Fault Detection and Fault Diagnosis in SRAM-Based FPGA Using BIST

[1]Latha.M [2]Senthilmurugan.M

*[1]Assistant professor, Department of Electrical and Electronics Engineering,
Bharathiyar College Of Engineering and Technology , Thiruvettakudi, karakal,– 609609,
Puducherry, S.India, e-mail:mehalatha@yahoo.com,*

*[2]Professor, Department of Computer Applications, Bharathiyar College of Engineering,
Karaikal-609 609, Puducherry, S.India
Ph-+91-9842091911, Email: smsenthil@hotmail.com*

Abstract – We propose a novel approach to detect and diagnosis the faults in SRAM FPGA. BIST is a design technique that allows a circuit to test itself. The BIST technique can provide shorter test time compared to an externally applied test and allows the use of low-cost test equipment during all stages of production. The use of field-programmable gate arrays (FPGAs) to implement complex logic functions in digital applications has become increasingly common. FPGAs are regular structures of logic modules that communicate through an interconnected architecture of lines and switches. The logic modules and the interconnect structures are programmed to select a particular function of each logic module and specific interconnect paths to realize the global function of the FPGA. This proposed work presents a built-in self-test (BIST) design for fault detection and fault diagnosis of static-RAM (SRAM)-based field-programmable gate arrays (FPGAs). The proposed FPGA BIST structure can test both the interconnect resources [wire channels and programmable switches (PSs)] and lookup tables (LUTs) in the configurable logic blocks (CLBs). The test pattern generator and output response analyzer are configured by existing CLBs in FPGAs; thus, no extra area overhead is needed for the proposed BIST structure. The target fault detection/diagnosis of the proposed BIST structure are open/short and delay faults in the wire channels, stuck on/off faults in PSs, and stuck-at-0/1 faults in LUTs.

Keywords:- Built-in self-test (BIST), configurable logic block(CLB), programmable switch (PS), Look up Table (LUP), field-programmable gate array (FPGA), Static Read only memory (SRAM).

I Introduction

The use of field-programmable gate arrays (FPGAs) and to implement complex logic functions in digital applications has become increasingly common. FPGAs are regular structures of logic modules that communicate through an interconnected architecture of lines and switches [1]. The logic modules and the interconnect structures are programmed to select a particular function of each logic module and specific interconnect paths to realize the global function of the FPGA [4]. FPGAs are generally classified into two major types. One is one-time programmable FPGAs such as an antifuse type [7]. The other is unlimited reprogrammable FPGAs such as a static RAM (SRAM)-based FPGA. In SRAM-based FPGAs, logic elements and programmable switches (PSs) can be reprogrammed by loading a configuration bitstream, giving FPGAs incredible flexibility that is enough to implement any digital circuit on the same piece of silicon [11]. In observation of the fact that FPGA results in a shorter time to market and increased flexibility for systems using logic circuits, many applications have been developed to make the best use of FPGA reprogrammability. Currently, very-large-scale integration (VLSI) technology keeps increasing circuit integration by ever greater degrees, and the rapid development in packaging technology greatly reduces the controllability and observability of internal nodes [8]. This significantly complicates the testing of the system. Generally, FPGAs can be configured in an incredibly large number of ways. From the manufacturing testing point of view, it must be ensured that the part is functional under all configurations. Where as the reconfiguration time (the time to load a configuration into an FPGA) ranges from milliseconds to intact seconds, depending on the size of the FPGA [5].

The BIST method involves configuring one part of the FPGA to undergo testing and configuring the other parts to generate test vectors and to analyze test results. Afterward, the sub circuits being tested and the resources of the FPGA change roles, so that the entire FPGA is eventually tested. This way, testability is achieved without any overhead, since the BIST logic will “disappear” when the circuit is reconfigured for its normal system operation. In contrast, conventional BIST approaches introduce both area overhead (typically between 10% and 30%) and delay penalties. The proposed BIST approach can be applied to any in-circuit reprogrammable SRAM-based FPGAs. The only cost will be the additional memory for storing the data required to reconfigure the FPGA [8]. However, this memory is usually part of the test machine or system maintenance processor, which controls the BIST process and does not involve resources of the device or system being tested.

II Review of Literature

Several previous works extensively studied the testing of FPGAs assuming that only configurable logic blocks (CLBs) can be faulty. Similarly, others deliberated only the testing of the interconnect resources, assuming that CLBs were fault free [3]. So far, there has been relatively little research conducted on both interconnect resource and CLB testing [5]. Although the synthesis algorithm for self-checking the combinational logic of FPGAs can detect the faults within the combinational functional block of a CLB and on the interconnect lines connecting the functional blocks, this approach is short of the ability of fault diagnosis [9].

III Proposed Method

1. INTRODUCTION

To test an SRAM-based FPGA using a BIST technique, it has to be configured into TCs, and then, test vectors, input values, and primary inputs must be applied in each configuration [7]. In use as one, the FPGA industry sometimes refers to these as “test patterns.” Moreover, output response observations can be taken either via off-chip pins or a built-in output response analyzer (ORA) using the available logic resources of the FPGA [8]. The BIST method in this paper involves configuring one part of the FPGA to undergo testing and configuring the other parts to generate test vectors and to analyze test results.

2. ARCHITECTURE OF FPGA

The SRAM-based FPGA (see Fig1) consists of an array of $n \times n$ CLBs and local/global interconnect resources. The CLBs can be programmed with configuration cell data to generate logical functions. The set of all configuration cell data makes up an FPGA

configuration. The basic internal architecture of a CLB is made up of three components:
1) lookup tables (LUTs); 2) multiplexers (MUXs); and 3) D-type flipflops (DFFs). In Fig. 1, the LUTs implement either any logical function with *k*1 inputs or RAM. Every CLB is configured with configuration cells C_1, C_2,. . . , *Cn*. The number of outputs of LUTs is assumed to be k_3, and the inputs k_2 directly drive. Some of the MUXs or DFFs, producing the CLB outputs. Note that parameters k_1, k_2, and k_3 depend on the FPGA type [12].

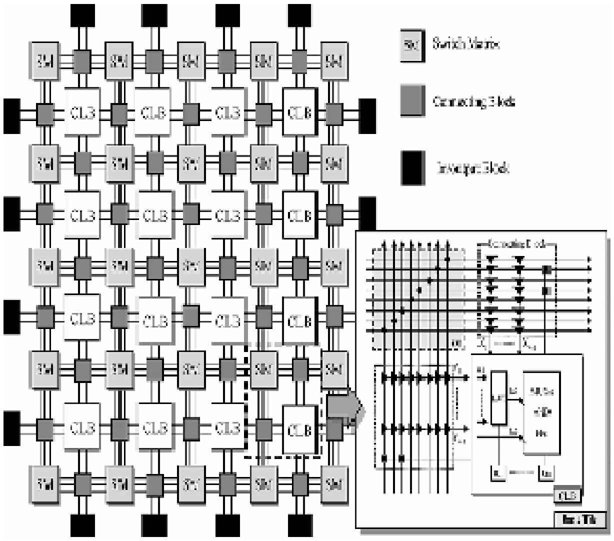

FIG 1 ARCHITECTURE OF FPGA

The local interconnects are associated with CLBs, including wire segments and connecting blocks. Note that a connecting block contains some programmable-interconnect-point PSs (PIP-PSs) and multiplexer PSs (MUX-PSs) to bring signals into and out of CLBs. On the other hand, wire segments and programmable cross-point PSs (PCP-PSs) within the switch matrix (SM) in global interconnects form horizontal and vertical routing channels that connect signals between CLBs [4]. The input–output signals can be transmitted into or out of the FPGA using the input–output blocks (IOBs). An SM is a programmable connecting element that receives *k* wires on each side. The wires connect SM pins identified as north, east, south, and west. Some pairs of pins in an SM cannot be connected; these are called nonconnectable pins. Pairs that can be connected are called connectable pins.

3. PROPOSED FPGA BIST DESIGN

The testing strategy of the proposed FPGA BIST structure is to configure groups of ten CLBs into a test block, as illustrated in Fig. 2 In each test block; four CLBs are configured as a TPG to generate the addresses for test patterns. Additionally, two CLBs are configured as an ORA for comparison with each output of the block under test (BUT) to observe the test results. The global/local interconnect resources and CLBs in a BUT, which are configured by four CLBs in a test block, are then sequentially tested. To guarantee the testing of all global/local interconnect resources and CLBs, the FPGA has to be reconfigured to shift the test blocks for testing. The test processes of the proposed FPGA BIST structure are simultaneously performed by a BIST controller, which repeatedly reconfigures the test blocks for testing. Briefly, the testing processes can be summarized in the following steps.

1) Reconfigure the FPGA to create test blocks.
2) Program the TCs.
3) Initiate the TS for global/local interconnect resources and CLBs.
4) Generate the test vectors.
5) Analyze the test results.

In other words, the test blocks are first (re)configured by the BIST controller. Second, the TCs should be reconfigured for global/local interconnect resource and CLB testing. Then, the LUT-based method is used to configure the TPG and ORA to generate the test vectors. Finally, the test results are analyzed.

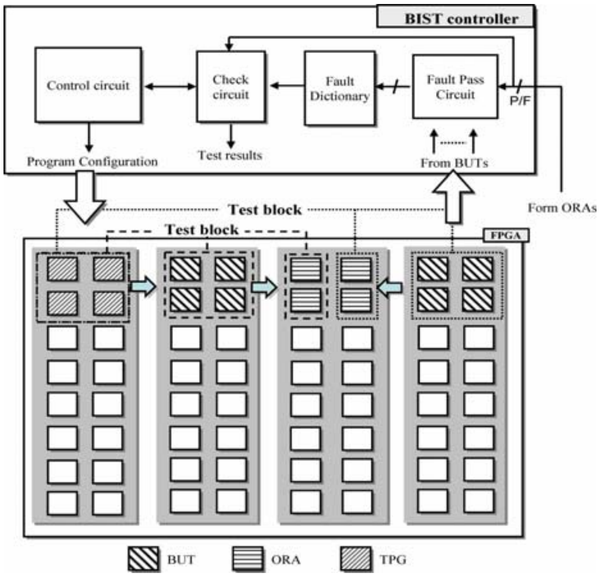

FIG. 2.PROPOSED BIST METOD

The fault models are summed up as follows:

1) *PIP-PS stuck-on fault*: The pass transistor of a nonconnectable PIP-PS is turned on in the local interconnect.

2) *PIP-PS stuck-off fault*: The pass transistor of a connectable PIP-PS is turned off in the local interconnect.

3) *MUX-PS stuck-on fault*: One of the pass transistors of a nonselective MUX-PS is turned on in the local interconnect.

4) *MUX-PS stuck-off fault*: One of the pass transistors of a selective MUX-PS is turned off in the local interconnect.

5) *Wire open fault*: A disconnection occurs on any wires in the global interconnect.

6) *Wires short fault*: A bridge occurs between two wires in the global interconnect.

7) *PCP-PS short fault*: A PIP-PS stuck-on fault occurs in the nonconnectable direction in a PCP-PS of wire segments.

8) *PCP-PS open fault*: A PIP-PS stuck-off fault occurs in the connectable direction in a PCP-PS of wire segments.

9) *LUT stuck-at-0 fault*: The RAM cell value of an LUT in the CLB is always 0.

10) *LUT stuck-at-1 fault*: The RAM cell value of an LUT in the CLB is always 1.

11) *Interconnect delay fault*: A transition is propagated from one end of a PUT, and the result is captured from the other end after a specified time (test clock period).

4. SCHEMA OF A BIST STRUCTURE IN A TEST BLOCK.

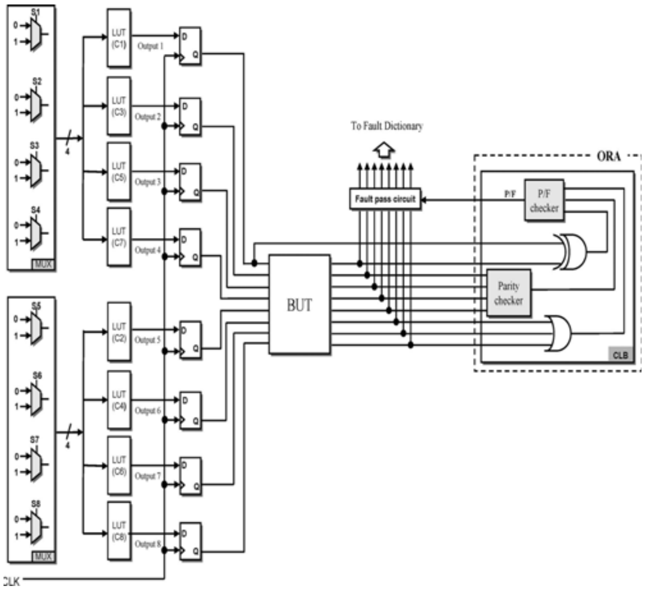

FIG. 3 BIST STRUCTURE

Fig. 3 to detect and diagnose the faults on both interconnects resources and LUTs in CLBs online. The testing process is performed by configuring the TPG, ORA, and BUT in each test block [11]. Additionally, the TPG and ORA of the proposed FPGA BIST structure are designed using the existing CLBs to reduce the extra area and test cost needed. The TPG in Fig. 3 is an address generator, which consists of eight MUXs, eight LUTs, and eight DFFs to continuously generate the addresses (0–15) in the LUTs to produce the corresponding test patterns. Significantly, to ensure the accuracy of LUTs in the TPG, an algorithm is needed to write different test patterns into LUTs, read the contents of LUTs, and check whether the read data are correct or not . Additionally, writing test patterns into LUTs needs configuration of the FPGA [10]. According to the algorithm, the test patterns can be obtained in Table I. In other words, according to the MUX selections, the LUTs' addresses can be figured out in sequence, and the test patterns can be generated by repeatedly checking the contents in Table I. It should be noted that the addresses do not need to be sequentially generated, but they have to cover the completed address space of the LUTs from 0 to 15 during testing. The data in the TPG, LUTs are important because on one hand, the combinations of data read from four LUTs have to generate all addresses ranging from 0 to 15, and on the other hand, they have to be memory test patterns in memory cells C1–C8, as shown in Table I. Fig. 4 clearly indicates that outputs 1–4 and 5–8 of the LUTs are configured according to patterns C1, C3, C5, C7 and C2, C4, C6, C8 in Table I. For example, if select lines S1–S4 and S5–S8 of the MUXs are set to "0010" and "0011," then the addresses of LUTs 1–4 and LUTs 5–8 are also indicated as "0010" and "0011," respectively. Thus, the corresponding cell values "0010" and "1100" in Table I from C1, C3, C5, C7 and C2, C4, C6, C8 can be captured at the LUTs' outputs. Finally, test pattern "00101100" can be generated when the DFFs are enabled. In other words, the different addresses of the LUTs can be obtained by carefully changing the signals of select lines S1–S4 and S5–S8 of the MUXs. Then, the series of test patterns can be sequentially captured when the DFFs are triggered at specific times. The ORA design in Fig. 3.4 also uses the existing CLBs in an FPGA [3]. In Fig.3.4, the two-input XOR gate and three-input OR gate in the ORA are used to test the global interconnect open and short faults, respectively. Otherwise, the faults presented in the LUTs and local interconnects are detected by using the four-input parity checker. The logic diagram of a simple parity checker is composed of a 4-b XOR and a unit delay. Significantly, the parity checker can be implemented with the internal logic gates and a DFF of a CLB [6]. Moreover, the P/F checker (functioning in the same way as a simple OR gate) in the ORA circuit is designed to generate a P/F signal to trigger the fault pass circuit to deliver the faults from the primary outputs of a BUT to the fault dictionary for fault diagnosis.

Address	**C1**	**C2**	**C3**	**C4**	**C5**	**C6**	**C7**	**C8**
0000	0	1	0	1	0	1	0	1
0001	0	1	0	1	0	1	1	0
0010	0	1	0	1	1	0	0	1
0011	0	1	0	1	1	0	1	0
0100	0	1	1	0	0	1	0	1
0101	0	1	1	0	0	1	1	0
0110	0	1	1	0	1	0	0	1
0111	0	1	1	0	1	0	1	0
1000	1	0	0	1	0	1	0	1
1001	1	0	0	1	0	1	1	0
1010	1	0	0	1	1	0	0	1
1011	1	0	0	1	1	0	1	0
1100	1	0	1	0	0	1	0	1
1101	1	0	1	0	0	1	1	0
1110	1	0	1	0	1	0	0	1
1111	1	0	1	0	1	0	1	0

TABLE 1
TEST PATTERNS FOR A FOUR-INPUT LUT

2) Fault Diagnosis: Fault detection is an important step in guaranteeing the quality of the product. Moreover, fault diagnosis is required, particularly if the manufacturer plans to enhance the yield or if the user intends to tolerate the faults. The target diagnosis in this paper is arrived at by means of locating the faulty BUT and its corresponding fault type. In other words, the SRAM-based FPGA is diagnosed here in two steps: 1) BUT diagnosis and 2) fault-type diagnosis. In step 1), the faulty BUTs can be located by directly observing the output signals, i.e., P/Fs, of ORAs. The fault-type diagnosis in step2) is performed by comparing the faults between the outputs of the faulty BUT and the fault dictionary. It should be noted that the fault dictionary records each of the faults, the 8-b signals from a faulty BUT*i* will be stored in register *i* (RGS*i*) when pass control circuit *i* (PCC*i*) is enabled by a faulty signal "1" from ORA*i*. Notably, all the faulty signals from BUTs in a specific TC can be simultaneously delivered and stored at their corresponding registers. Moreover, the registers are sequentially triggered by the timing circuitry. In other words, the faulty signals from the registers will be transmitted into the fault dictionary in sequence to indicate the fault type.

5. TEST RESULT

Test pattern	0010 1100
Fault dictionary	1010 1101
Open fault	**x**010 1100(if w1 is open)
Short fault	00**1**0 110**1**(if w3 and w8 are shorted)
Stuck off fault at 101	0010 **1**000(w5 is stuck off)
Stuck on fault at 011	001**1** 1100 (w4 is stuck on)
Stuck at '0' on LUT	1010 **0**101 (c5 stuck 0)
Stuck at '1' on LUT	1010 111**1** (c7 stuck 1)
Pass fail output	**1**

6. CONCLUSION

A new BIST approach for fault detection and fault diagnosis of SRAM-based FPGAs has been proposed in this paper. The proposed FPGA BIST structure has high fault coverage on the modeled interconnect and CLB faults, including short/open and delay faults in wire channels, stuck on/off faults in PSs, and stuck-at-0/1 faults in LUTs. The test results for the XC4000-series FPGAs have shown that adequate performance in fault coverage, test time, and area overhead can be achieved by using the proposed BIST structure. Comparisons with previous works have also shown that the proposed FPGA BIST structure possesses the ability to simultaneously detect and diagnose faults on both interconnect resources and CLBs.

REFERENCES

[1] O. Heron, T. Arnaout, and H. J.Wunderlich, "On the reliability evaluation of SRAM-based FPGA designs," in Proc. Int. Conf. Field Programmable Logic Appl., pp. 403–408, Aug. 2005.

[2] L. Sterpone and M. Violante, "A new reliability-oriented place and route algorithm for SRAM-based FPGAs," IEEE Trans. Comput., vol. 55, no. 6, pp. 732–744, Jun. 2006.

[3] The Programmable Logic Data Book, Xilinx Inc., San Jose, CA, 2003.

[4] M. Abramovici and E. Charles, "BIST-based test and diagnosis of FPGA logic blocks," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 9, no. 1, pp. 159–172, Feb. 2001.

[5] S. Xiaoling, J. Xu, and P. Trouborst, "Testing Xilinx XC4000 configurable logic blocks with carry logic-modules," in Proc. IEEE Int. Symp. Defect Fault Tolerance VLSI Syst., pp. 221–229, Oct. 2001.

[6] E. Atoofian and Z. Navabi, "A BIST architecture for FPGA look-up table testing reduces recomfigurations," in Proc. IEEE Asiau Test Symp., pp, 84–89 Nov. 2003.

[7] C. Stroud, J. Nall, M. Lashinsky, and M. Abramovici, "BIST-based diagnosis of FPGA interconnect," in Proc. Int. Test Conf., pp. 618–627, Oct. 2002.

[8] J. Liu and S. Simmons, "BIST-diagnosis of interconnect fault locations in FPGAs," in Proc. IEEE Elect. Comput. Eng., pp. 207–210, May 2003.

[9] J. Smith, T. Xia, and C. Stroud, "An automated BIST architecture for testing and

diagnosing FPGA interconnect faults," J. Electron. Test., Theory Appl., vol. 22, no. 3, pp. 239–253, Jun. 2006.

[11] S. Toutounchi and A. Lai, "FPGA test and coverage," in Proc. Int. TestConf., Baltimore, MD, 2002, pp. 599–607.

[12] Mrs. Shilpa Dandoti, Dr. V. D Mytri, "Detection and Location of Bridging Faults in Interconnects of FPGA Using Phase Path Method", pp. 178-182, International Journal of Advanced Engineering & Application, Jan. 2010.