Arid Zone Journal of Engineering, Technology and Environment. August, 2012; Vol. 8, 37-45

Print ISSN: 1596-2490, Electronic ISSN: 2545-5818
www.azojete.com.ng

TECHNOLOGIES FOR FAULTS DIAGNOSIS OF FPGA LOGIC BLOCKS

Ngene, C.U[1*]. and Hahanov, V.I[2].

([1]Department of Computer Engineering University of Maiduguri, Maiduguri, Nigeria,
[2]Department of Computer Design Automation, Kharkov National University of Radioelectronics, Kharkov, Ukraine)
*Corresponding author, e-mail address: ngene@unimaid.edu.ng

Abstract

The critical issues of testing field programmable gate arrays (FPGA) with a view to diagnosing faults are an important step that ensures the reliability of FPGA designs. Correct diagnosis of faulty logic blocks of FPGAs guarantees restoration of functionality through replacement of faulty block with replacement units. This process can be done autonomously or without the intervention of an engineer depending on application area. This paper considers two methods for analysing test results of FPGA logic blocks with the purpose of localising and distinguishing faults. The algebraic logic and vector-logical methods are proposed for diagnosing faulty logic blocks in FPGA fabric. It is found that the algebraic logic method is more useful for processing of sparse faults tables when the number of coordinates with 1s values with respect to zero values is not more than 20%, whereas the vector-logical method facilitates the analysis of faults table with predominance of 1s values.

Keywords: Diagnosis of faults, FPGA, Boundary scan, Fault table, Sum of Products, Product of Sums

1. Introduction

Field programmable gate arrays (FPGA) are programmable structured and regular device that consists of complex programmable logic blocks and interconnects. The escalating cost, time, and risk associated with custom integrated circuit (IC) fabrication has driven increased FPGA usage across electronics applications. Since their introduction by Xilinx in 1985, FPGAs have gained wide acceptability as a single prototyping platforms that incorporates the design simplicity of programmable logic devices (PLD) and design complexity of application specific integrated circuit (ASIC). FPGAs are larger, faster, and more power-efficient than ever, and bring a number of capabilities unavailable in custom silicon design, such as field updates, multi-function devices, and simplified prototyping, making them an attractive option. Though FPGA supports complex design and a single prototyping platform, their complexity creates new verification and testing challenges in real time. Furthermore, these challenges have given rise to other issues relating fault diagnosis and subsequent repair of faulty logic block (Kwang-Ting, 2006; Subhasish *et al.*, 2004). In connection with this, an algebra- and vector -logical methods are proposed for diagnosing faulty logic blocks in FPGA fabric. This is facilitated by the use of fault tables for troubleshooting and analysis in real time.

An FPGA comprises an array of independent Configurable Logic Blocks (CLBs), surrounded by a periphery of Input/Output Blocks (IOBs), which are interconnectable by configurable routing resources. The CLBs consists of logic look-up table (LUT). FPGAs can be based on 3-, 4-, 5- and even 6-input LUTs. A typical logic block consists of LUT, multiplexer and flip-flop. One of the most pressing problems in the market of electronic technologies is the tripod issues - testing, diagnosis and repair of digital systems, designed in the form of System-on-Chip. However, the maintenance of functional units, memory, and repair of failed components in real time remains an open topic for researchers. A typical structure of an FPGA containing other embedded components is depicted in Figure 1.

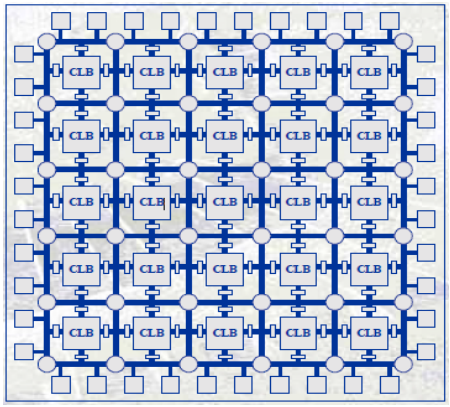

Figure 1: FPGA structure

In most cases FPGAs end up in Printed circuit boards (PCB) alongside other chips and passive elements. In order to test PCB interconnects and other components on board a boundary scan infrastructure (BSI) is provided. Boundary scan infrastructure is primarily used for the testing of component interconnect. As a standard all devices that are placed on PCB must also contain boundary scan infrastructure for the testing of that component. Though FPGAs provide infrastructure for implementing SoCs, they also contain BSI for their testing and programming. The principle of boundary scan is to allow the outputs of each IC to be controlled and inputs to be observed. A board with boundary scan infrastructure is shown in Figure 2 and the detail of a typical boundary scan cell is depicted in Figure 3. Every boundary scan-compliant component (e.g. FPGA) has common test architecture, shown in Figure 4.

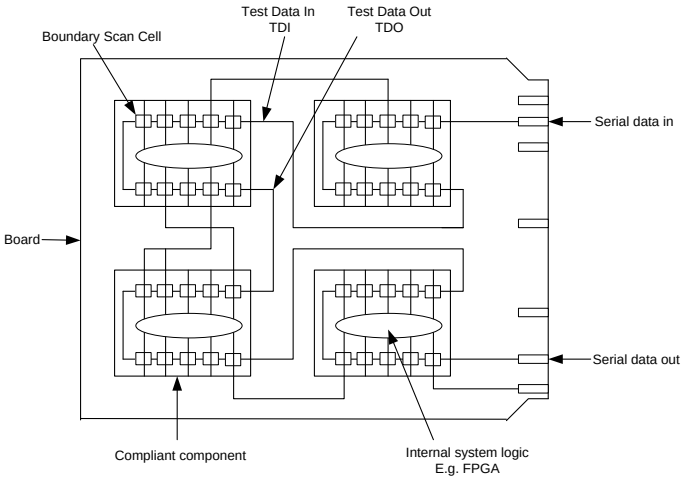

Figure 2: Board with Boundary Scan

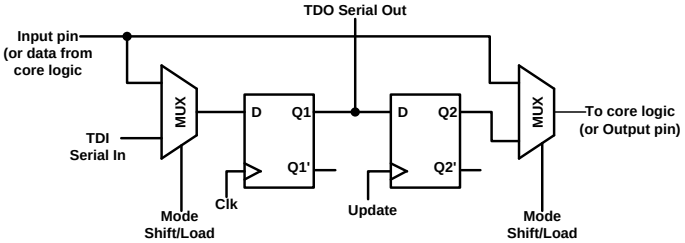

Figure 3: Typical Boundary Scan Cell

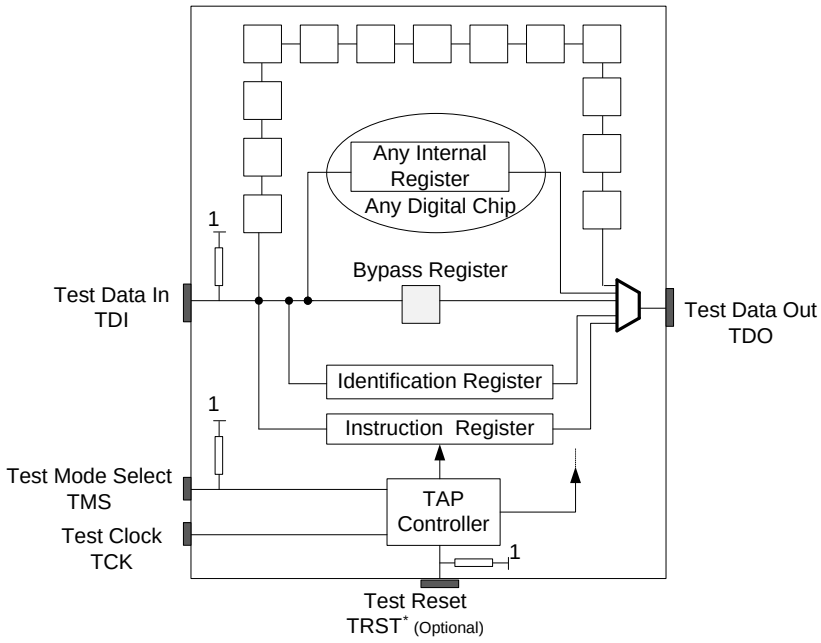

Figure 4: Boundary Scan Infrastructure IEEE 1149.1

In order to diagnose and repair faulty logic blocks within FPGA some level of redundancy is provided inform of replacement units. The more redundancy that is provided the easier it becomes to carryout replacement of multiple defective logic blocks. Physical defects in a chip, resulting from manufacturing process or operation appear as logical or delay defects and lead to malfunction of digital products. Defects are linked not only to gates or LUT-components, but also to a particular place on the chip.

2. Previous Work

In recent years there have been dozens of works that address issues relating to the problem of testing, diagnosis and repair of digital systems on chips FPGA. The critical issue of restoration of functionality the FPGA depends largely on the effectiveness and efficiency of locating and replacing failed components. The summary of various solutions that have been proposed include but not limited to the following.

In Ross and Hall (2006), the duplication of logic elements or regions of the chip, leading to a doubling of the hardware implementation was considered. When you locate a defective element or area a switching from the defective component to a good one is done with the help of a multiplexer). The proposed models for the repair of FPGAs from Xilinx can also be used for the repair of Altera FPGAs since the basic unit of repair is a column and row. The use of genetic algorithms to diagnose and restore functionality based on autonomous reconfiguration of FPGA chip without any external control devices (Habermann *et al.*, 2006). This result in high reliability of defects diagnosis up to 99% and repair time of 36 milliseconds instead of 660 seconds required for the standard configuration of the project. Restoration of functionality of FPGA chips is not time-critical, by replacing the local CLB with a spare as suggested in Pontarelli *et al.* (2006) and Koal and Vierhaus (2008). Moderate level of combining CLBs to be replaced for mission-critical applications is in the order of thousands of logic elements.

Repair of defective components in digital systems leads to the exclusion of the defective component at first instance and then replacing it with a healthy component. In an autonomous system this replacement is done independent of human intervention. This requires an extra circuit to carry out this procedure. Such a circuit is known as built-in-self repair (BISR). There are two procedures that exist for the replacement of the faulty component. One of the procedures requires repetitive Place and Route procedures after diagnosis. Restoration of failed CLB can be done by repeatedly carrying out the procedure of place and route to isolate the failed CLB. Repetition of place and route is time consuming and therefore, not suitable for time critical systems. In real time systems the preliminary preparation of all possible bitstreams are required to isolate future defective areas as part of the redundant non-functional region of the chip.

In order to handle multiple defects that are not covered by a reserve area, the project should be segmented, breaking it first into disjoint parts, which have their own maps of Place and Route. In this case, you can repair a failed digital system with distributed *n* defects on the wafer with *m* spare segments. In this case, the total chip area is n + m equal parts. This could be desirable for only mission critical systems which require high availability.

3. Technology for diagnosing FPGA blocks

The main role in the diagnosis of faulty FPGA blocks is assigned to the built-in chip boundary scan infrastructure (IEEE, 2001), which is aimed at solving practically all problems of service support for FPGA functionality. Access Controller provides monitoring of all problematic internal lines of the design with the help of boundary-scan register. The number of bits must conform to a prescribed depth of diagnosis or diagnostic resolution, and therefore the number of FPGA LUTs or cells. The service infrastructure for the diagnosis of defects in FPGA is shown in Figure 5.

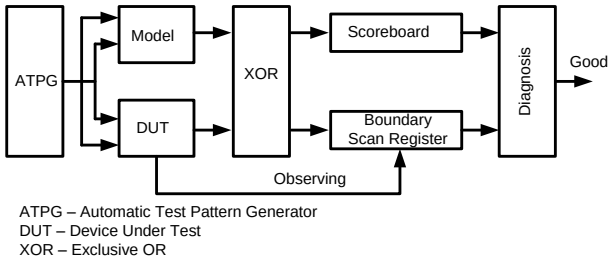

Figure 5: Process model for diagnosing FPGA

The exclusive-OR (XOR) block examines the module output response of the model and the real device (DUT) on the test vectors coming from the internal oscillator input sequences. The Boundary Scan Register Block is a multi-probe designed to monitor the state of all blocks or cells and components of the chip. The Scoreboard module performs the function of analyzing the results of monitoring for the diagnosis and subsequent repair of the components of SoC. The result of diagnosis of chip is a set of faulty units to be excluded from a functional operation by replacing them with spare components. An interesting solution to the problem of diagnosis can be obtained through the use of Boolean algebra to analyze the fault table (FT) M, which is the Cartesian product of the test T and the set of predetermined defects F. The output response vector R is equal to the length of the distinguishable test segments, which allows the procedure for search of defects to be used for coverage problem. This provides the most accurate results in

a sum of product form (SOP), where each term is a possible option of the presence of subset or a combination of defects in the device. The process model of diagnosis is presented in the following form:

$$\begin{aligned} &A = < T, F, M, R >, \\ &T = (T_1, T_2, ..., T_i, ..., T_n);\ F = (F_1, F_2, ..., F_j, ..., F_m); \\ &M = \left| M_{ij} \right|, i = \overline{1,n};\ j = \overline{1,m};\ R = (R_1, R_2, ..., R_i, ..., R_n); \\ &R_i = G(T_i) \oplus G^*(T_i);\ \{R_i, T_i, M_{ij}, F_j\} \in \{0,1\}. \end{aligned} \tag{1}$$

The value of the coordinates of vector R is the result of the XOR operation on the generalized model and the actual response output of the device. If the vector R is 1 for the i-th test segment on at least one output of the device, the generalized state of the output is 1. Otherwise, the value of the vector R is equal to 0.

4. Algebraic logic method for diagnosing faulty logic blocks

The solution to diagnostic problem is reduced to the analysis of fault table derived from the simulation of the defects of FPGA components. This is achieved by writing a product of sums (POS) of faults table rows recorded for which the output response vector R assumes the binary value 1:

$$F = \bigwedge_{\forall R_i = 1}^{i=\overline{1,n}} \left(\bigvee_{\forall M_{ij} = 1}^{j=\overline{1,m}} F_j \right). \tag{2}$$

∧ - Logical conjunctive operator (logical AND); ∨ - Logical disjunctive operator (logical OR)

The product of sum (POS) obtained from fault table is transformed to a sum of product (SOP) using the equivalence transformations (logical multiplication to minimization and absorption) (Rossen, 2003). The result is a Boolean function, where the terms - logical product represents the complete set of solutions in the form of a combination of defects (which gives the binary coordinates of V at the outputs of the FPGA or its components):

$$F = \bigwedge_{\forall R_i = 1}^{i=\overline{1,n}} \left(\bigvee_{\forall M_{ij} = 1}^{j=\overline{1,m}} F_j \right) = \left| \begin{matrix} a \vee ab = b \\ a \vee a = a \end{matrix} \right| = \bigvee_{i=1}^{2^m} \left(\bigwedge_{j=1}^{m} k_j F_j \right), k_j = \{0,1\}. \tag{3}$$

where*: k* is fixed number of 1s per row.

Equation 3 expresses diagnosis as a subset or combinations (multiple) of defects that need to be further clarified by the use of additional internal sensing points using boundary-scan register. The number of one's (1s) in R forms the number of sum terms of POS (3). Each term is the progressive (per line) recording of defects (through logical OR operation) affecting output functionality. Table M_1 is presented in the analytical form and by using the POS it is possible to significantly reduce the amount of diagnostic information to locate defects. The subsequent conversion of POS to SOP on the basis of the identities of Boolean algebra can significantly reduce the Boolean function.

The Algebra-logical method for a given a fault table M_1 shown in figure 6 is based on the following algorithm.

$M_1 =$

T_i / F_i	F_1	F_2	F_3	F_4	F_5	F_6	R_i
T_1	1			1			1
T_2							1
T_3			1	1		1	1
T_4	1		1				1
T_5		1			1	1	0

Figure 6: Fault Table

1. Definition of fault table rows corresponding to R equal to zero values in order to reset the one (1) coordinates in the matched rows. In this case - this corresponds to row T5 in figure 6.

2. Location of all the columns that have zero values in the row coordinates corresponding to R equal to zero. Reset the one (1s) values in the located columns. In this case: F2, F5, F6.

3. Remove from the fault table all rows and columns with only zero values (found in points 1 and 2). This is depicted in figure 7.

The empty cells in the table are zeroes and signify that a given test pattern T is unable to detect a specific fault F. For simplicity 0 has not been shown in the empty cells.

$M_1 =$

T_i / F_i	F_1	F_2	F_3	F_4	F_5	F_6	R_i
T_1	1			1			1
T_2		0			0		1
T_3			1	1		0	1
T_4	1		1				1
T_5		0			0	0	0

$=$

T_i / F_i	F_1	F_3	F_4	R_i
T_1	1		1	1
T_3		1	1	1
T_4	1	1		1

Figure 7: Reduced Fault Table

4. Construction of a POS for every R= 1. POS is converted to SOP by multiplying all the sum terms and further minimise the Boolean function using other minimization techniques. And this is expressed in equation 4.

$$
\begin{aligned}
&F = (F_1 \vee F_4) \wedge (F_3 \vee F_4) \wedge (F_1 \vee F_3) = (F_1F_3 \vee F_3F_4 \vee F_1F_4 \vee F_4F_4) \wedge (F_1 \vee F_3) = \\
&= F_1F_1F_3 \vee F_1F_3F_4 \vee F_1F_1F_4 \vee F_1F_4F_4 \vee F_1F_3F_3 \vee F_3F_3F_4 \vee F_1F_3F_4 \vee F_3F_4F_4 = \\
&= F_1F_3 \vee F_1F_3F_4 \vee F_1F_4 \vee F_3F_4 \vee F_1F_3F_4 \vee F_3F_4 = \\
&= F_1F_3 \vee F_1F_4 \vee F_3F_4.
\end{aligned}
\qquad (4)
$$

The proposed algorithm is focused on the analysis of faults table in order to reduce its size and simplify subsequent computations relating to the construction of SOP, which generates all the solutions for the diagnosis of FPGA functionality. The algorithm enables the processing of sparse faults tables when the number of coordinates with 1s values with respect to zero values is not more than 20%. Further refinement of the diagnosis is only possible with the use of a multi-sensor based on boundary-scan register (IEEE 2001).

5. The vector-logical method of diagnosing faults

The processing of FT for diagnosis is performed according to scenario that is based on the use of logical AND, logical OR and negation (NOT) operations on the rows of the fault table. Analytic vector-logical model of diagnosis of multiple defects is determined by the conjunction of two components, where the first is the disjunction of the vectors corresponding to the coordinates of the output response vector with ones (1s) values and the second - an inversion of the disjunctions of vectors corresponding to zero coordinates of R as expressed in equation 5.

$$F = M^1 \wedge \overline{M}^0 = \left(\bigvee_{R_i=1} M_i \right) \wedge \left(\overline{\bigvee_{R_i=0} M_i} \right). \quad (5)$$

The diagnostic model for a single defect is different in execution from the first step using conjunction operation (instead of disjunction) of all vectors corresponding to the coordinates with R equals 1 as expressed in equation 6.

$$F = M^1 \wedge \overline{M}^0 = \left(\bigwedge_{R_i=1} M_i \right) \wedge \left(\overline{\bigvee_{R_i=0} M_i} \right). \quad (6)$$

Example1. In this example a multiple defects diagnosis of FPGA is carried out by using vector-logical method for which specified faults table and output response vector are provided:

Table 1: Example of Multiple Defects Table

S/No.	Ti / F_j	F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8	F_9	F_{10}	R
1.	T_1				1						1	1
2.	T_2		1					1				0
3.	T_3			1			1			1		0
4.	T_4	1										0
5.	T_5					1			1			1
6.	T_6	1	1									0
7.	T_7			1								0
8.	T_8				1							1
9.	T_9					1	1					0
10.	T_{10}							1				0
11.	T_{11}								1	1	1	1
12.	$\vee M^1$				1	1			1	1	1	1
13.	$\wedge M^1$	0	0	0	1	0	0	0	1	0	1	1
14.	$\vee M^0$	1	1	1		1	1	1		1		0
15.	$\overline{M}^0$	0	0	0	1	0	0	0	1	0	1	1
16.	F	0	0	0	1	0	0	0	1	0	1	1

Table 1 shows eleven test vectors (T_1 – T_{11}) used to diagnose ten faults (F_1 – F_{10}). The processing of the fault table in accordance with equation (5) yields the results shown in the last four rows (rows 12 to 15) of Table 1. The last row of the table 1 is derived using equation (6) by taking logical conjunction of rows 13 and 15. This indicates the presence of defects in the circuit, which is represented in a vector or a set-theoretic form F = (0001000101) = {F_4, F_8, F_{10}}. This shows that faults F_4, F_8, and F_{10} have been detected. Taking a closer look at the table you find out that T_1 detects two faults F_4 and F_{10} and the two faults are not distinguishable but T_8

detects only F_4. Most of the Test vectors could not distinguish the faults hence the use of equation (6) to determine the exact fault and their subsequent location.

The list of detected faults can be further reduced by transforming this solution to SOP of the multiple defects. This is achieved by masking vector F in table 1 using equation 7.

$$F = M^1 \wedge F = \bigwedge_{R_i=1} (M_i \wedge F). \tag{7}$$

The resulting vectors, which is limited by the number of 1s available in R and the vectors are logically multiplied. In addition, each vector can be compactly written as a disjunction of the coordinates with ones (1s) values. The synthesis of the SOP results in equation 8.

$$\begin{aligned} &F = (F_4 \vee F_{10})(F_8)(F_4)(F_4 \vee F_{10}) = \\ &= F_4F_8F_4F_4 \vee F_{10}F_8F_4F_4 \vee F_4F_8F_4F_{10} \vee F_{10}F_8F_4F_{10} = \\ &= F_4F_8 \vee F_{10}F_8F_4 \vee F_4F_8F_{10} \vee F_{10}F_8F_4 = F_4F_8. \end{aligned} \tag{8}$$

This result is interesting, thanks to the fact that the defects were written in the form of SOP, which covers all 1s coordinates of R. It is now possible to eliminate the defect $F_{10} \in F$ from the list of faults.

The advantage of vector-logical method is in its use to analyse technological effectiveness of the fault table with computational complexity that has a multiplicative dependence on the number of defects and the power of test: $Q = n \times m$. The method should be used with a predominance of coordinates with ones (1s) values in the faults table when algebraic logic method gives a high assessment by using Quine complexity for SOP and POS. The disadvantage is the inability to represent all combinations of defects that form terms that cover the ones coordinates of the output response vector.

6. Conclusions

The proposed methods of diagnosis: algebraic logic and vector-logical offer to a specialist in the area of design and test of digital systems-on-chip mathematical tools that can accomplish diagnosis of defective parts given a pre-built faults table. With the vector solutions that were effectively obtained in the second method, we can represent all possible combinations of defects in the form of terms of SOP, which is characteristic of the first method. The second method is effective when the number of ones in a fault matrix exceeds 10-20%.

References

Habermann, S., Kothe, R. and Vierhaus, HT. 2006. Built-in self repair by reconfiguration of FPGAs // Proceedings of the 12th IEEE International Symposium on On-Line Testing, pp. 187-188.

IEEE 2001. IEEE Standard 1149.1: Test Access Port and Boundary-Scan Architecture. Available from: https://standards.ieee.org/findstds/standard/1149.1-2001.html/ accessed June 20, 2008.

Koal, T. and Vierhaus, HT. 2008. Basic Architecture for Logic Self Repair // 14th IEEE International On-Line Testing Symposium. pp. 177–178.

Kwang-Ting, C. 2006. The Need for a SiP Design and Test Infrastructure // IEEE Design and Test of Computers, May–June, pp. 181.

Pontarelli, S., Ottavi, M., Vankamamidi, V., Salsano, A. and Lombardi, F. 2006. Reliability Evaluation of Repairable/Reconfigurable FPGAs // 21st IEEE International Symposium on Defect and Fault-Tolerance in VLSI Systems (DFT'06), October, pp. 227-235.

Ross, R. and Hall, R. 2006. A FPGA Simulation Using Asexual Genetic Algorithms for Integrated Self-Repair // Adaptive Hardware and Systems, AHS 2006, First NASA/ESA Conference, 15-18 June, 2006, pp. 301 – 304.

Rossen, K. 2003. Discrete Mathematics and its Applications, McGraw Hill, 2003. Pp 824.

Subhasish, M., Huang, WJ., Saxena, NR., Yu, SY. and McCluskey, EJ. 2004. Reconfigurable architecture for autonomous self-repair // IEEE Design & Test of Computers, 21(3): 228-240.