialized (State 1). When the RESET signal is active low, the input seed is given to LFSR. From that seed the LFSR starts to produce the output. The first output of the LFSR is given to control unit (State 2). Now the counter is incremented. The control unit then produces a control signal (S = 1/0) according to the LFSR output to both shifter (State 3) and LFSR (State 1). If the shifting of the LFSR output is possible means, then the control signal S = 1 is sent else S = 0 is sent. According to the possible shifting, the number of shifts also decided by the control signal. For each and every shift the counter is incremented. If the counter value reaches the period then both LFSR and counter resets.

Table 3 shows clock pulse activity of proposed method. This also shows that conventional LFSR need only 5 clock pulses to activate.

RESULTS AND DISCUSSION

The experimental results for the proposed methodology are shown in this section. Efficient compression is achieved by proposed LFSR reseeding method. This proposed encoded method is done with VHDL and it is tested with ISCAS89 benchmark circuits in full scan based design. Tetramax test pattern

Table 3: Proposed Clock Saving From this, it is clearly known that the

Clock pulse (conventional LFSR)	4-bit LFSR output	Clock pulse (proposed design)
1	0001	1
2	1000	-
3	0100	-
4	0010	-
5	1001	2
6	1100	-
7	0110	-
8	0011	-
9	1101	3
10	1110	-
11	0111	-
12	1011	-
13	0101	4
14	1010	-
15	1111	5

Table 4: Experimental results of proposed CR in %

Circuits	No. of bits per pattern length	Existing standard LFSR	Proposed LFSR in CR (%)
S27	5	31	77
S298	20	1048575	95
S344	33	8589934592	97
S382	27	134217728	96.2

with 100% fault coverage is used in this proposed method.

Table 4 shows various ISCAS89 benchmark circuit in its first column. In the second and third column it gives number of bits per pattern length and existing standard LFSR pattern. In the last column proposed Compression Ratio (CR) is shown. This shows significant improvement in test pattern compression when compared with existing LFSR method. Proposed Compression Ratio (CR) is achieved by 77 to 97%. This directly shows that memory space for storing the test pattern and its activation time is also considerably reduced.

proposed design is memory efficient. The proposed reseeding methodology reduces the number of transitions when compared to the conventional LFSR. Note that, the transition values may vary according to the seed taken.

CONCLUSION

LFSR reseeding is a powerful methodology for test data compression and also for reducing the test data storage. A new LFSR reseeding scheme is proposed for effective test data compression. In the proposed methodology we are going to store the data whenever

the clock is active. From the stored data the remaining values (when the clock is inactive) are derived. By this way we can achieve the efficient test data compression and also we can reduce the test data storage. Experimental results show that above 77% of test pattern compression is achieved by using LFSR reseeding. It shows that clearly proposed design is superior in terms of efficient memory and reduced clock transitions.

REFERENCES

Balwinder, S., K. Arun and B. Sukhleen, 2009. Power optimization of Linear Feedback Shift Register (LFSR) for low power BIST. IEEE International Advance Computing Conference, IACC 2009, Mohali, pp: 311-314.

Jinkyu, L. and N.A. Touba, 2007. LFSR-reseeding scheme achieving low-power dissipation during test. IEEE T. Comput. Aid. Design Integ. Circuit. Syst., 26(2).

John, K., 1996. What's an LFSR., Texas Instruments Incorporated.

Lee, J. and N.A. Touba, 2004. Low power test data compression based on LFSR reseeding. Computer Proceedings of IEEE International Conference on Design: VLSI in Computers and Processors, ICCD, TX, USA, pp: 180-185.

Mayank, S. and K.K. SoundraPandian, 2009. A Power Reduction Technique for Built-In-Self Testing Using Modified Linear Feedback Shift Register. World Academy of Science, Engineering and Technology.

Myung-Hoon, Y., K. Yongjoon, P. Youngkyu, L. Daeyeal, Y. Hyunjun and K. Sungho, 2007. LFSR reseeding methodology for low power and deterministic pattern. Proceedings of International on SOC Design Conference, pp: 145-148.

Rajendra, S.K., R. Xiaoyu and K. Hareesh, 2006. Multiple-output low-power linear feedback shift register design. IEEE T. Circuit. Syst. Regular studies, 53(7).

Seongmoon, W. and K.G. Sandeep, 2002. DS-LFSR: A BIST TPG for low switching activity. IEEE T. Comput. Aid. Design Integ. Circuit. Syst., 21(7).

Tehraniipoor, M., M. Nourani and N. Ahmed, 2005. Low transition LFSR for BIST-based applications. Proceedings of 14th Asian Test Symposium, 18-21 Dec., Univ. of Maryland Baltimore County, pp: 138-143.