1997 IEEE Asian Test Symposium, pp. 276-281, November 1997.

On the Complexity of Universal Fault Diagnosis for Look-up Table FPGAs

Tomoo Inoue, Satoshi Miyazaki[†], and Hideo Fujiwara
Graduate School of Information Science, Nara Institute of Science and Technology
8916-5 Takayama, Ikoma, Nara 630-01, Japan

Abstract— *In this paper, we introduce* ***universal fault diagnosis*** *such that when applied to an unprogrammed FPGA, it locates a fault in any faulty programmed FPGA corresponding to the unprogrammed FPGA. If a faulty part in an FPGA can be identified prior to programming it, we can implement a required logic function on the fault-free part by isolating the faulty part. Then, we propose a* ***universal fault diagnosis procedure*** *that locates a faulty CLB in a look-up table FPGA. The complexity of the universal diagnosis procedure for FPGAs with block-sliced loading is independent of its array size, i.e.,* ***C-diagnosable****.*

I. Introduction

Field-programmable gate arrays (FPGAs) are digital devices that can implement logic circuits required by users in the field [1], [2]. Because of their short turnaround time, low manufacturing cost and programmability in the field, there has been an increasing interest in system prototyping and system reconfiguration using FPGAs. There are many different architectures of FPGAs driven by different programming technologies. One important class is the SRAM-based FPGAs (e.g. Xilinx [1]–[3]), also called the *look-up table FPGAs*, which can be reprogrammed any number of times. In this paper, we shall consider look-up table FPGAs.

Testing for FPGAs, as well as conventional digital ICs, is an indispensable process to ensure the proper operation of circuits. Many works on testing FPGAs have been reported [4]–[10]. For FPGAs, two types of testing are considered; one is testing for unprogrammed FPGAs, and the other is testing for programmed FPGAs. As the former testing, i.e., testing for unprogrammed FPGAs, Inoue *et al.* [7] introduced *universal test* such that when applied to a given unprogrammed FPGA, it ensures that all programmed FPGAs corresponding to the unprogrammed FPGA are fault-free, and proposed a *universal test procedure* for look-up tables in FPGAs. Michinishi *et al.* [8] and Renovell *et al.* [9] presented a test methodology for the interconnect structure of unprogrammed look-up table FPGAs. Michinishi *et al.* [10] also presented a test methodology for configurable logic blocks (CLBs) in unprogrammed look-up table FPGAs.

Fault diagnosis for FPGAs is also one of the important problems. In the same way as testing, two types of fault diagnosis can be considered; one is fault diagnosis for unprogrammed FPGAs, and the other is that for programmed FPGAs. The former, fault diagnosis for unprogrammed FPGAs is particularly important — if a faulty part in an FPGA can be identified prior to programming it, by isolating the faulty part, we can implement a required logic on the FPGA using only the fault-free part. Lombardi *et al.* [11], [12] presented a method for diagnosing interconnects and its application to unprogrammed FPGAs.

In this paper, as an approach to fault diagnosis for unprogrammed FPGAs, we shall introduce *universal fault diagnosis* such that when applied to an *un*programmed FPGA, it locates a fault in *any* faulty programmed FPGA corresponding to the unprogrammed FPGA. Based on a test procedure for CLBs presented in [10], we shall propose a *universal fault diagnosis procedure* of which diagnostic resolution is 1 CLB, i.e., it can locate a fault just to one CLB. Then, we shall present *universal fault diagnosis complexity* which refers to the time required to perform a universal fault diagnosis procedure. The universal fault diagnosis complexity of the proposed diagnosis procedure for *sequentially loadable* FPGAs (*SL-FPGAs*)[7] is $O(N^2 n \log n)$, where N is the array size of the FPGAs and n is the size of a look-up table, i.e., it depends on the array size of the FPGA.

If we can make universal diagnosis complexity independent of the array size, the complexity is considerably reduced. Hence, by extending a class of *C-testable* FPGAs[7], we shall propose a class of *C-diagnosable* FPGAs, where an C-diagnosable FPGA is an FPGA such that there exists a universal fault diagnosis procedure of which complexity is independent of its array size. Then, we shall present another universal fault diagnosis procedure of which complexity is independent of the array size N, and shall show its application to *block-sliced* SL-FPGAs[7]. The complexity of this universal fault diagnosis procedure for block-sliced SL-FPGAs is $O(n \log n)$, and thus we can obtain C-diagnosable FPGAs.

II. Architecture of FPGA

The architecture of field-programmable gate arrays (FPGAs) considered in this paper is illustrated in Fig. 1. An FPGA consists of an array of $N \times N$ programmable logic blocks called configuration logic blocks (CLBs), programmable I/O blocks and a programmable interconnect structure. On each side of the FPGA, tN I/O blocks are placed, i.e., the FPGA has $4tN$ I/O blocks.

Each CLB consists of a single look-up table (LUT), two multiplexers (MUXs) and a single D-type flip-flop (DFF) connected as shown in Fig. 1(b). An LUT implements combinational logic as $2^k \times 1$ memory composed of configuration memory cells (CMCs), where k is the num-

[†] S. Miyazaki is currently with SHARP Corporation, Tenri, Nara, Japan.

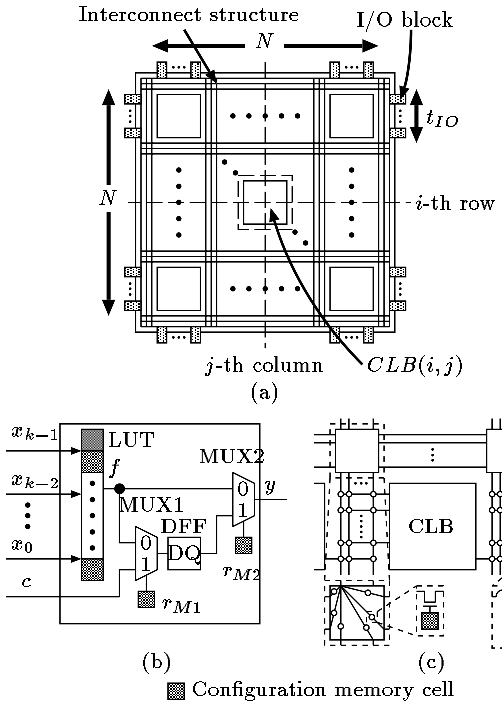


Fig. 1. Architecture of FPGA.(a) $N \times N$ FPGA, (b) CLB, (c) Interconnect structure.

ber of input lines of the LUT. When an input pattern is applied to an LUT, the LUT selects a CMC addressed by the input pattern, and the output of the cell provides the value of the function. An LUT can therefore implement any of 2^n functions of its inputs, where $n = 2^k$. When the FPGA is programmed, the memory is loaded with the bit pattern corresponding to the truth table of the function. The connections among the input and output lines, the LUT and the DFF in a CLB is configured by MUXs controlled by CMCs.

A connection among CLBs and I/O blocks is implemented via the interconnect structure which surrounds CLBs. A connection in the interconnect structure is configured by a pass transistor, also controlled by a CMC, as shown in Fig. 1(c).

A look-up table FPGA is programmed by loading a program composed of a bit sequence into its CMCs Each bit of the program is stored in the corresponding CMC, and consequently all the CLBs and interconnections are configured. Accordingly, a logic function or a *configuration* is implemented on the FPGA. The FPGA must include circuitry to load a program. Two types of programming scheme have been presented [7].

Sequential loading: When an FPGA is programmed, the program is shifted into the FPGA, and each bit of the program is stored in the corresponding configuration memory cell. This type of loading scheme is called *sequential loading*, and an FPGA with this type of loading is called a *sequentially loadable* FPGA (*SL-FPGA*). Whenever an SL-FPGA implements configurations, it loads all configuration memory cells.

Random access loading: Each configuration memory cell is directly addressable. When an FPGA is programmed, each bit is loaded by means of its address, and stored in the corresponding cell. This type of loading scheme is called *random access loading* and an FPGA with this type of loading is called a *random access loadable FPGA* (*RAL-FPGA*). An RAL-FPGA can implement a configuration by loading only the bits which differ from those of the previous one.

In the rest of paper, unless described explictly, we discuss fault diagnosis for SL-FPGAs and omit "sequentially loadable" for the sake of simplicity. Fault diagnosis for RAL-FPGAs can be considered in the same way as SL-FPGAs.

III. Universal Fault Diagnosis

For look-up table FPGAs, we can consider two types of testing; one is testing for programmed FPGAs and the other is testing for unprogrammed FPGAs. As the latter testing, we have been proposed *universal test* such that when applied to an unprogrammed FPGA, it ensures that all the configuration implemented on the FPGA are fault free[7]. In the same way as testing, we can consider two types of fault diagnosis for FPGAs; one is fault diagnosis for programmed FPGAs, and the other is that for unprogrammed FPGAs. As the latter approach, we introduce universal fault diagnosis such that when applied to an unprogrammed FPGA, it locates a fault in any faulty configuration implemented on the FPGA. Even if an FPGA is faulty , by applying such universal fault diagnosis to the faulty FPGA prior to programming it, we can isolate the faulty part identified from fault-free part and can implement a required logic function using only the fault-free part.

A. Fault Model

The following faults are defined.

Faults of LUT: Let $A = \{a_0, a_1, ..., a_{n-1}\}$ denote a set of input patterns for an LUT, i.e., a set of addresses of configuration memory cells of the LUT. Let $M = \{m_0, m_1, ..., m_{n-1}\}$ denote a set of configuration memory cells in an LUT. The decoding function of an LUT can be modeled as a mapping from A to 2^M. The function of a fault-free (i.e. correct) LUT is expressed as $f(a_i) = \{m_i\}$ for $0 \leq i \leq n-1$. Note that the range of the fault-free decoding function f is a singleton for all a_i. Then, we define four fault models for an LUT as follows.

Stuck-at fault: The value that is read out from the memory cell m_i corresponding to input pattern a_i is always either 0 or 1, irrespective of configurations.

Incorrect access fault: For some input pattern $a_i \in A$, whenever memory cell m_i is to be accessed by input pattern a_i, another memory cell m_j which does not correspond to a_i is accessed, i.e., $f(a_i) = \{m_j\} \neq \{m_i\}$.

Non-access fault: For some input pattern $a_i \in A$, whenever memory cell m_i is to be accessed by input pattern a_i, no memory cell is accessed. That is, $f(a_i) = \phi$. The output value for the input pattern a_i becomes the same as the output value before applying the input pattern.

TABLE I
Test procedure for CLB: TP_{CLB}

(a) Configuartions C_i

i	LUT	MUX1	MUX2
1 to k	$f = x_{i-1}$	1	0
$k+1$ to $2k$	$f = \overline{x_{i-1}}$	1	0
$2k+1$	$f = x_0$	1	1
$2k+2$	$f = x_0$	1	1
$2k+3$	$f = x_0$	1	0
$2k+4$	$f = x_0$	0	1

(b) Input sequences for C_1 to C_{2k}

Config.	Input sequence S_i	
	$x_{k-1} \cdots x_1 x_0$	c
C_i for $1 \le i \le k$	$0...01, 0...10, \ldots, 1...11, 0...00$ (applied in this order)	0
C_i for $k+1 \le i \le 2k$	$0...01, 0...10, \ldots, 1...11, 0...00$ (applied in arbitrary order)	0

(c) Input sequences for C_{2k+1}

Config.	Input sequence S_i	
	$x_{k-1} \cdots x_1 x_0$	c
C_{2k+1}	$0...00$	$0, 0, 0, 1, 1, 0$ (in this order)

(d) Input sequences for C_{2k+2} to C_{2k+4}

Config.	Input sequence S_i $(x_{k-1} \ldots x_1 x_0, c)$
C_i for $2k+2 \le i \le 2k+4$	$(0...00, 0), (0...01, 1), (1...11, 0), (1...11, 1)$

Multiple access fault: For some input pattern $a_i \in A$, whenever only memory cell m_i is to be accessed by input pattern a_i, more than one memory cells are accessed. That is, $|f(a_i)| \geq 2$. The output value under this fault is formed by the bitwise OR or AND function, which depends on the device technology, over the memory cells that are accessed by a_i. over the memory cells in the set $f(a_i)$.

Functional fault of MUX: A fault that changes an MUX to another combinational logic function.

Functional fault of DFF: A fault that changes a DFF to another sequential circuit such that the number of its states is the same as the DFF (i.e., 2 states).

In the following discussion, we assume that multiple faults that are defined above may occur in a CLB and that the number of faulty CLBs (i.e., CLBs including the above faults) is at most one in an FPGA. Then, we consider a *univesal fault diagnosis procedure* of which *diagnostic resolution* is 1 CLB, i.e., it can locate faults just to one CLB.

B. F ault Diagnosis with Univesal T est Procedure

As an approach to universal test, a test procedure for a CLB in SL-FPGAs has been presented [10]. The test procedure is denoted by TP_{CLB} here. Test procedure TP_{CLB} is performed by repeating implementation of a configuration and application of an input sequence to the configuration alternately. That is, test procedure TP_{CLB} can be represented by a sequence of pairs of a configuration and an input sequence applied to the configuration as follows:

$$TP_{CLB} = \langle (C_1, S_1), (C_2, S_2), \ldots, (C_{n_c}, S_{2k+4}) \rangle$$

where C_i is the i-th configuration, S_i is the input sequence which is applied to the i-th configuration C_i, and k is the number of input lines of an LUT. T able 1 shows the configurations and the input sequences in test procedure TP_{CLB}. It is proved that test procedure TP_{CLB} can detect any fault defined in Section III.A [10].

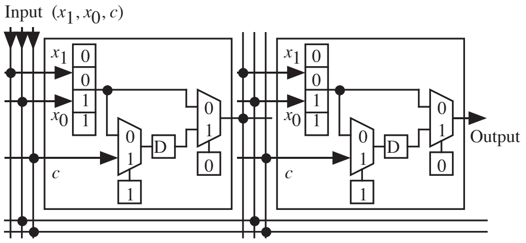


Fig. 2. Example: Testing of two CLBs (at configuration C_2 in test procedure TP_{CLB}).

It is clear that if this test procedure TP_{CLB} is applied to each CLB in an FPGA with the implementation of a connection between the CLB and I/O blocks, faulty CLBs can be identified, though much time will be consumed. Otherwise, if connections such that an input sequence can be applied to all the CLBs and the output sequences from each CLB can be configured, test procedure can be applied to all the CLBs simultaneously, and consequently faulty CLBs can be located. However, the number of CLBs in an FPGA is$N^2 = N \times N$, whereas the number of I/O blocks in an FPGA is$O(N) = 4tN$. Hence, if the array size N becomes large, it will be impossible to configure such connections.

C. Universal Fault Diagnosis Procedure

Suppose the input sequence S_i at a configuration C_i in test procedure TP_{CLB}. Let R_i be the output sequence obtained by applying input sequence S_i to a fault-free CLB that implements configuration C_i. It has been shown that output sequence R_i can be used as a certain bit sequence of S_i for other CLBs implementing the same configuration C_i [10]. Therefore, if the output line of a CLB is connected with an appropriate input line of another CLB, multiple CLBs in an FPGA can be simltaneously tested with a small number of I/O blocks (See an example in Fig. 2.). In [10], based on this idea, a test procedure such that several CLBs are connected in a cascade to test all the CLBs in an FPGA concurrently has been presented.

However, recall that the our goal of diagnostic resolution is 1 CLB. As long as output responses from several CLBs are observed via a single primary output implemented by an I/O block, as shown in Fig. 2, we cannot see which of the CLBs is faulty . Hence, in order to identify just one faulty CLB by means of a limited number of I/O blocks, we have to apply test procedures several times with the change of configurations in the interconnect structure.

Here, let us consider the number of test procedures required to locate a faulty CLB.

Suppose that N_O primary outputs can be implemented at each configuration in the underlying test procedure. Then, the average number$d$ of CLBs that are connected to a primary output, i.e., the diagnostic resolution (i.e., the number of candidates for faulty CLBs)

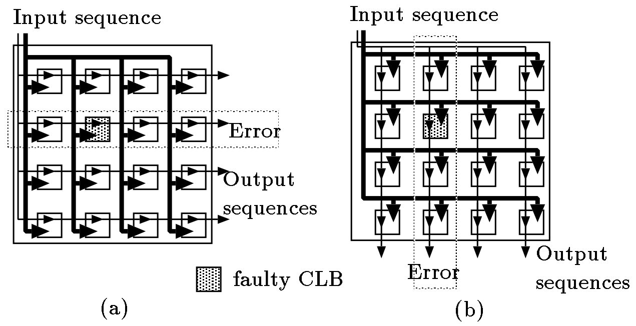


Fig. 3. Universal diagnosis procedure DP_1.

of the first test procedure is $d = N^2/N_O$ where N^2 is the total number of CLBs. By applying the test procedure with different connections of CLBs again, the diagnostic resolution can be further reduced to $d = N^2/N_O^2$. Hence, the diagnostic resolution after applying the test procedure i times can be expressed as $d_i = N^2/N_O^i$. When N_I primary inputs are implemented to feed the same input sequences to all CLBs concurrently, the numberN_O of primary inputs that can be implemented is expressed as $N_O = 4tN - N_I$ where $4tN$ is the total number of I/O blocks. Since the number of input lines of a CLB is $k+1$, $N_I = k+1$. Hence, $N_O = 4tN - (k+1)$. From the above equations, the condition of the number i of test procedures for 1 CLB diagnostic resolution can be expressed as

$$N^2 \le (4tN - (k+1))^i. \qquad (1)$$

When $i = 2$, any FPGA can be considered to satisfy this condition even if its array size N is large. Thus, here we present a universal diagnosis procedure which consists of two test procedures.

A universal diagnosis procedure, denoted by DP_1, consists of two steps. At every step in DP_1, all the CLBs are tested concurrently in the same way as test procedure TP_{CLB}, i.e., configuration C_i is implemented on each CLB and input sequence S_i is applied to all the CLBs for $1 \le i \le 2k+4$, where C_i and S_i are the configuration and the input sequence in test procedure TP_{CLB}, as shown in Table 1. Hence, universal diagnosis procedure DP_1 can be also expressed as

$$DP_1 = \langle (C_1, S_1), (C_2, S_2), \ldots, (C_{n_c}, S_{n_c}) \rangle$$

where

$$n_c = 2(2k+4) = 4k+8. \qquad (2)$$

The difference between the two steps is the configurations of connections implemented on the interconnect structure as follows (Fig. 3).

Step 1 (Horizontal diagnosis): Let $CLB(l,m)$ be the CLB at the l-th column in the m-th row. At all the configuration C_i, the output line of $CLB(l,m)$ is connected with an appropriate input line of $CLB(l+1,m)$ for $1 \le l \le N-1$ for all m. The output line of the rightmost CLB $CLB(N,m)$ is connected with an I/O block as a primary output for all m. The other input lines are connected with remaining I/O blocks as primary outputs.

Step 2 (Vertical diagnosis): By exchanging the column number$l$ for the row number m, the interconnect structure is configured in the same way as Step 1.

Note that a cascade consisting of N CLBs is implemented on each row (resp. column) at Step 1 (resp. Step 2).

In diagnosis procedure DP_1, Step 1 and Step 2 identify the row and the column that include a faulty CLB with N primary outputs, respectively. As a result, DP_1 can identify just one CLB. Note that this universal diagnosis procedure DP_1 is a *preset* one, i.e., Step 2 is executed irrespective of the result of Step 1. An *adaptive* type of universal diagnosis procedure can be also considered similarly.

D. Complexity of Universal Fault Diagnosis

The time required to perform a universal diagnosis procedure for an FPGA is referred to as *universal diagnosis complexity* of the FPGA. Suppose a universal diagnosis procedure DP for an FPGA G such that

$$DP = \langle (C_1, S_1), (C_2, S_2), \ldots, (C_{n_c}, S_{n_c}) \rangle$$

where n_c is the number of configurations. Let$c(i)$ be the number of configuration memory cells (CMCs) that are loaded to implement the i-th configuration C_i. Let $s(i)$ be the length of input sequence S_i for the i-th configuration C_i. The time required to implement all the configurations in universal diagnosis procedure DP for FPGA G is given by $T_G^C(DP) = \sum_{i=1}^{n_c} t_c c(i)$ where t_c is the time required to load one bit of a program in to a CMC in FPGA G. The time required to apply all the input sequences in test procedure DP for FPGA G is given by $T_G^S(DP) = \sum_{i=1}^{n_c} t_s s(i)$ where t_s is the clock cycle time of a configuration implemented in FPGA G. Thus, the universal diagnosis complexity of diagnosis procedure DP for FPGA G is given by

$$\begin{aligned} T_G(DP) &= T_G^C(DP) + T_G^S(DP) \\ &= \textstyle\sum_{i=1}^{n_c} (t_c c(i) + t_s s(i)). \qquad (3) \end{aligned}$$

Here, let us consider the universal diagnosis complexity of diagnosis procedure DP for SL-FPGAs.

Whenever a configuration is changed to another one on an SL-FPGA, all the program bits to be loaded are required. Hence, the time required to implement a configuration of a universal diagnosis procedure DP for an SL-FPGA is expressed as $c(i) = N_m$ for all i, where N_m is the total number of CMCs in FPGAG. As shown by Equation (2), the number of configurations in diagnosis procedure DP is expressed as

$$n_c = 2(2k+4) = O(\log n) \qquad (4)$$

where k is the number of input lines of an LUT and$n$ is the size of an LUT, i.e., $n = 2^k$.

Without loss of generality, the total number of CMCs in an FPGA is assumed to be proportional to the number of CLBs, N^2, the size of an LUT (or the number of CMCs in an LUT), 2^k, i.e., $N_m = O(N^2 n)$ where $n = 2^k$. Hence, the time required to implement all the configurations in DP_1 can be expressed as

$$\begin{aligned} T_{SL}^C(DP_1) &= t_c \times N_m \times O(\log n) \\ &= O(N^2 n \log n). \qquad (5) \end{aligned}$$

Recall the configurations in test procedure TP_{CLB} (Table 1). Note that each of configurations C_{2k+1}, C_{2k+2} and C_{2k+4} in TP_{CLB} implements a path including a DFF. Accordingly, each CLB cascade at the corresponding configurations in universal diagnosis procedure DP_1 includes a shift-register composed of N DFFs. Consequently, the output response for an input pattern can be observed N clock cycles later after applying the input pattern. From Table 1, the length $s(i)$ of input sequence S_i applied to the i-th configuration C_i in universal diagnosis procedure DP_1 can be expressed as

$$s(i) = \begin{cases} n & (1 \leq i \leq 2k,\ 2k+5 \leq i \leq 4k+4) \\ 6+N-1 & (i = 2k+1, 4k+5) \\ 4 & (i = 2k+3, 4k+7) \\ 4+N-1 & (i = 2k+2, 2k+4, 4k+6, 4k+8) \end{cases} \tag{6}$$

Thus, the total time required to apply the input sequences in universal diagnosis procedure DP_1 is expressed as

$$\begin{aligned} T^S_{SL}(DP_1) &= (4kn + (2N+10) + 8 + (4N+12))t_s \\ &= O(N + n \log n) \end{aligned} \tag{7}$$

From Equations (5) and (7), the complexity of universal diagnosis procedure DP_1 for SL-FPGAs is expressed as

$$T_{SL}(DP_1) = O(N^2 n \log n). \tag{8}$$

From this equation, we can see that the universal diagnosis complexity of diagnosis procedure DP_1 for SL-FPGAs depends on the array size N of the FPGAs. If we can make universal diagnosis complexity independent of the array size, the complexity is considerably reduced. In the next section, we present another universal diagnosis procedure and a class of FPGAs, called *C-diagnosable* FPGAs, for which the universal diagnosis complexity is independent of the array size.

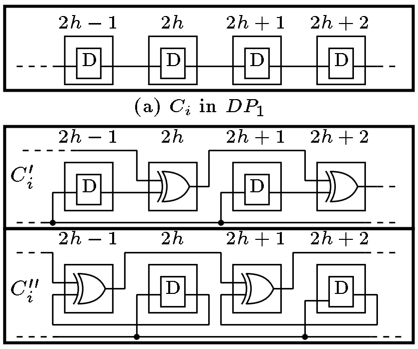
IV. C-Diagnosable FPGA

Since an FPGA consists of an array of CLBs, it can be considered to be an iterative system. *C-testability*[13] is a term which expresses an important class of testable iterative systems. In [7], the concept of C-testability for general LSIs has been extended to that for FPGAs, and C-testable FPGAs has been presented. Here we further extend the concept of C-testability to fault diagnosis for FPGAs and propose *C-diagnosable* FPGAs as follows.

Definition (C-diagnosable): An FPGA is said to be *C-diagnosable* if there exists a universal fault diagnosis procedure of which complexity is independent of the array size of the FPGA.

As shown by Equation (8), the universal diagnosis complexity of DP_1 depends on the array size of the FPGAs, and hence the FPGAs may not be C-diagnosable.

As shown by Equation (3), the complexity of a universal diagnosis procedure is the sum of the total implementation time of configurations and the total application time of input sequences. From Equations (5) and (7), we can see that both of the implementation time and the application time of diagnosis procedure DP_1 for SL-FPGAs depend on the array size. Here, we consider methods for making both times independent of the array size severally.



(a) C_i in DP_1

(b) C'_i and C''_i in DP_C

Fig. 4. Modification of DP_1 to DP_C. (a) Configuration C_i in DP_1. (b) Configurations C'_i and C''_i in DP_C.

First, let us consider the application time of input sequences in diagnosis procedure DP_1.

As mentioned in the previous section, there exist configurations that implement shift-registers in diagnosis procedure DP_1, and hence the application time for such configurations depends on the array size. Note that an LUT in CLBs can implement any k-input logic function. Hence, if the output lines of multiple CLBs are connected with the input lines of another CLB that implements an exclusive-OR logic (XOR), the output response of those CLBs (except the CLB implementing an XOR) can be observed from the output of the XOR without cascading DFFs.

Thus, we present another universal diagnosis procedure DP_C, which is obtained by modifying part of configurations in DP_1 as follows.

Universal diagnosis procedure: DP_C
The following two configurations are substituted for Configuration C_i in DP_1 for $i = 2k+1, 2k+2, 2k+4, 4k+5, 4k+6, 4k+8$ (Fig. 4).

(1) C'_i: For all rows m (resp. columns l):

- For $h = 1, 2, \ldots, N/2$, the CLB at the odd column (row), $CLB(2h-1, m)$ $(CLB(l, 2h-1))$, as well as its input lines implements the same configuration as C_i in diagnosis procedure DP_1.
- For $h = 1, 2, \ldots, N/2$, the CLB at the even column (row), $CLB(2h, j)$ $(CLB(i, 2h))$, implements a 2-input XOR.
- For $h = 1, 2, \ldots, N/2 - 1$, the output line of $CLB(2h, m)$ $(CLB(l, 2h))$ is connected with an input line of the XOR on $CLB(2h+2, m)$ $(CLB(l, 2h+2))$.
- The output line of the rightmost (bottommost) CLB, $CLB(N, m)$ $(CLB(l, N))$ is connected with an I/O block.

(2) C''_i: By exchanging the even CLBs for the odd ones, configurations are implemented in the same way as C'_i.

The above diagnosis procedure DP_C partitions CLBs into two groups; $N/2$ CLBs that are diagnosed and $N/2$ CLBs that configure $N/2$-input XOR to compress the output sequences from diagnosed FPGAs, and exchanges the role of these groups for each other by means of the doubled configurations. As a result, any configuration in DP_C includes no cascade of DFFs, and accordingly the output responses can be observed at just one cycle later after applying the corresponding input pattern.

Consequently, although the number of configurations increases, the application time becomes independent of the array size N. By substituting $2(6 \times 2)$ and $2(4 \times 4)$ for $(2N+10)$ and $(4N+12)$ in Equation (7), respectively, the total time required to apply the input sequences in DP_C can be expressed as

$$\begin{aligned} T^S_{SL}(DP_C) &= (4kn + 24 + 8 + 32)t_s \\ &= O(n \log n), \end{aligned} \qquad (9)$$

which is independent of the array size N.

Next, let us consider the configuration time of this universal fault diagnosis procedure DP_C.

At the twelve substituted configurations (i.e., C'_i and C''_i for $i = 2k+1, 2k+2, 2k+4, 4k+5, 4k+6, 4k+8$) in DP_C, a pair of a configuration to be diagnosed and an XOR is implemented alternately in each row or in each column. Hence, if we regard 2×2 CLBs ($CLB(l,m)$,$CLB(l,m{+}1)$,$CLB(l{+}1,m)$,$CLB(l{+}1,m{+}1)$ for $l,m = 1,3,5,...$) as one block, an iterative system is implemented at any of the substituted configurations. Also, each CLB implements the same logic function at any other configuration in diagnosis procedure DP_C. Therefore, we can see that universal diagnosis procedure DP_C always implement iterative systems such that each logic block consists of 2×2 CLBs. To such a diagnosis procedure, the programming scheme called *block-sliced loading*[7] can be adopted. An FPGA having block-sliced loading can load the same program into each block concurrently.

Let t_b be the time required to load the same program bit into the corresponding CMC in each block. The number N_b of CMCs in a block is expressed as

$$N_b = 2^2 O(n) = O(n),$$

where n is the size of an LUT. On the other hand, as mentioned above, when universal diagnosis procedure DP_C is derived from DP_1, six configurations in DP_1 are doubled, and hence, from Equation (4), the number of configurations in DP_C is expressed as

$$n_c = 2(2k+4) + 6 = O(\log n).$$

Hence, the time required to implement configurations in universal diagnosis procedure DP_C on a block-sliced SL-FPGA can be expressed as

$$\begin{aligned} T^C_{BSSL}(DP_C) &= t_b \times N_b \times O(\log n) \\ &= O(n \log n). \end{aligned} \qquad (10)$$

Note that the application time of input sequences does not depend on the programming scheme. Therefore, from Equations (9) and (10), the universal fault diagnosis complexity of DP_C for block-sliced SL-FPGAs can be expressed as

$$T_{BSSL}(DP_C) = O(n \log n).$$

Thus, we can obtain C-diagnosable FPGAs.

V. Conclusions

In this paper, we have introduced *universal fault diagnosis* such that when applied to an unprogrammed FPGA, it locates a fault in any faulty programmed FPGA corresponding to the unprogrammed FPGA. As an important class of FPGAs, we have proposed *C-diagnosable* FPGAs such that there exists a universal fault diagnosis procedure of which *universal fault diagnosis complexity* is independent of its array size.

Focusing on fault diagnosis of configurable logic blocks (CLB) in an look-up table FPGA, we have presented *universal fault diagnosis procedures* of which diagnostic resolution is 1 CLB, i.e., it can locate a fault just to one CLB. The complexity of the proposed universal fault diagnosis for FPGAs with the programming scheme called block-sliced loading is independent of the array size, i.e., C-diagnosable.

Acknowledgments

The authors would like to thank Profs. Takuji Okamoto, Tokumi Yokohira and Hiroyuki Michinishi of Okayama University, Japan, and Profs. Toshimitsu Masuzawa and Michiko Inoue of Nara Institute of Science and Technology, Japan for their helpful comments and discussions on this work.

References

[1] S.D. Brown, R.J. Francis, J. Rose, and S.G. Vranesic, *Field-Programmable Gate Arrays*, Kluwer Academic Publishers, 1992.

[2] S.M. Trimberger, Ed., *Field-Programmable Gate Array Technology*, Kluwer Academic Publishers, 1994.

[3] *The Programmable Logic Data Book*, Xilinx, 1994.

[4] M. Hermann and W. Hoffmann, "Fault modeling and test generation for FPGAs," in *Lecture Notes in Computer Science, Field Programmable Logic*, R.W. Hartenstein and M.Z. Servít, Eds. 1994, pp. 1–10, Springer-Verlag.

[5] R.O. Durate and M. Nicolaidis, "A test methodology applied to cellular logic programmable gate arrays," in *Lecture Notes in Computer Science, Field Programmable Logic*, R.W. Hartenstein and M.Z. Servít, Eds. 1994, pp. 11–22, Springer-Verlag.

[6] W.K. Huang and F. Lombardi, "An approach for testing programmable/configurable field programmable gate arrays," in Proc. *the 14th IEEE VLSI Test Symp.*, Apr. 1996, pp. 450–455.

[7] T. Inoue, H. Fujiwara, H. Michinishi, T. Yokohira, and T. Okamoto, "Universal test complexity of field-programmable gate arrays," in Proc. *the Fourth IEEE Asian Test Symp.*, Nov. 1995, pp. 259–265.

[8] H. Michinishi, T. Yokohira, T. Okamoto, T. Inoue, and H. Fujiwara, "A test methodology for interconnect structures of LUT-based FPGAs," in Proc. *the Fifth IEEE Asian Test Symp.*, Nov. 1996, pp. 68–74.

[9] M. Renovell, J. Figueras, and Y. Zorian, "Test of RAM-based FPGA: methodlogy and application to the interconnect," in Proc. *the 15th IEEE VLSI Test Symp.*, Apr. 1997, pp. 230–237.

[10] H. Michinishi, T. Yokohira, T. Okamoto, T. Inoue, and H. Fujiwara, "A test methodology for configurable logic blocks of look-up table based FPGAs," *IEICE Trans. D-I*, vol. J79-D-I, no. 12, pp. 1141–1150, Dec. 1996 (in Japanese).

[11] F. Lombardi, D. Ashen, X. Chen, and W.K. Huang, "Diagnosing programmable interconnect systems for FPGAs," in Proc. *Int. Symp. on FPGAs*, Feb. 1996, pp. 100–106.

[12] X.T. Chen, W.K. Huang, and F. Lombardi, "On the diagnosis of programmable interconnect systems: theory and application," in Proc. *the 14th IEEE VLSI Test Symp.*, Apr. 1996, pp. 204–209.

[13] A.D. Friendman, "Easily testable iterative systems," *IEEE Trans. Comput.*, vol. C-22, no. 12, pp. 1061–1064, Dec. 1973.