Design and Analysis of Low-Transition Address Generator

Sivasankaran Saravanan[1](✉), Mikias Hailu[2], G. Mohammed Gouse[3], Mohan Lavanya[4], and R. Vijaysai[4]

[1] College of Engineering, Debre Berhan University, Debre Berhan, Ethiopia
saran@dbu.edu.et

[2] Electrical and Computer Engineering Department, Debre Berhan University, Debre Berhan, Ethiopia
hailumikias@dbu.edu.et

[3] College of Computing, Debre Berhan University, Debre Berhan, Ethiopia
galety.143@dbu.edu.et

[4] School of Computing, SASTRA Deemed University, Thanjavur, Tamilnadu, India
m_lavanyass@ict.sastra.edu, vijaysai@it.sastra.edu

Abstract. In high-speed Nano-scale VLSI designs, memory plays a vital role of operation. Built-In Self-Test (BIST) for memory is an essential element of the system-on-chip (SoC). Investigating memory with low power techniques have been emerging in the market. Address generators to access memory cores consecutively should have low transition. This paper, attempted to put forward a proposed architecture of address generator with low-transition. In this novel technique, the address generator is constructed by a blend of modulo-counter and binary to gray code convertor with a bit-reversal block. Efficient employment of this architecture has cut-down the switching activity considerably. This proposed work compared the switching activity with conventional Linear Feedback Shift Register (LFSR), Bit-Swapping LFSR (BS-LFSR) and gray-code generator. Simulated and synthesized of the proposed architecture was done by Xilinx tool. The final result shows more than 95% reduction on dynamic power consumption related to the traditional LFSR.

Keywords: Address generator · Gray code counter · Memory built in self-test (MBIST) · Low power switching · Linear Feedback Shift Register (LFSR)

1 Introduction

Low-power designs, especially microprocessors, have received a large amount of attention recently in portable and wireless related applications. Also, even the highest performance design power has become an issue. In high performance devices extremely high frequencies are attained, there by leading to greater power consumption and higher heat dissipation. To overcome this problem, heat sinks are used, resulting in higher costs and potential reliability problems. It is evident that a system consumes power twice more in testing mode when compared to usual mode. This is because of

F. A. Zimale et al. (Eds.): ICAST 2018, LNICST 274, pp. 239–247, 2019.
https://doi.org/10.1007/978-3-030-15357-1_19

applying parallel testing, wherein multiple embedded memories will be tested simultaneously while few memories are accessed during the normal mode. If the power consumed during the testing mode exceeds the power constraint of the chip, then the chip may become structurally degraded and may be damaged [1].

According to the state of system operation, power dissipation is classified into two major types. They are Static and Dynamic power. In static power, it is the power consumed while there is no circuit activity. In dynamic power, it is the power consumed while the inputs are in change or active. It is the major component of the total chip power consumption. Short- circuit and leakage power are the other two minor components and they can be minimized only at the stage of fabrication process. Charging and discharging of capacitive loads at the output of gates result in dynamic power consumption. Wiring capacitance, junction capacitance, and the input or gate capacitance of the fan-out gates contribute capacitive loads that results dynamic power. Hence, this paper considers dynamic power dissipation, which is determined by the equation

$P_{avg} = \alpha T C_{load} V_{dd}^2 f_{clk}$
αT = Switching activity f actor of gate
C_{load} = Total load capacitance
V_{dd} = Supply voltage
f_{clk} = Operating frequency

In the above equation, the average power is in direct proportion to the T. Therefore, the power dissipation can be scaled down by managing the switching activity when the system is in testing mode. The progress in submicron manufacturing technology and SoC design methodology has resulted in abnormal number of cores, particularly the memory cores included in a single chip. It is forecasted that by the year of 2014, memory cores may use 94% area of a typical SOC [2]. Memory thus plays a significant role in SoC based design. As the prospect of memory flaws is largely contrasted with other kind of defects in a circuit, that are necessary for testing of memory. Anyhow, on account of the existence of a minimum number of I/O pins in a circuit, BIST for Memory (MBIST) is employed as a key to this vexed issue [3]. Thus, this paper focuses on low transition address generator to reduce unnecessary switching in memory access process.

Hierarchy of address generators start with a normal binary up-down counter that produce sequence of addresses. When area and power are concerned, binary up-down counters occupy more area and also consumes more power. It is significant that the address generator plays an imperative role in MBIST. The blueprint of address generator is the foremost intricate problem in MBIST in view of its mammoth area and speed limit [4]. Reducing the switching activity in the address decoder using the Single Bit Change (SBC) instead of using the normal counter was proposed in [5]. This technique minimizes the switching activity in the address bus but it entails large overhead in the hardware area since a modified counter is used. Another technique was based on generating new march sequences with low switching activity by reordering the test sequences using genetic algorithm in [6]. This reordering reduces the switching activity in the data bus whereas it remains the same in the address bus.

To overcome the area problem, linear feedback shift registers (LFSRs) were introduced. It succeeded in scaling down the area occupied by address generators. But it failed to answer the power consumption caused by switching activity. Several algorithms were later introduced to cut down the switching in LFSR. In Bit-swapping LFSR [7] comprise traditional LFSR and 2:1 multiplexers. To overcome power consumption problem, diverse kinds of address generators are found employed for MBIST. In a programmable MBIST, two counters and multiplexer combination are employed as an address generator [8, 20]. A new reseeding technique with a considerable scan power reduction was proposed earlier [18]. The reseeding is applied on general LFSR and modified LFSR in two ways. These generated patterns are sent to a XOR network which will generate an output. A new LFSR reseeding technique for efficient reduction of test pattern was proposed in [19]. A new encoding technique to reduce the size of the test data was proposed in this study. Size of the test data was reduced by clock in LFSR which is in the state of inactive for several clock cycles after the seed is given to the input. Thus, reduction in test data volume is achieved by storing the data only when the clock is in active state. All the remaining test vectors was derived with in the reduced clock.

Linear Feedback Shift Register (LFSR) which is a combination of Bit Swapping LFSR (BS-LFSR) and Dual Speed LFSR (DS-LFSR) has been proposed early [10, 17, 21]. In this paper the modified Dual Speed LFSR (MDSLFSR) consists of two BS-LFSRs, one is the slow speed BS-LFSR and another is the normal speed BS-LFSR each has independent clock rates. The modified DS LFSR lowers the transition density at the input side. Thus, it reduces overall switching activity. In SoC environment, significant changes in testing methods are to be done for memory arrays. Built-In-Self-Test for optimized memory repair analyzer which works with optimal repair rate for memory arrays in [11]. Single test is required for this method even for worst case. The Must-Repair-Analysis (MRA) technique is done on fly during test, it stores faulty addresses and final analysis is done to find a solution to eliminate the analyzed faults. It executes an efficient redundancy analysis algorithm to generate repair solutions. A new method for generating configurations for application dependent testing of a SRAM-based FPGA interconnect was proposed in [12]. This method connects an activating input to multiple nets, thus generating activating test vectors for detecting stuck-at open, and bridging faults. A new technique for reduced switching activity in the address bus when testing SRAM of personal devices has been proposed in previous works [13]. Even though these algorithms deal with power minimization, the power reduction is an endless requirement of modern VLSI world. In certain other cases, Binary up-down counter and Gray code Counters are employed as address generators [9]. In paper [22] shows improved low transition test pattern generator method and targeted to low power application. With a concern to still decrease the switching activity, in our work, we have developed and executed a new address generator which uses bit reversal technique along with a modulo counter and grey code convertor.

2 Proposed Architecture

Linear Feedback Shift Registers (LFSR) or counters generally employed as address generators for memory locations, need to be investigated for flaws. The traditional LFSR is incapable of producing the entire zero patterns because when all the flip-flop output becomes zero then the XOR output is also zero hence the feedbacks to the 1st flip-flop input are also zero. Hence the LFSR becomes stuck at zero stage. To overcome this problem, a complete memory address generator (CLFSR) [15] was proposed in previous works. But power consumption is considerably high when traditional architectures are employed as address generators. To overcome the power consumption problem, a novel architecture is proposed in Fig. 1.

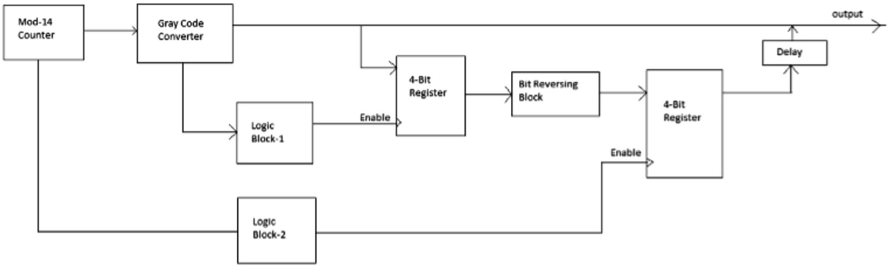


Fig. 1. Proposed block diagram

The architecture of address generator proposed in this paper is a blend of modulo counter and gray code convertor with a blend of bit reversal block. This paper focuses mainly on utilizing the reversible bit patterns. That is when we look at the patterns, we come across certain patterns which can be reversible. For example, 0000000001 is a pattern which is reversible. When certain pattern is reversed, the other address or pattern, 1000000000 can be obtained. Thus, an n-bit length pattern a (0 → n − 1) is said to be reversible if

$$a\,(0 \rightarrow n - 1) \neq a\,(n - 1 \rightarrow 0)$$

There exist certain other patterns which cannot be reversed. For example, 1000000001 or 1001001001 are some patterns which results in the same pattern even if we reverse them. Thus, an n-bit length pattern a (0 → n − 1) is said to be irreversible if

$$a\,(0 \rightarrow n - 1) = a\,(n - 1 \rightarrow 0)$$

So, in this paper we tried to make use of one such pattern which is the final outcome from gray code counter. In our observation, the final sequence of a gray code can be generated from the respective reversible pattern which is already generated prior to the final sequence. In this proposed architecture, two logic blocks along with the counter and gray code generator are used.

Algorithm 1 Logic Block-1

```
Require: Enable = $(G0 \oplus G3) \wedge (G1 \downarrow G2)$
Repeat while B $[3:0] \neq 1111$ do
    if Enable=1 then
        G $[3:0] \rightarrow$ G$[0:3]$
    else G$[3:0] \rightarrow$ G$[3:0]$
    end if
end while
until B$[3:0] = 1110$
```

Algorithm 2 Logic Block-2

```
Require: Enable = $(B0 \wedge B1 \wedge B2) \wedge (-B3)$
Repeat while B$[3:0] \neq 1110$ do
    if Enable=1 then
        G[0:3] delayed OUTPUTBUS
    else G$[3:0] \rightarrow$ OUTPUTBUS
    end if
end while
until G$[3:0] = 1000$
```

The logic block-1 is used to detect the reversible pattern from which the reversed sequence can be generated. The logic block-1 for 4 bit, is designed in such a way to execute the logic as shown in the Algorithm 1. Thus, the detected reversible sequence is given to bit reversing block for reversing the sequence to generate new bit sequence. This reversed sequence has to be inserted in the respective place that is at the end of modulo sequence. The end sequence of modulo gray code counter will be 1110. We use logic block-2 (Algorithm 2) to detect this sequence. The reversed bit stream has to be inserted in output with some delay when the logic block-2 detects the appropriate position.

3 Results and Comparison

The complete block diagram embedded with the two logic blocks included is shown in the Fig. 1. This block diagram consists of a modulo gray code counter which is a combination of modulo counter and gray code converter. It can be implemented for any size of addresses with respective changes in logic blocks. The test outcomes of the proposed method are furnished below. The novel design is simulated and synthesized in Xilinx tool. The simulation-based outcomes and switching activity of the parallel existing methods are contrasted below. The Table 1 furnish comparison between total and dynamic power consumption in the recent methodologies for a 4-bit address. Thus, the proposed technique offered 98.19% saving of dynamic power and 92.28% saving of total power consumption, which offers the low power design of address generator at the cost of 0.031 W of power.

Table 1. Comparison between switching activity with various method

Address generator	Total dynamic power (W)	Saving (%)	Total power (W)	Saving (%)
LFSR [7]	0.222	–	0.402	–
BS-LFSR [7]	0.199	10.36	0.378	5.97
DC-LFSR [14]	0.012	94.59	0.179	55.47
Proposed method	0.004	98.19	0.031	92.28

The Table 2 furnish the details of bit transition and comparison with novel technique. By examining both the tables we can find that the switching activity undergoes considerable alteration which is encouraging outcome to achieve low power design of address generator.

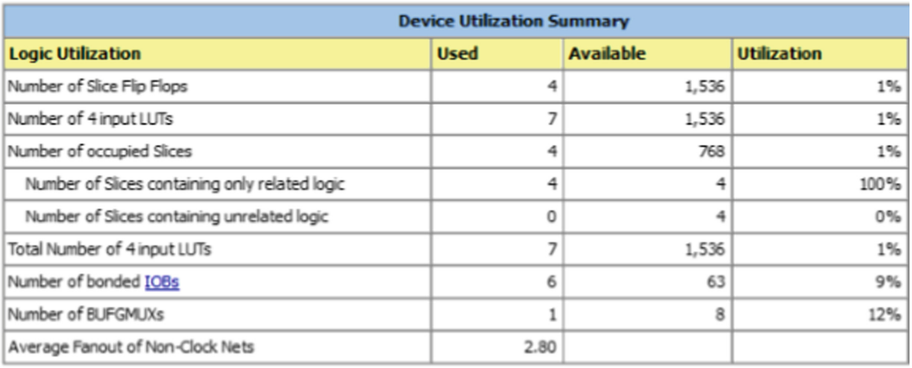
Table 2. Comparison of bit-transition with proposed method

LFSR [7]	BS-LFSR [7]	Gray code [9]	Proposed
1111	1111	0000	0000
0101	0101	0001	0001
1011	1011	0011	0011
0111	0111	0010	0010
1111	1111	0110	0110
1110	1110	0111	0111
1100	1100	0101	0101
1000	0100	0100	0100
0001	0001	1100	1100
0010	0010	1101	1101
0100	1000	1111	1111
1001	1001	1110	1110
0011	0011	1010	1010
0110	1010	1011	1011
1101	1101	1001	1001
1010	0110	1000	
Bit transition values in each weight			
(8)(7)(8)(7)	(9)(4)(8)(7)	(1)(2)(4)(8)	(1)(2)(4)(7)

The Table 3 furnish the switching activity of proposed technique and percentage of saving with the conventional address generator for bit length N = 5 and N = 10 in comparison with several existing methods. The simulation resulted in reduction of switching activity up to 64.70% for a N = 5-bit address generator and 80.08% reduction in N = 10-bit address compared to the conventional LFSR. Here by it is evident that this value may increase when we go for higher bit length address generators. The comparison is also made with the existing methodology [14, 16], which resulted in almost 10% decrease in switching activity. Figure 2 shows Xilinx simulation report of device utilization summary. Figure 3 shows power report of proposed method in Xilinx simulation.

Table 3. Comparison between power consumption

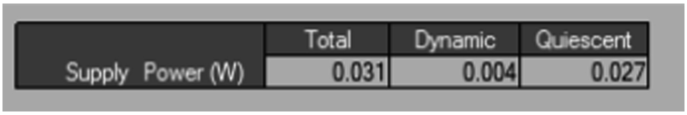
Method	Switching activity			
	Bit length N = 5	Saving (%)	Bit length N = 10	Saving (%)
LFSR [14]	85	–	5130	–
BS-LFSR [7]	69	19	4106	20
DS-LFSR [17]	70	18	2904	43
Bipartite LFSR [16]	45	47	2462	52
BS-DS LFSR [14]	64	25	2376	54
DC LFSR [14]	36	57.65	1783	65.24
Proposed method	30	64.70	1022	80.08



Device Utilization Summary

Logic Utilization	Used	Available	Utilization
Number of Slice Flip Flops	4	1,536	1%
Number of 4 input LUTs	7	1,536	1%
Number of occupied Slices	4	768	1%
Number of Slices containing only related logic	4	4	100%
Number of Slices containing unrelated logic	0	4	0%
Total Number of 4 input LUTs	7	1,536	1%
Number of bonded IOBs	6	63	9%
Number of BUFGMUXs	1	8	12%
Average Fanout of Non-Clock Nets	2.80		

Fig. 2. Device utilization summary from Xilinx simulation



	Total	Dynamic	Quiescent
Supply Power (W)	0.031	0.004	0.027

Fig. 3. Xilinx simulation power report

4 Conclusion

This paper attempted to introduce an efficient VLSI architecture for a low power address generator based on modulo counter which is having low dynamic power. It is achieved by decreasing the switching activity of the output address provoked. In this architecture, address generator consumes less area and less power consumption. Tradeoff between power and area is minimal in proportion. Proposed method's dynamic power dissipation reduces 90% when compared to traditional LFSR and BS-LFSR. It also 20% superior when compared to conventional gray code generator. The proposed address generator can be employed for MBIST by devising an appropriate MBIST controller. With the proficient employment of this Low power address generator, the entire MBIST unit can be adapted to taste the test patterns with less power consumption.

References

1. Abuissa, A.S., Quigleyr, S.F.: LTPRPG: power minimization technique for test-per-scan BIST. In: International Conference on Design and Technology of Integrated Systems in Nanoscale Era, pp. 1–5 (2008)
2. Marinissen, E.J., Prince, B., Keitel-Schulz, D., Zorian, Y.: Challenges in embedded memory design and test. In: Proceedings of Design Automation Test in Europe, vol. 52, pp. 722–727 (2005)
3. Noor, N.Q., Yusof, Y., Sparon, A.: Low area FSM-based memory BIST for synchronous SRAM. In: Proceedings of the International Colloquium of Signal Processing and Its Application, pp. 409–412 (2009)
4. Van de Goor A.D.J., Kukner, H., Hamdioui, S.: Optimizing memory BIST address generator implementations. In: 6th International Conference on Design & Technology of Integrated Systems in Nanoscale Era, pp. 1–6 (2011)
5. Wang, L.T., Stroud, C.E., Toubam, N.A.: System On Chip Test Architectures, pp. 308–339. Morgan Kaufmann, Los Altos (2008)
6. Gayathri, C.V., Kayalvizhi, N., Malligadevi, M.: Generation of new march tests with low test power and high fault coverage by test sequence reordering using genetic algorithm. In: International Conference on Advances in Recent Technologies in Communication and Computing, pp. 699–703 (2009)
7. Reddy, C.R., Zilani, S., Sumalatha, V.: Low power, low-transition random pattern generator. Int. J. Eng. Res. Technol. (IJERT) **1**(5), 1–6 (2012)
8. Park, Y., Park, J., Han, T., Kang, S.: An effective programmable memory BIST for embedded memory. IEICE Trans. Inf. Syst. **92**(12), 808–818 (2009)
9. Yarmolik, S.V., Yarmolik, V.N.: Modified gray and counter sequences for memory test address generation. In: Proceedings of International Conference, pp. 572–576 (2006)
10. Chandrakala, S., Banupriya, C.: A low power built in repair analyzer for word-oriented memories with optimal repair rate. In: Green Computing Communication and Electrical Engineering (ICGCCEE), pp. 1–5 (2014)
11. Ohler, P., Bosio, A., Di Natale, G., Hellebrand, S.: A modular memory BIST for optimized memory repair. In: 14th IEEE International on-line Testing (2008)
12. Kumar, T.N., Lombardi, F.: A novel heuristic method for application-dependent testing of a SRAM-based FPGA interconnect. IEEE Trans. Comput. **62**(1), 163–172 (2013)

13. Awad, A.N., Abu-Issa, A.S.: Low power address generator for memory built-in self-test. Res. Bull. Jordan ACM **III**(II), 52–56 (2007)
14. Krishna, K.M., Sailaja, M.: Low power memory built in self-test address generator using clock controlled linear feedback shift registers. J. Electron. Test. **30**, 77–85 (2014)
15. Wang, W.-L., Lee, K.J.: A complete memory address generator for scan-based march algorithms. In: Proceedings of the I.E. International Workshop on Memory Technology, Design, and Testing (MTDT05), p. 83–88 (2005)
16. Tehranipoor, M., Nourani, M., Ahmed, N.: Low transition LFSR for BIST-based applications. In: Proceedings of the 14th Asian Test Symposium, pp. 138–143 (2005)
17. Wang, S., Gupta, S.K.: DS-LFSR: a BIST TPG for Low switching activity. IEEE Trans. Comput. Aided Design Integr. Circuit Syst. **21**(7), 842–851 (2002)
18. Sowmiya, G., Premalatha, P., Rajaram, A., Saravanan, S., Vijay Sai, R.: Design and analysis of scan power reduction based on linear feedback shift register reseeding. In: IEEE Conference on Information and Communication Technologies, pp. 638–641 (2013)
19. Saravanan, S., Upadhyay, H.N.: Effective LFSR reseeding technique for achieving reduced test pattern. Res. J. Appl. Sci. Eng. Technol. **4**(22), 4783–4786 (2012)
20. Kumar, S., Manimegalai, R.: Efficient memory built in self-test address generator implementation. Int. J. Appl. Eng. Res. **10**(7), 16797–16813 (2015)
21. Hussain, S., Priya, P.: Test pattern generator (TPG) for low power logic built in self-test (BIST). Int. J. Adv. Res. Electr. Electron. Instrum. Eng. **2**(4), 1634–1640 (2013)
22. Vellingiri, G., Jayabalan, R.: An Improved low transition test pattern generator for low power application. Design Autom. Embedded Syst. **21**(3–4), 247–263 (2017)