Low Power Address Generator for Memory Built-In Self Test

Ahmed N.Awad & Abdallatif S.Abu-Issa
Faculty of Information Technology
Birzeit University-Palestine
Ahmed.n.Awwad @gmail.com, aabu_issa@yahoo.com

Abstract- Memory is one of the basic computer components that is intensively accessed. Therefore, it is more likely to be affected by manufacturing faults rather than other components in the System on Chip (SoC). Memory Built-in Self Test (MBIST) is the most commonly used to test embedded memories. Although many algorithms were developed for MBIST, only few of these techniques focus on reducing the test power which plays an important role in evaluating the effectiveness of the test. This paper deals with reducing the switching activity in the address bus when testing SRAM of personal devices. The MBIST architecture was programmed using VHDL and then five address generators were simulated using Xilinx ISE tools and compared with each others in terms of their switching activity which is proportional to the test power.

I. Introduction

During manufacturing and the application, memory goes through the testing phase in which it is tested for physical faults that may affect the functionality of the design. In Memory Built-in Self Test (MBIST), embedded memories can be tested without any communication with the external world [1]. With the advances in VLSI technology, more and more contents are integrated together in System on Chip (SoS), embedded memories can be considered as the densest circuitry in the chip, and it is expected that by 2014, around 94% of the silicon area in SoC will be occupied by embedded memories [2] [3], thus, using MBIST will be more efficient than using external testing which could be useful for standalone memories, not embedded ones. The MBIST system is illustrated in Figure 1.

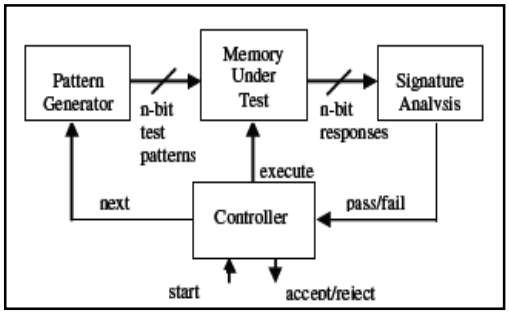

Figure 1: MBIST architecture [4]

ISSN: 2078-7952 (print); 2078-7960 (online)

With the advances in semiconductor technology, the power becomes one of the important issues that should be taken into consideration. The test power is a principal contributor for this power dissipation. Researches show that the consumed power during the testing mode could be twice the power consumed in the normal mode [5]. This is because of 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 subjected to structural degradation and may be damaged [6].

VLSI circuits are based on Complementary Metal Oxide Semiconductor (CMOS) technology in which dynamic power is the dominant source of power dissipation. The dynamic power is related to the number of Switching Activities (SA) during testing.. For this reason, reducing the switching activity during testing is a basic technique for power saving.

Many techniques were developed for memory testing, such as Zero-One, Walking 1/0, checker board, March sequences and others [7]. Unfortunately, only few techniques focus on reducing the test power. A good technique was based on reducing the switching activity in the address decoder using the Single Bit Change (SBC) instead of using the normal counter [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 [8]. This reordering reduces the switching activity in the data bus whereas it remains the same in the address bus. Another good approach is based on reducing the peak power in the SoC when multiple memories are being tested in parallel, in this approach; a skew is applied in order to avoid simultaneous write operations, since the write operation draws about 10 times more current than a read operation [9]. Using this approach will not affect the power consumed in each individual memory. Reducing the pre-charging activity in SRAM during testing was another method used for test power saving [10].

Usually in personal mobile devices such as mobile phones and digital cameras, the embedded memories are tested for stuck at faults since these faults are the most common specially during online testing. Thus, the test power has to be reduced since those small devices have low power envelop and they may be damaged if the power consumed exceeds the power constraint. In other applications, such as military applications, the memory has to be tested intensively for several types of faults. In some cases, the switching activity in the address decoder has to be maximized such as delay related faults.

This paper focuses on reducing the switching activity in the address decoder during testing SRAM for stuck-at faults. Five

address generators were simulated and compared in their switching activity. All these simulations were performed using **Xilinx ISE tools.**

This paper is organized as following: Section II describes the main SRAM faults. Section III summarizes the proposed design and the types of address generators used. In Section IV, the simulation results are shown and analyzed. Finally, the conclusions and the future work are proposed in section V.

II. Memory Faults

In this paper, bit-oriented memory (in which each memory location consists of 1 cell) is considered. Since the memory is intensively accessed through read and write operations, it is more likely to be affected by manufacturing faults. The following is a brief description about the main types of these faults [7].

A. Stuck-at Faults: This is the most common type of faults in memory. The memory cell itself may be stuck at 0 or stuck at 1. Figure 2 shows the state diagram for this type of faults.

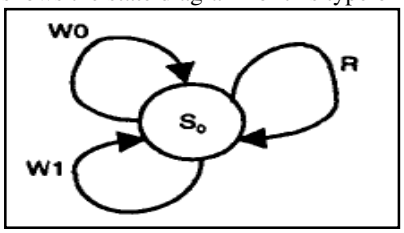

Figure 2: Stuck-at 0 cell

B. Address Decoder Fault (AF): This type of faults can be classified under the stuck-at faults since one of the nodes in the address decoder may stuck at 0 or 1 leading to accessing wrong address, no address, or multiple addresses. Figure 3 illustrates this type of faults.

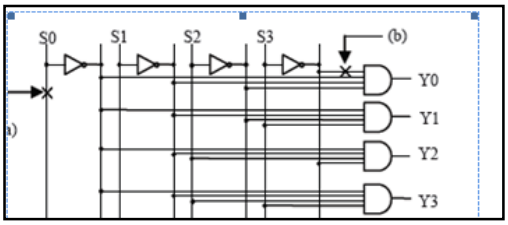

Figure 3: Address Decoder Fault [11]

C. Transition Fault (TF): This fault causes the memory cell state to go into one direction only. For example, , if the cell contains 0 and 1 has been written to it, then it cannot be written back to 0 and vice-versa.

D. Coupling Fault (CF): In this fault, transition in one cell may affect its neighboring cell and cause it to go into erroneous state. The cell that causes the coupling fault in its neighbor is called the aggressor whereas the affected cell is called the victim cell**.**

III. Design and Address Generators

In MBIST, the BIST engine generates the testing patterns that are applied to the Memory Under Testing (MUT); each pattern consists of the address that has to be accessed, the data that will be written to the memory, and the control signal that determines whether a read or a write operation is performed. Figure 4 shows the design for MBIST used in this paper.

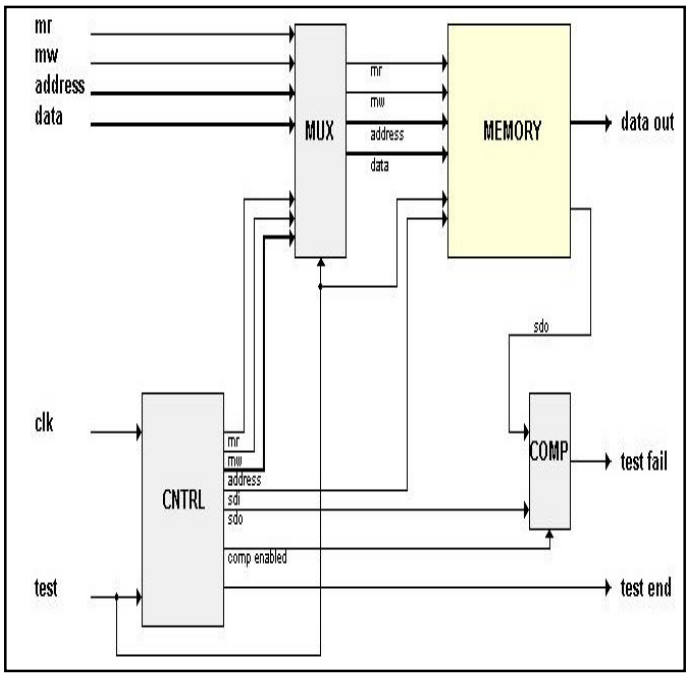

Figure 4: MBIST Used Design

As shown in figure 4, the BIST engine (controller) provides mr and mw signal to determine whether the operation is read or write operation. It also generates the address that will be accessed, and the data (sdi) that will be written to the memory in case of write operation. If the operation is read, then the excepted data to be read (sdo) will be sent to the comparator that compares this data with the one read from the memory. The Mux has the test signal as its selection line in order to determine whether the system is in the normal mode or in the testing mode. Finally, the comparator compares the data read from the memory with the expected data, if these two values are different, then the test fail signal will be activated. The test continues until the test end signal is activated by the BIST engine. In [12] more details are provided about building MBIST systems.

Testing memory for stuck-at faults is usually performed using Zero-One testing pattern. This pattern consists of the following operations [7]

↕ (W0);↕ (R0); ↕ (W1); ↕ (R1).

During this test, zero is written to all memory locations then it is read from these locations in order to detect any stuck-at one cell. Then, the same operations are performed with writing 1 in order to detect the stuck-at zero cells. The symbol ↕ indicates that the order of addresses that will be accessed is not important.

Zero-One testing algorithm has a significant problem; it contains high switching activity in address bus since nowadays, large address space memories (such as 30-bit address bus) are used and going through all these locations

four times means increasing the switching activity in the address bus that will increase the testing power consumption.

A better technique to be used is to perform the four operations for each memory location, and then proceeding to the next location. This will reduce the switching activity in the address decoder effectively, so the sequence of operations is modified as following [5]:

↕ (W0, R0, W1, R1).

In testing for stuck-at faults, the order of the addresses to be accessed is not important. To exploit this fact, a low power address generator has to be selected. Usually, the Linear Feedback Shift Register (LFSR) is used as an address generator due to its little overhead in the hardware area. With the appropriate location for the XOR gate; it could generate all the possible addresses of the memory. Figure 5 shows the LFSR structure.

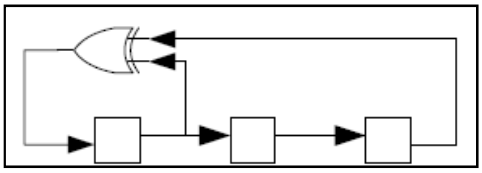

Figure 5: LFSR (x^3+x+1)

Although the LFSR occupies a low hardware area, there is low correlation between the addresses generated by it which means a high switching activity in the address decoder. Therefore, other types of LFSR were dedicated in order to reduce the switching activity in the vectors generated. The following provides a brief description about these types of LFSR.

A. Bit-Swapping LFSR (BS-LFSR): In this type, the LFSR structure is modified to apply swapping between the neighboring bits. The last bit is the selection line for the swapping process; if the last bit is 0, then swapping is performed, otherwise, nothing will change. It was proved that the BS-LFSR reduces the switching activity in the inputs of the Circuit Under Testing about 25%. Figure 6 illustrates this type of LFSR [13]

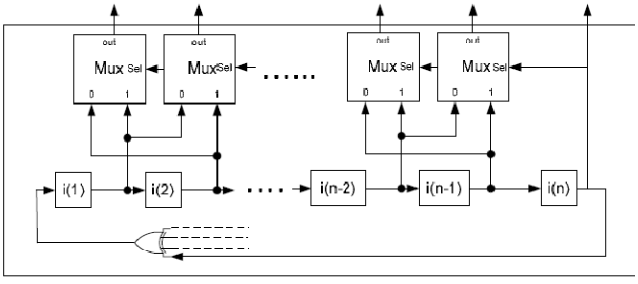

Figure 6: BS-LFSR

B. Dual Speed LFSR (DS-LFSR): This LFSR is commonly used in testing since it reduces the switching activity effectively. Instead of using one LFSR, two LFSRs are used: slow speed LFSR and normal speed LFSR. The slow-speed LFSR is driven by a slow clock whose speed is a fraction of the speed of the normal clock that drives the normal-speed LFSR. This will reduce the frequency of transitions in the circuit inputs that are driven by the slow-speed LFSR. Figure 7 shows the DS-LFSR [14].

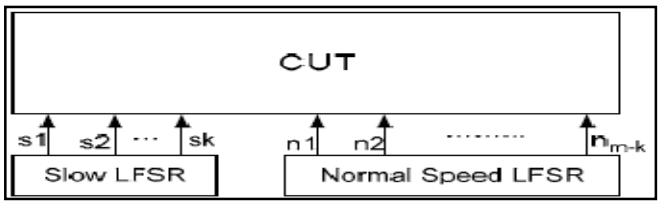

Figure 7: DS-LFSR

C. Bipartite LFSR: This LFSR is based on reducing the switching activity between the consecutive patterns through combining the second half of the current vector and the first half of the next vector into an intermediate vector. Figure 8 illustrates the functionality and the structure of this LFSR [15].

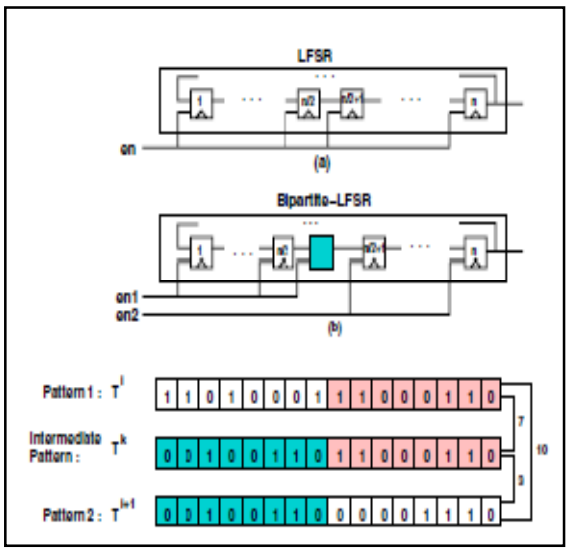

Figure 8: Bipartite LFSR

IV. Simulation And Results

A tool called **Xilinx ISE Design Suite 12.1 [16]** was used to simulate the system which was programmed using **VHDL** in behavioral model**.** The architecture shown in figure 4 was programmed so that the BIST engine applies the following sequence of operations in order to test the memory for the existence of stuck-at faults:

↕ (W0, R0, W1, R1).

Each read or write operation is performed in one clock cycle, thus, each address is accessed within four clock cycles. The period of the clock used was **1 ns.**

Five address generators were used in this system; **LFSR, BS-LFSR, DS-LFSR, Bipartite LFSR, and a combination of BS and DS LFSR.** Then**,** the switching activity in the

address decoder was calculated for each one. This was performed for various address bus lengths as shown in Tables 1 and 2. As mentioned before, Bit oriented memory was considered in the test. For each test, the maximal length LFSR was used in order to check all memory locations. Two seeds were used for each test. Figure 8 shows the simulation for fault free memory using 3-bit normal LFSR as an address generator.

Tables 1 and 2 show the switching activity for each of these address generators. Then, each one was compared with the normal LFSR by calculating the saving percentage in the SA.

As shown in the results, the LFSR causes a high switching activity in the address decoder due to the low correlation between the addresses generated .The main advantage of using the LFSR as an address generator is the low overhead in the hardware area.

The BS-LFSR reduces the switching activity if it is compared with the LFSR due to swapping process. Note that changing the seed doesn't affect the switching activity when LFSR and BS-LFSR are used.

It could be found that the DS-LFSR is more efficient in reducing the switching activity than the BS-LFSR since the frequency of transitions is reduced. The DS-LFSR reduces power consumption effectively and this appears well in large address spaces.

The bipartite LFSR is also efficient in reducing the switching activity, but using this address generator may lead to redundancy in the sequence of addresses generated and this will reduce the fault coverage achieved within the same testing time. Thus, bipartite LFSR will need more testing time to cover more memory locations. In general, bipartite LFSR reduces instantaneous power more than average power.

Using a DS-LFSR with BS-LFSR for its low and normal generators reduces switching activity significantly. Also the fault coverage will not be affected during testing. It could be found from the results that using this combination has the least switching activity in the address decoder.

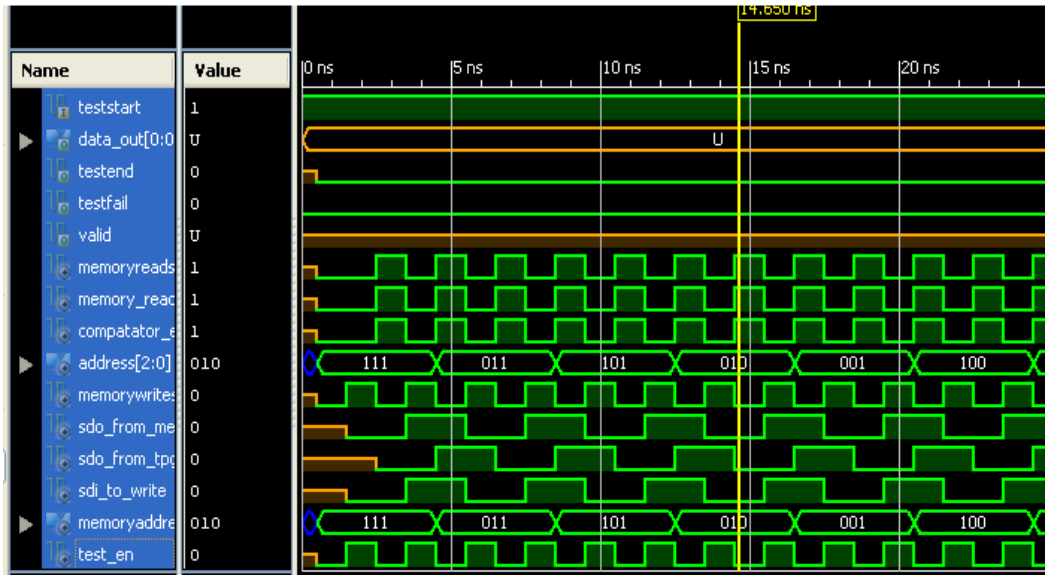

Figure 9: Fault Free Memory Simulation

Table 1 (seed="111….1")

Address Bus Length (bits)	Testing Time (ns)	LFSR (SA)	BS-LFSR (SA)	Saving %	DS-LFSR (SA)	Saving%	Bipartite LFSR (SA)	Savin g%	BS & DS LFSR (SA)	Saving %
5	128	85	69	19	70	18	45	47	64	25
10	4096	5130	4106	20	2904	43	2462	52	2376	54
15	131072	245775	188431	23	133324	46	121659	50	108652	56
20	4194304	10485780	8126484	22	5266450	50	5238247	50	4216850	60
25	134217688	419430211	318767107	24	218202082	48	209716053	50	167889901	60
30	152976488	516276423	454130346	12	286869408	44	286733465	44	219941228	57

Table 2 (seed="0101….01")

Address Bus Length (bits)	Testing Time (ns)	LFSR (SA)	BS-LFSR (SA)	Saving %	DS-LFSR (SA)	Saving %	Bipartite LFSR (SA)	Saving %	BS & DS LFSR (SA)	Saving %
5	128	85	69	19	55	35	48	44	57	33
10	4096	5130	4106	20	2643	48	2428	53	2315	55
15	131072	245775	188431	23	131524	46	121693	50	107626	56
20	4194304	10485780	8126484	22	5250055	50	5239003	50	4199893	60
25	134217688	419430211	318767107	24	218128300	48	209749569	50	167881672	60
30	152976488	516276423	454130346	12	286469208	44	277195287	46	219921231	57

V. Conclusions

The test power of embedded memories is a serious concern that may damage the chip if it exceeds the power envelop of the mobile device. High switching activities is the dominant source of power dissipation during testing. Thus, these signal activities have to be minimized.

In Zero-One testing algorithm, the order of addresses generated is not important, to exploit this fact, a low power address generator has to be used. LFSR occupies a low overhead in hardware area. Nevertheless, it causes a high switching activity in the address decoder. To solve this issue, other types of LFSR can be used as address generators for MBIST. Obtained results show that using BS and DS LFST as address generator can achieve around 60% reduction in switching activity in the address decoder for large address spaces with negligible overhead in hardware area. One of the possible future work is to combine the low power address generators with other low power algorithms such that low power March tests so that total switching activity in address bus and data bus is minimized.

References

[1 Nur Q.Noor, Yusrina Yusof, and Azilah Sparon, "Low Area FSM-Based Memory BIST for Synchronous SRAM, proceeding of the international colloquium of Signal Processing and Its application, 2009, pp. 409-412.

[2] Tzu F.Chien, Wen C.Chao, Chein M.Li, Yao W.Chang, Kuan Y.Laio, Ming T.Chang Min H.Tsai and Chih M.Tseng, "BIST Design Optimization for Large-Scal Embedded Memory Cores", ICCAD, SanJose, California, USA, 2009

[3 Said Hamdoui, and Zaid Al-Ars, "Scan More with Memory Scan Test", proceedings of 4th International DTIS conference, 2009, pp.204-209

[4] Nor Z.Haron, Siti A.Junos, Abdul Hadi A.Razak, and Mohd Y.Idris, "Modeling and Simulation of Finite State Machine Memory Built-in Self Test Architecture for Embedded Memories", Asia-Pacific Conference On Applied Electromagnetics", 2007

[5] Laung T.Wang, Charles E.Stroud, and Nur A.Toubam "System On Chip Test Acrhitectures",Morgan Kaufmann Publishers, 2008, pp.308-339

[6] Abdellatif S.Abu-issa, and Steven F.Quigley, "LT-PRPG: Power Minimization Technique for Test-Per-Scan BIST", International Conference on Design & Technology of Integrated Systems in Nanoscale Era, 2008

[7] R. Dean Adams, "High Performance Memory Testing: Design Principles, Fault Modeling and Self-Test, Kluwer Academic Publisher, 2003, pp. 104-139.

[8] Gayathri CV, Kayalvizhi N, and Malligadevi M, "Generation of New March Tests with Low Test Power and High Fault Coverage By Test Sequence Reordering Using Genetic Algorithm", International Conference on Advances in Recent Technologies in Communication and Computing,2009

[9] Yuejian Wu, and Andre Ivanov, "Low Power SoC Memory BIST", proceeding of the 21st IEEE international Symposium on Defect and Fault Tolerance in VLSI Systems, 2006.

[10] Luigi Dilillo, Paul Rosinger, and Bashir M.Al-Hashimi, "Minimizing Test Power in SRAM through Reduction of Pre-charge Activity", proceeding the Design, Automation, and Test in Europe (DATE) conference,2006,pp.1-6

[11 Mohammad Niamat, Manoj Lalla, and Junghwan Kim," Testing Faults in SRAM Memory of Vertex-4 FPGA", proceeding the IEEE International Midwest Symposium, 2009, pp.965-970

[12] Maria Fischerova, Tomas Pikula, Martin Simlastik, Alberto Bosio, Stefano, and Giorgio d.Natale, "A tool for teaching memory testing based on BIST", proceeding the international Baltic Electronic Conference, 2006

[13] A.S Abu-Issa, and S.F Quigley, "Bit-swapping LFSR for low-power BIST", Electronic Letters, 13 March, 2008

[14] Seongmoon Wang, and Sandeep K.Gupta, "DS-LFSR: A BIST TPG for Low Switching Activity, IEEE Transactions On Computer-Aided Design of Integrated Circuits And Systems, 7,July, 2002

[15] Mohammad Tehranipoor, Mehrdad Nourani, and Nisar Ahmed, "Low Transition LFSR for BIST-Based Applications", Proceedings of the 14th Asian Test Symposium, 2005

[16] http://www.xilinx.com/support/download, Xilinx Inc, Xilinx Design Suite Version 12.1

Mr. Ahmed Awad received his B.Sc degree in computer engineer from Birzeit University in Palestine. Then he got his master degree in computing also from Birzeit University. He worked and did researches in several areas in computer engineer. He is interested in VLSI and mainly in memory testing and has two publications in this field. His master thesis was about reducing test power of embedded memories. He is also interested in other topics such as computer networks - mainly Wireless Sensor Networks which was his graduation project- and Cryptography and network security. He is currently working as a Quality Assurance Engineer for CISCO Systems for automation backend, GUI and Regression testing.

Dr. Abu-Issa is the head of Computer Systems Engineering Department at Birzeit University – Palestine. He got his PhD in 2009 from the University of Birmingham – UK in the area of testing digital VLSI circuits with low power consumption. He has many publications in different areas mainly on testing of digital circuits and e-learning.