# An LFSR-based address generator using optimized address partition for low power memory BIST

YU Zhi-guo[1,2], LI Qing-qing[1,2], FENG Yang[1,2], GU Xiao-feng[1,2]

(1. *Department of Electronic Engineering, Jiangnan University, Wuxi 214122, China*;
2. *Engineering Research Center of IoT Technology Applications of Ministry of Education, Jiangnan University, Wuxi 214122, China*)

**Abstract**: Power consumption in test mode is much higher than that in normal mode, which is prone to causing circuit damage and reducing the yield of chips. To reduce the power dissipation efficiently, a modified linear feedback shift register (LFSR) is designed to decrease switching activity dramatically during the generation of address sequences for memory built-in self-test (MBIST). The address models are generated by a blend of two address generators with an optimized address partition and two distinct controlled clock signals. An address generator circuit for MBIST of 64 k×32 static random access memory (SRAM) is designed to illustrate the proposed scheme. Experimental results show that when the address bus size is 16 bits, compared with the traditional LFSR, the proposed LFSR can reduce the switching activity and dynamic power by 71.1% and 68.2%, respectively, with low area overhead.

**Key words**: address sequence; linear feedback shift register (LFSR); memory built-in self-test (MBIST); address generator; switching activity

**CLD number**: TP333 **doi**: 10.3969/j.issn.1674-8042.2020.03.001

## 0 Introduction

With process technology continuing to shrink, a large number of embedded memories have been integrated into system on chips (SoCs)[1], which may make memories more susceptible to defects[2-4]. Due to high efficiency and simplicity, March test algorithms are widely used to detect the faults in memories[5-6]. Memory built-in self-test (MBIST) has become a standard industry practice in testing numerous embedded memories[7-8]. Based on efficient circuits and algorithms, MBIST effectively detects defects in random access memory (RAM) and read-only memory (ROM) and generates multiple test vectors, each of which focuses on testing a particular circuit or error. However, because of the increasing size of embedded memories, the switching activity in the address sequence dramatically increases when complex test vectors are loaded, which leads to an extreme dynamic power consumption[9]. In addition, the excessive power dissipation in test mode considerably influences the reliability of SoCs[10-11]. Hence, reducing the test power consumption becomes an imperative concern in the process of MBIST, in which it makes sense to design an address generator with low toggle rate in test mode. Over the years, a series of solutions have been proposed for low power MBIST. Since linear feedback shift register (LFSR) can produce a pseudo-random test pattern with a small area overhead, it is widely used to generate the address sequence in MBIST[12-13]. A complete LFSR with up/down control signals was proposed in Ref. [5]. While this LFSR can generate complete addresses in $2^n$ up and $2^n$ down sequences, it fails to reduce the transitions effectively for address generator in MBIST. Nourani et al. proposed a low-transition linear feedback shift register (LT-LFSR) to reduce the switching activity among patterns[14]. This method reduces the average and peak power of a circuit during the test, whereas the generator that adopts this method needs a longer sequence of test vectors to get high fault coverage. Vellingiri et al. combined LFSR with Bipartite LFSR and then proposed an improved LT-LFSR to deal with this

**Received date**: 2020-05-19

**Foundation items**: Fundamental Research Funds for the Central Universities (No. JUSRP51510); Primary Research & Development Plan of Jiangsu Province (No. BE2019003-2)

**Corresponding author**: GU Xiao-feng (xgu@jiangnan.edu.cn)

problem[15]. This method expresses relative superiority in reducing the number of switching activity between patterns without affecting the randomness. Wang et al. proposed a dual-speed LFSR (DS-LFSR)[16-17]. It consists of two LFSRs, a slow LFSR and a normal-speed LFSR, and effectively decreases the number of transitions during the test. Afterwards, there have been some researches adopting this method for the reduction of the number of transitions. Yang et al. split the LFSR into two LFSRs in order to generate a zero-set and one-set cube in the test cube[18]. An LFSR reseeding approach in Ref. [19] adopts dual-LFSR for test cubes. Test cubes in an LFSR reseeding scheme can generate proper values to cover don't-care bits and reduce the switching activity for low-power testing successfully. Based on the modified zero-one algorithm, Krishna et al. proposed an address generator consisting of two different clock signals and a blend of LFSR and a 2-bit pattern generator[20]. This method effectively decreases the switching activity between adjacent address sequences. However, it is not optimized for various address bus widths and lacks flexibility, as well as realizing a low test coverage.

In this study, a modified LFSR-based address generator suitable for the March test algorithm is proposed. We first obtain the most suitable partition of the address bus, and then divide the address generator into two optimized and reversible generation structures with two distinct clock signals. Finally, the proposed address generator has a significant reduction in switching activity with area overhead, which is illustrated in a 64 k×32 static random access memory (SRAM).

# 1 MBIST and power analysis

## 1.1 Structure of MBIST

As shown in Fig. 1, a typical structure of MBIST is mainly composed of a built-in self-test (BIST) controller and a data comparator. The BIST controller, consisting of a signal generator, a test vector generator, an address generator and so forth, is used to generate applicable test vectors, read commands, write commands, and analyze the outcomes. The address generator is used to generate a set of address sequences for detecting flaws in SRAM. When the start signal BIST_Test, the clock signal CLK, and the selection signal BIST_CS are valid, the BIST controller starts to test the memory, and meanwhile, the comparison between the test results and the expected results determines whether the memory has flaws or not.

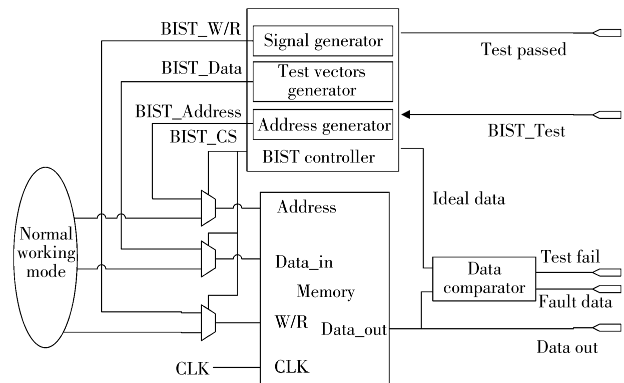


Fig. 1 Structure of MBIST

## 1.2 March algorithm and power analysis

March algorithms have different elements, such as $\Updownarrow(W_0)$, $\Uparrow(R_0, W_1, W_1, R_1, R_1, W_0)$, $\Downarrow(R_1, W_0)$, etc. Each March element consists of an addressing direction and a finite number of read or write operations. The special symbols "$\Uparrow$" and "$\Downarrow$" denote increasing and decreasing address orders, respectively; "$\Updownarrow$" is used to indicate the non-significant order of addresses; and $W_x$ or $R_x$ means that a logical value '$x$' will be written into or read from a memory cell.

Although the March algorithm has a superior test coverage, it is generally affected by the complexity. Because of some long and complex transitions during the detection process, the amount of switching activity increases sharply, which results in a high dynamic power consumption. Dynamic power consumption can be expressed by

$$P_{\text{avg}} = \alpha_{\text{T}} \times C_{\text{load}} \times V_{\text{dd}}^2 \times f_{\text{clk}}, \tag{1}$$

where $\alpha_{\text{T}}$ is related to toggle rate; and $C_{\text{load}}$, $V_{\text{dd}}$ and $f_{\text{clk}}$ denote total load capacitance, supply voltage and working frequency, respectively.

In theory, we can reduce the dynamic power consumption by adjusting the parameters in Eq. (1). However, it is not easy to scale down the power consumption by decreasing $C_{\text{load}}$, $V_{\text{dd}}$ and $f_{\text{clk}}$. Therefore, the decrease of $\alpha_{\text{T}}$ becomes an available way to reduce the power consumption. If the number of switching activities between adjacent address sequences decreases, the dynamic power consumption can be also dramatically reduced during the memory-testing process.

## 2 Improved LFSR address generator

Based on the conventional LFSR algorithm, the address generator shows a high toggle rate. Accordingly, we improve the traditional LFSR algorithm and design an address generator with a superior reduction in switching activity between adjacent address sequences. Firstly, the feedback structure of LFSR is altered to get the reversible LFSR. Secondly, we partition the reversible LFSR into an $H$-bit LFSR and an $L$-bit LFSR, and then work out the optimized partition ratio by finding the minimum value of transitions in the proposed LFSRs. After that, the clock signal H_LFSR_CLK controls the $H$-bit LFSR, the clock signal L_LFSR_CLK controls the $L$-bit LFSR, and H_LFSR_CLK is derived from L_LFSR_CLK. Finally, the two LFSRs controlled by the above clock signals will produce address sequences with a low toggle rate.

### 2.1 Design of reversible LFSR

The traditional LFSR cannot produce the complete address sequence because when all the flip-flop outputs are zero, the LFSR will maintain all-zero state (see Fig. 2). Thus, as shown in Fig. 3, some OR and NOR gates are inserted in the traditional LFSR to generate a full sequence. This structure is considered as the $k$-stage complete LFSR.

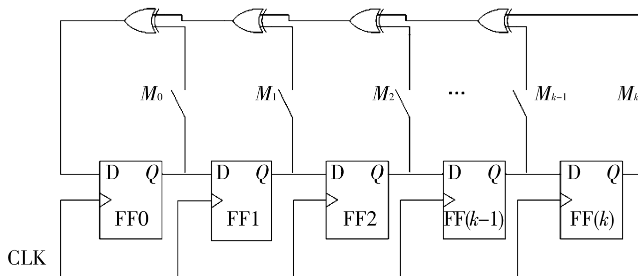


**Fig. 2 Structure of traditional LFSR**

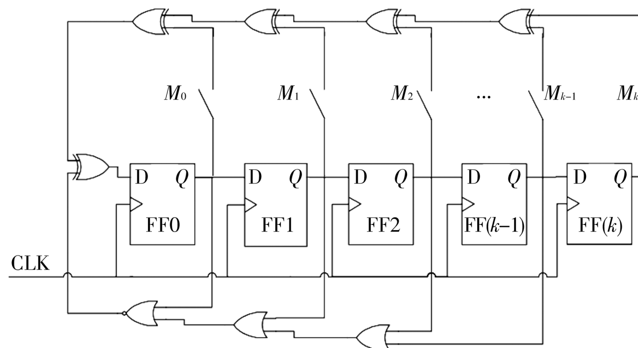


**Fig. 3 Structure of a $k$-stage complete LFSR**

For a $k$-stage complete LFSR, its primitive polynomials can be expressed as

$$H(X,k) = M_0X^0 + M_1X^1 + \cdots + M_kX^k, \quad (2)$$

$$G(X,k) = M_0X^k + M_1X^{k-1} + \cdots + M_kX^0, \quad (3)$$

where $M_i$ ($i = 0, 1, \cdots, k$) is the value of the corresponding flip-flop output, and the values of $M_0$ and $M_k$ are always "1". The difference between $H(X,k)$ and $G(X,k)$ is that the sequence orders of Eqs. (2) and (3) are reversed.

### 2.2 Optimized address partition

A $k$-bit LFSR generating all address in $2^k$ clock cycles will produce $2^{k-1}$ transitions at the output of each LFSR cell[12]. Thus, in the process of generating a complete $k$-bit pattern address sequence, a $k$-bit LFSR will switch $k\times 2^{k-1}$ times. Therefore, the number of output switching activities in an $H$-bit LFSR and an $L$-bit LFSR are $H\times 2^{H-1}$ and $L\times 2^{L-1}$, respectively. As mentioned above, we transform the $k$-bit LFSR into a blend of an $H$-bit and an $L$-bit LFSR, where $H+L=k$. When the high $H$-bit LFSR switches one time, the $L$-bit LFSR switches $L\times 2^{L-1}$ times. Thus, during the generation of a full address sequence, the number of transitions in the output bits of LFSR can be defined as

$$Y = H\times 2^{H-1} + 2^H \times L \times 2^{L-1}. \quad (4)$$

In order to find the minimum value of $Y$, we obtain Eq. (5) by taking the derivative of both sides of Eq. (4), namely

$$Y' = (H\times 2^{H-1} + 2^H \times (k-H)\times 2^{k-H-1})'. \quad (5)$$

Assuming that $Y'=0$, we have

$$2^k = 2^H(1 + H\ln 2). \quad (6)$$

Consequently, we can figure out the optimized address partition and the minimum switching activities by Eqs. (4) and (6). Table 1 shows the optimal partitions for various address bus widths. According to Table 1 and Eq. (4), we obtain a comparison of the output switching activities for different address generators using traditional LFSR, the combined LFSR in Ref. [20] and the proposed LFSR, respectively. The results are shown in Fig. 4.

**Table 1 Optimized partition for different address generators**

| Address bus width $N$ (bit) | $L$ (bit) | $H$ (bit) |
|---|---|---|
| 10—19 | 3 | $N-3$ |
| 20—24 | 4 | $N-4$ |

The number of the output switching activities of the proposed LFSR is less than those of the other two methods. As the address bus width of address generator increases, the improvement of the

proposed method becomes larger.

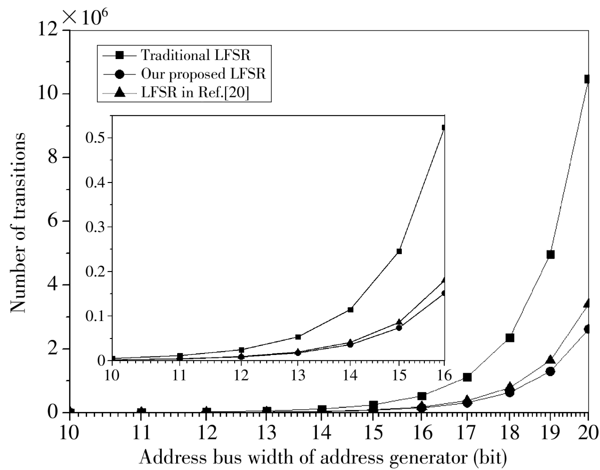


**Fig. 4 Change of the number of transitions generated by different size of address generators using different LFSRs**

## 2.3 Design of clock signal

To make the improved combined LFSR function efficiently, we design two distinct clock signals: H_LFSR_CLK and L_LFSR_CLK. H_LFSR_CLK is a divided clock from the clock L_LFSR_CLK, and the timing relationship between H_LFSR_CLK and L_LFSR_CLK is shown in Fig. 5. In an H_LFSR_CLK cycle, the maximum number of transitions in $L$-bit pattern is $2^L$.

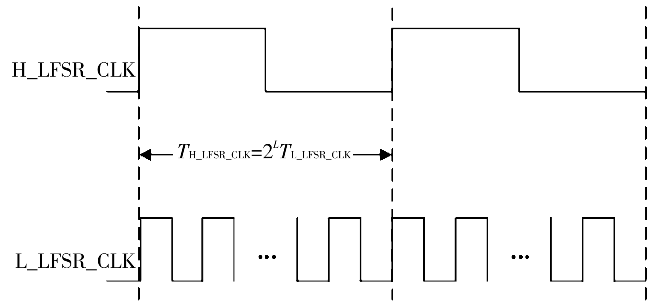


**Fig. 5 Timing relationship between L_LFSR_CLK and H_LFSR_CLK**

# 3 Experiment and analysis

Our proposed method is implemented on 64 k×32 SRAM, whose address bus width is 16 bits. According to Table 1, we split the 16-bit LFSR into a 13-bit LFSR and a 3-bit LFSR. Moreover, we add a select signal "updn" for the generation circuit to ensure that the address generator can generate two address sequences in reversed order. H_LFSR_CLK is obtained from L_LFSR_CLK utilizing a frequency divider. A circuit of address generator is designed as shown in Fig. 6.

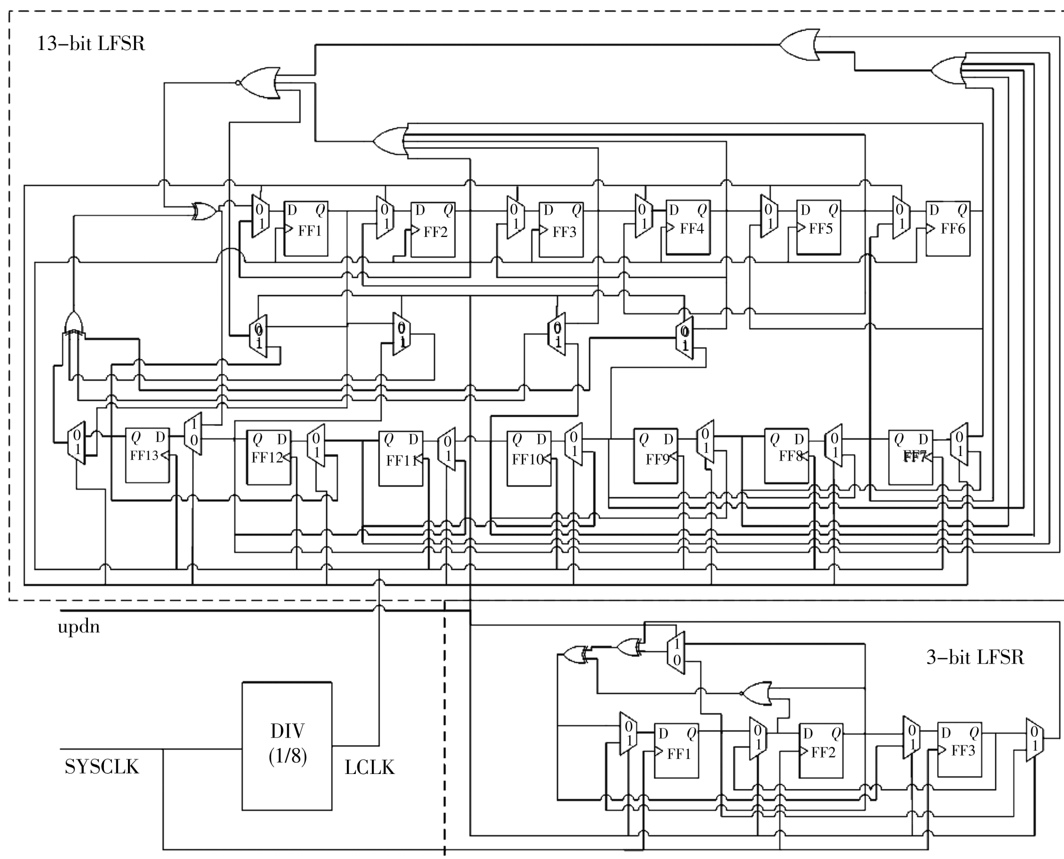


**Fig. 6 Design of proposed 64 k address generator**

We stimulate the proposed LFSR, traditional LFSR and a combined LFSR in Ref. [20] with Cadence NClaunch. Simulation waveforms of the proposed method, traditional LFSR and a combined LFSR in Ref. [20] are shown in Fig. 7. We can see that the proposed LFSR finishes generating a full address sequence including 65 536 addresses "FFFF, FFFE, FFFD, FFFA, …, 7FFC, 7FF8, 7FF9, 7FFB" (LFSR_13_3), and the combined LFSR produces a complete address sequence "FFFF, FFFE, FFF8, FFF9, …, 7FFF, 7FFE, FFFC, FFFD" (LFSR_14_2). However, the traditional LFSR totally generates 65 535 addresses "FFFF, FFD3, FF8B, FF3B, …, BFEE, 7FF1, FFE2, FFE9" (LFSR_tradition_16) without address "0000". In addition, we use three counters count_tradition, count_14_2, and count_13_3 to display the output switching activity of the 16-bit conventional LFSR, the combined LFSR in Ref. [20], and our improved LFSR, respectively. In Fig. 7, the count_tradition shows that the 16-bit conventional LFSR switches 524 288 (hexadecimal value is "80000h") times; the count_14_2 describes that the number of transitions in the combined LFSR is 180 244 ("2c000h"); the count_13_3 records that our improved LFSR only switches 151 522 ("25000h") times for producing a complete address sequence. Compared with the traditional LFSR and the combined LFSR in Ref. [20], the switching activity of our work decreases by 71.1% and 15.9%, respectively.

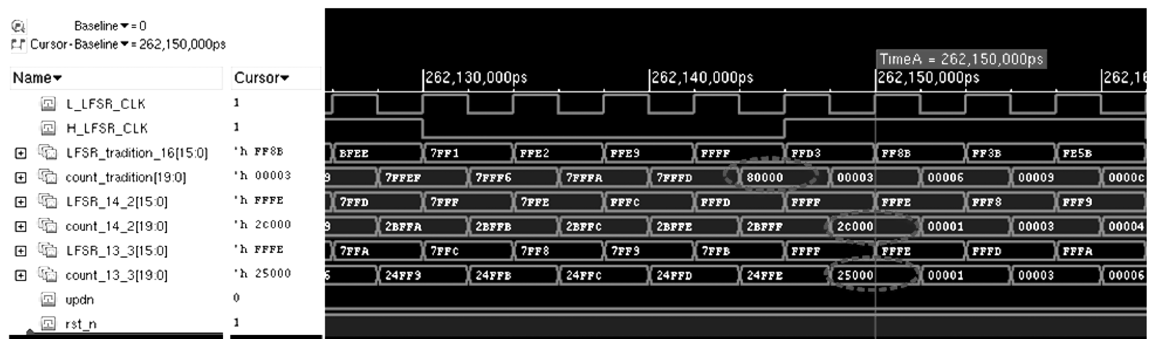


Fig. 7 Waveform of 64 k×32 address generator

The area overhead and power analysis are performed on 65-nm CMOS standard cell library using Synopsys Design Compiler and Prime Time. Table 2 shows the comparison of the power consumption and equivalent gate counts between the traditional LFSR, the LFSR in Ref. [20], and the proposed method.

**Table 2 Comparison of dynamic power consumption and equivalent gate count**

| Method | Power consumption (μW) | Equivalent gate count |
|---|---|---|
| Traditional LFSR | 30.13 | 89 |
| LFSR in Ref. [20] | 18.34 | 128 |
| Proposed method | 9.587 | 155 |

The address generator employing the traditional LFSR algorithm consumes 30.13 μW in dynamic power. The dynamic power dissipation in a combined LFSR[20] decreases to 18.34 μW. The proposed address generator in the depth of 64 k only consumes 9.587 μW, achieving 68.2% and 47.7% dynamic power saving compared to the 16-bit conventional LFSR and the combined LFSR, respectively. Furthermore, the proposed address generator circuit has a slight increase in the equivalent gate count. The equivalent gate count for the proposed address generator is about 155 equivalent gates, while the equivalent gate counts for the traditional LFSR and the combined LFSR are 89 and 128, respectively.

## 4 Conclusion

This paper presents an optimized address generator based on LFSR for low power MBIST. The LFSR can be split into two LFSRs with a series of optimized address partitions and drastically decreases the switching activity during the generation of a complete address sequence. As a result, the proposed address generator significantly reduces the dynamic power. Experimental results in a 16-bit address generator demonstrate a significant reduction in power consumption via the proposed method.

## References

[1] Querbach B, Khanna R, Puligundla S, et al. Architecture of a reusable BIST engine for detection and auto correction of memory failures and for IO debug, validation, link training, and power optimization on 14-nm SoC. IEEE Design & Test, 2016, 33(1): 59-67.

[2] Mukherjee N, Pogiel A, Rajski J, et al. BIST-based fault diagnosis for read-only memories. IEEE Transactions on

Computer-Aided Design of Integrated Circuits and Systems, 2011, 30(7): 1072-1085.

[3] Naeem A, Jantsch A, Lu Z. Scalability analysis of memory consistency models in NoC-based distributed shared memory SoCs. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2013, 32(5): 760-773.

[4] Goh S H, Chan Y H, Lin Z, et al. Concurrent built-in self-testing under the constraint of shared test resources and its test time reduction. Integration the VLSI Journal, 2017, 59: 198-205.

[5] Harutyunyan G, Shoukourian S, Vardanian V, et al. An effective solution for building memory BIST infrastructure based on fault periodicity. In: Proceedings of IEEE 31st VLSI Test Symposium (VTS), Berkeley, CA, USA, 2013: 1-6.

[6] Goor A J V D, Hamdioui S, Kukner H. Generic, orthogonal and low-cost march element based memory BIST. In: Proceedings of IEEE International Test Conference(ITC), Anaheim, CA, USA, 2012: 1-10.

[7] Park Y, Park J, Han T, et al. An effective programmable memory BIST for embedded memory. IEICE Transactions on Information and Systems, 2009, 92(12): 2508.

[8] Awad A N, Abu-Issa A S. Low power address generator for memory built-in self test. The Research Bulletin of Jordan ACM, 2011, 2(3): 52-56.

[9] Wang J, Hoefler A, Calhoun B H. An enhanced canary-based system with BIST for SRAM standby power reduction. IEEE Transactions on Very Large Scale Integration Systems, 2011, 19(5): 909-914.

[10] Benini L, Bogliolo A, Micheli G D. A survey of design techniques for system-level dynamic power management. readings in hardware/software co-design, 2002, 8(3): 231-248.

[11] Kim Y, Jang J, Son H, et al. Pattern mapping method for low power BIST based on transition freezing method. IEICE Transactions on Information & Systems, 2010, 93-D(3): 643-646.

[12] Abu-Issa A S, Quigley S F. Bit-swapping LFSR for low-power BIST. Electronics Letters, 2008, 44(6): 401-402.

[13] Wang W L, Lee K J. A complete memory address generator for scan based March algorithms. In: Proceedings of IEEE International Workshop on Memory Technology, Design, and Testing, Taipei, Taiwan, China, 2005: 83.

[14] Nourani M, Tehranipoor M, Ahmed N. Low-transition test pattern generation for BIST-based applications. IEEE Transactions on Computers, 2008, 57(3): 303.

[15] Vellingiri G, Jayabalan R. An improved low transition test pattern generator for low power applications. Design Automation for Embedded Systems, 2017, 21(7): 1-17.

[16] Wang S, Gupta S K. DS-LFSR: A new BIST TPG for low heat dissipation. In: Proceedings of International Test Conference, Washington, DC, USA, 1997: 848.

[17] Wang S, Gupta S K. DS-LFSR: A BIST TPG for low switching activity. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2002, 21(7): 842-851.

[18] Yang M H, Kim Y, Park Y, et al. Deterministic built-in self-test using split linear feedback shift register reseeding for low-power testing. IET Computers and Digital Techniques, 2007, 1(4): 369-376.

[19] Ying J C, Tseng W D, Tsai W J. Asymmetry dual-LFSR reseeding for low power BIST. Integration, the VLSI Journal, 2018, 60: 272-276.

[20] Krishna K M, Sailaja M. Low power memory built in self test address generator using clock controlled linear feedback shift registers. Journal of Electronic Testing, 2014, 30(1): 77-85.

# 基于改进型 LFSR 的低功耗 MBIST 地址生成器

虞致国[1,2]，李青青[1,2]，冯　洋[1,2]，顾晓峰[1,2]

(1. 江南大学 电子工程系，江苏 无锡 214122；
2. 江南大学 物联网技术应用教育部工程研究中心，江苏 无锡 214122)

**摘　要：** 存储器进行内建自测试(Mernory built-in self-test，MBLST)时，其功耗远远高于普通模式下的功耗，致使电路易损坏并降低了芯片成品率。针对上述问题，提出了一种改进的线性反馈移位寄存器，可在存储器内建自测试的地址序列生成过程中大幅降低翻转率。首先基于优化的地址分割比生成两个优化的、可逆的地址生成器，随后利用时钟信号分别控制两个地址生成电路的时序关系，最后对 64 k×32 SRAM 的 MBIST 的地址生成器进行了仿真验证。结果表明，改进的结构与传统的线性反馈移位寄存器(Linear feedback shift register，LFSR)的地址生成结构相比，地址序列间的翻转率和动态功耗分别降低了 71.1%和 68.2%，同时具有面积成本低、速度快等特点。

**关键词：** 地址序列；线性反馈移位寄存器；存储器内建自测试；地址生成器；翻转率

**引用格式：** YU Zhi-guo, Li Qing-qing, FENG Yang, et al. An LFSR-based address generator using optimized address partition for low power memory BIST. Journal of Measurement Science and Instrumentation, 2020, 11(3): 205-210. [doi: 10.3969/j.issn.1674-8042.2020.03.001]