Efficient On-line Interconnect Testing in FPGAs with Provable Detectability for Multiple Faults *

Vishal Suthar and Shantanu Dutt
Dept. of ECE, University of Illinois-Chicago
{vsuthar, dutt}@ece.uic.edu

Abstract **We present a very effective on-line interconnect built-in-self-test (BIST) method I-BIST for FPGAs that uses a combination of the following novel techniques: a track-adjacent and a switch-adjacent (also called "mirror adjacent") pairwise net comparison mechanism that achieves high detectability, a carefully designed set of only five net-configurations that cover all types and locations of wire-segment and switch faults, a 2-phase global-detailed testing approach, and a divide-and-conquer technique used in detailed testing to quickly narrow down the set of potential suspect interconnects that are then detail-diagnosed. These techniques result in I-BIST having provable detectability in the presence of an unbounded number of multiple faults, very high diagnosability of 99-100% even for high fault densities of up to 10% that are expected in emerging nano-scale technologies, and much lower test times or fault latencies than the previous best interconnect BIST techniques. In particular, for application to on-line testing, our method re- quires $2n$ roving-tester (ROTE) configurations to test an entire $n \times n$ FPGA, while the previous best online interconnect BIST technique re- quires n^2 configurations. Thus, I-BIST is an order of magnitude more time- as well as power-efficient, and will scale well with rapidly increas- ing FPGA device sizes that are expected in emerging technologies.**

1 Introduction

An FPGA consists of an array of programmable logic blocks (PLBs) interconnected by a programmable routing network, consisting of wire-segments and switches, which we refer to as *interconnects*. Current as well as future double and single-digit nanometer FPGAs are susceptible to *permanent or hard faults*, both during fabrication and field operation. In order to isolate these faults using the re-programmability feature available in many FPGAs, each fault needs to be detected as well as diagnosed, i.e., located. Built-in self-test (BIST) approach to test FPGAs, wherein the test circuit is placed within the FPGA, has gained popularity due to its advantages of at-speed and low-cost testing. There are two types of BIST techniques, *offline* and *online*. In offline testing, the FPGA is tested while no application circuit is executing on it, while in online testing, the FPGA is tested with the application circuit mapped to and executing on it. In applications such as life critical medical appliances, space hardwares and remote installations, where the circuit mapped on the FPGA needs to run continuously with minimal or no interruption, online testing is a must [11].

Figure 1(a, b) shows how online testing is performed by testing only a small part of the FPGA at a time using a ROving TEster (ROTE). Each ROTE comprises of multiple built-in self-testers (BISTers) (Fig. 1[c]), each formed of a test pattern generator (TPG), which applies test vectors to two identically configured cells under test (CUTs) in the case of PLB testing, and two sets of wires-under-test (WUTs) in the case of interconnect testing. The outputs of the CUTs/WUTs are then compared by an output response analyzer (ORA), that checks for any discrepency in the two outputs, indicating a presence of fault(s).

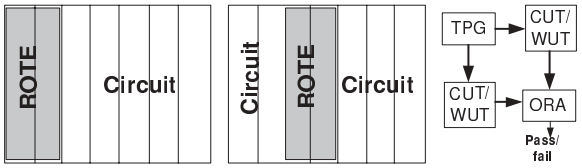

Figure 1: *Online testing concept: (a) Only a part of the FPGA (say, 2 columns) is under test using a ROTE, while the remaining part of the FPGA performs the circuit function. Once testing of this part is completed, the ROTE is reconfigured to test another part of the FPGA, as shown in (b). (c) A Built-in self-tester (BISTer) tile.*

Since interconnects occupy approximately 90% of the FPGA area, the chances of faults occuring in interconnects are much greater than that for PLBs. However, although there are many BIST techniques for PLB testing in FPGAs ([4, 6, 12, 13, 17] are some recent ones), there are very few BIST techniques for FPGA interconnects [1, 2, 8, 9, 14, 15]. Table 1 enumerates the salient features of these interconnect BIST techniques[1]. As implied by the table, none of the existing interconnect BIST techniques possess provable detectabilty in the presence of multiple faults, which can be quite frequent in emerging FPGA technologies. Thus none of these techniques are effective for the test and diagnosis of interconnect faults in nano-scale FPGAs.

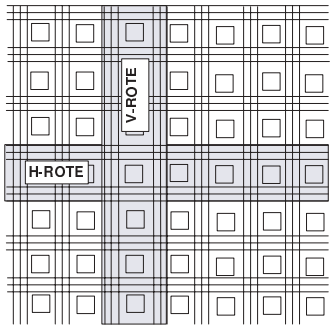

Figure 2: *Horizontal and Vertical ROTE*

In this paper we present a novel BIST technique for interconnects, applicable to both online and offline testing, that has provable detectability and empirically high-diagnosability in the presence of multiple faults. We refer to our technique as *Interconnect-BIST (I-BIST)*. We explain our technique with respect to online testing; the extension to offline testing is straight-forward. The types of interconnect faults considered are *wire stuck-at* 0/1 *fault*, *wire open fault*, *bridge fault* and *switch stuck-open/closed fault*. The types of bridge faults considered are *wired-OR* and *wired-AND*.

The rest of the paper is organized as follows. Section 2 discusses the general idea of the I-BIST technique, while Sec. 3 and 4 discusses the two phases, global and detailed testing, of I-BIST in detail. Section 5 provides simulation results and we conclude in Sec. 6.

*This work was supported in part by NSF grant CCR-0204097.

[1] k in the table is different for different techniques. It is, however, a function of the number of tracks per channel.

Technique	*Fault detection coverage*	*Fault diagnosis coverage*	*No. of Test vectors required*	*Applicable to online testing ?*	*Comments — diagnosability results, experimental results and test complexity*
Sun et. at. [15, 14]	50%	0%	$2k + 2^k$	No	No fault diagnosis results provided or proved.
Niamat et. al. [9]	$\approx 100\%$	$1/k \times 100\%$	2^k	No	No fault diagnosis results provided or proved.
Liu et al. [8]	$\approx 100\%$	$1/k \times 100\%$	$2k$	No	Theoritical proof for fault diagnosis applicable only to single fault in WUTs. No experimental results provided.
Stroud & Abramovici et. al. [1, 2]	$\approx 100\%$	$\geq 1/k \times 100\%$	2^k	Yes	Requires n^2 ROTE configurations and $O(log\ k)$ configurations in each ROTE position. No theoritical proof for fault diagnosis. Exp. results provided for max. 3 faults in entire FPGA area.

Table 1: *Comparison of different interconnect BIST techniques for FPGAs of size $n \times n$. k is the number of nets in a WUT.*

2 Interconnect-BIST (I-BIST) Technique

Before we explain our I-BIST technique, we discuss which interconnects will be tested under any ROTE area in online testing.

2.1 ROTE configuration

Typical $n \times n$ FPGAs have wire-segments of lengths one (single), two (double) and n (long), spanning both horizontally as well as vertically across the FPGA. A ROTE normally occupies 2 or 3 columns/rows in the FPGA. Hence to test both horizontal and vertical long lines we require a Horizontal-ROTE (H-ROTE) as well as a Vertical-ROTE (V-ROTE) [3]. Figure 3(a) shows the interconnects (wire-segments and switches) that will be tested under a particular V-ROTE position. As shown in Fig. 3(a), each V-ROTE area includes three channels (a single vertical or horizontal line in Fig. 3(a) represents an entire channel[2]). All vertical wires of these three channels plus the single and double length horizontal wires under the ROTE area will be tested for faults. All the switches in the middle channel (channel 2) will also be tested. For channels 1 (left) and 3 (right) only three switches of each switch-box will be tested. The remaining (untested) switches, except switch S_{ew}, of channels 1 and 3 are tested in either a previous (left) or subsequent (right) ROTE position. Switch S_{ew} of channels 1 and 3 are not tested in V-ROTE as it can interfere with the FPGA resources implementing the application circuit. Hence switch S_{ew} of channels 1 and 3, and, the horizontal long wires are tested in H-ROTE. We refer to the set of interconnects tested under a particular V-ROTE (H-ROTE) position as the *V-set (H-set)*. Also, we refer to the set of i'th wiring tracks in each of the three vertical channels under the ROTE area along with their associated horizontal wire-segments and switches, as *track-set* i. We next discuss the general idea of the I-BIST technique. I-BIST is discussed in this paper for single-length track segments but is easily extended to multiple-length track segments.

2.2 Main Concepts of I-BIST

The main objectives of an effective BIST technique are:

1. To maximize diagnosability, even in presence of multiple faults.
2. To reduce test time (i.e., fault latency), and especially the dominant component of test time, *the configuration time*, i.e., the time to set LUT or switch configuration bits. Hence the main objective is to minimize the number of configurations required for testing interconnects. The secondary objective is to reduce the number of test vectors, used in each configuration.

The I-BIST technique achieves these objectives using a two-phase test-and-diagnosis scheme, wherein, in the first phase, *global testing*, the possible fault locations are isolated to a small set of interconnects, termed the *suspect set*, using very few configurations. In the second phase, *detailed testing*, the actual faulty interconnects are identified from among the suspect set.

[2]A channel is a set of tracks, each formed of multiple wire-segments and switches, spanning either horizontally or vertically across the entire FPGA.

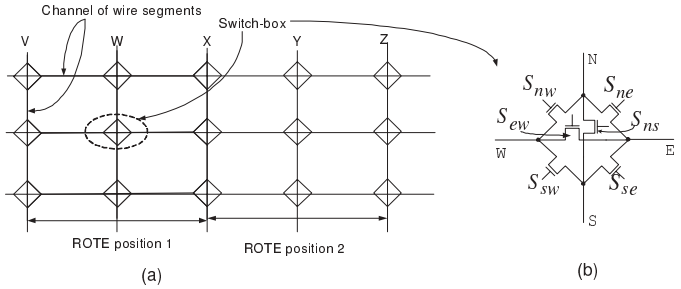

Figure 3: *(a) Interconnects tested under a ROTE position. V, W, X, Y and Z represent vertical channels. The interconnect channels and switches tested in ROTE position 1 are shown by darker/thicker lines. (b) Switch-box structure and nomenclature.*

We next discuss the basic concepts for testing interconnect faults that we use in both the testing phases. In our approach, nets formed of multiple interconnected wire-segments and switches belonging to the same track-set, are configured (Fig. 4[a]). For the global testing phase the nets are configured to span the entire length of the FPGA in the V-ROTE area (entire width of the FPGA in the H-ROTE area), while in detailed testing, which is composed of a series of more specific targeting of smaller interconnect sets, the nets span smaller sub-lengths of the FPGA. Two adjacent nets act as a WUT pair and are tested by passing 2-bit test vectors from the TPG. Their outputs are compared using an ORA configured as a 2-bit XOR. We now discuss how each type of interconnect fault, present in a WUT pair, are detected by our approach. Note that complementary-bit test vectors $\{01, 10\}$ applied to a fault-free WUT pair will result in the ORA output of 1, while for identical-bit test vectors $\{00, 11\}$, the ORA output will be 0.

Bridge fault: A bridge fault between two WUTs (nets) will result in both the WUTs having the same value for any test vector, resulting in an ORA output of 0. Hence applying complementary-bit test vectors 01 or 10 will result in detection of bridge fault(s).

Stuck-at-$0/1$ (s-a-$0/1$) fault: A stuck-at fault on either one of the two WUTs will result in both nets having identical values for one of the two complementary-bit test vectors, 01 or 10, thus resulting in idenfication of this fault. However, if the two WUTs have complementary stuck-at faults, i.e., n_1 is s-a-0 and n_2 is s-a-1, then both the nets will have complementary values for any test vector. Thus in this particular case we require an additional identical-bit test vector, 00 or 11, that will result in an ORA output of 1 for this faulty scenario, as opposed to the fault-free output of 0. Thus three test vectors 01, 10 and 00 (or 11) are required to identify all possible stuck-at fault patterns in

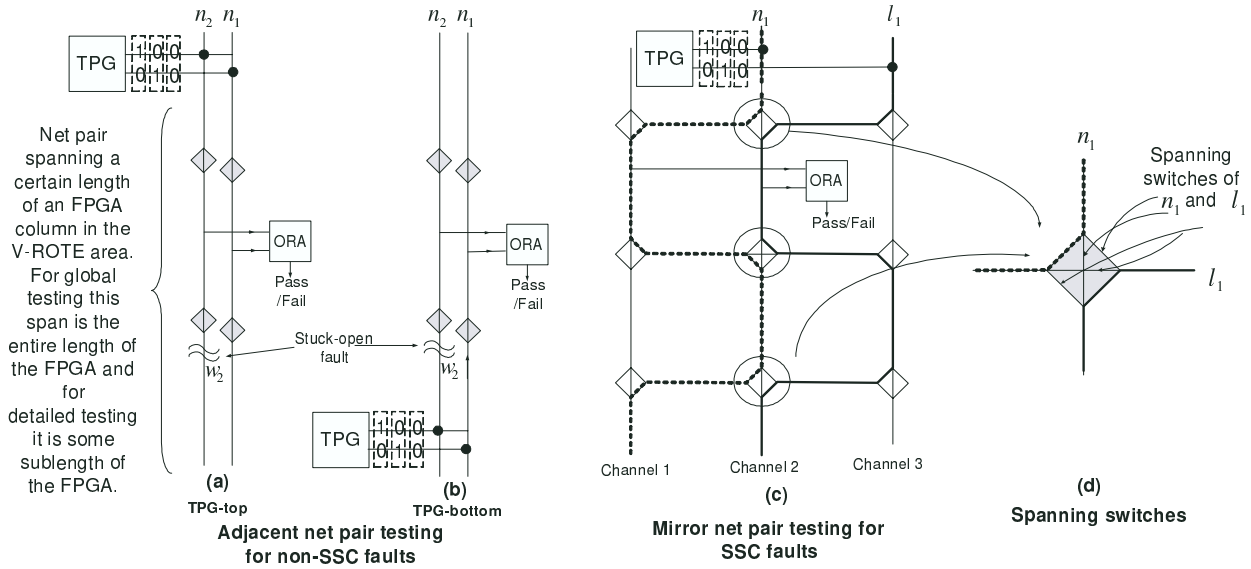

Figure 4: *Basic mechanism of the I-BIST technique. (a) and (b) The two stages of testing,TPG-top and TPG-bottom, required for detecting bridge, stuck-at and stuck-open faults (combinedly referred to as non-SSC faults). (c) Mirror net pair testing required for switch-stuck-closed (SSC) fault. (d) Spanning switches between nets n_1 and l_1 whose stuck-closed faults are detected as they create a bridge fault between n_1 and l_1.*

the WUT pair.

Stuck-open fault: A bridge or stuck-at fault in a WUT pair will result in an erroneous value on the entire net(s) in the WUT pair, irrespective of the location of the fault. However, a stuck-open fault in either a wire-segment or a switch will only be detected if that wire-segment or switch is present on the part of a net between the TPG and the corresponding ORA, because unlike stuck-at or bridge faults, a stuck-open fault will not result in an erroneous value on the entire net but will only isolate the portion of the net that lies between the point of its occurrence and the sink of the net. E.g., in Fig. 4(a), a stuck-at-x fault on wire-segment w_2 will result in the entire net n_2 having value x and hence this fault will be detected, whereas a stuck-open fault present at the same location w_2 will not affect the portion of the net between the TPG and the ORA, resulting in non-detection of this fault by the ORA. Thus two stages of testing are required, with the TPG positions in these stages being at the opposite ends of the WUTs[3]; see Figs. 4(a) and (b). The ORA's position in these two stages remains unaltered[4]. A stuck-open fault present on the portion of the net above the ORA will be detected in stage 1, while a stuck-open fault present in the portion of the net below the ORA will be detected in stage 2. Stuck-open fault(s) present on either or both the nets in the WUT pair will result in an incorrect ORA output for at least one of the two test vectors 01 and 10 in at least one of the two stages, and hence will be detected. We will refer to the two stages of testing as *TPG-top* and *TPG-bottom*.

Switch stuck-closed (SSC) fault: Unlike the above mentioned faults, an SSC fault cannot be detected by including the corresponding switch on a net, as the switches included on any net are deliberately kept closed. The only fault a stuck-closed switch can produce is that it will bridge the two wire-segments that it spans (i.e., can connect). Thus, forming nets within a channel, as shown in Figs. 4(a, b), will not aid in detecting SSC faults, since no two nets in the same channel are spanned by a switchbox swich. In order to detect SSC faults we introduce the concept of *mirror nets*. Mirror nets are mirror-image nets in different channels that share at least one common switch box and are spanned by an open switch in each common switchbox. In other words, these nets are formed in such a way that an SSC fault in the switch(es) under consideration will result in bridging of these mirror nets. E.g., as shown in Fig. 4(c), nets n_1 and l_1 are mirror nets that form a WUT pair; a presence of an SSC fault on the switches pointed by the arrows will result in bridging of these two nets. This situation is then similar to wire-segment bridge faults discussed earlier. Hence the SSC fault(s) in the switch(es) under consideration can then be detected by passing complementary-bit test vectors on the two mirror nets and comparing their outputs using an ORA. We term such a switch, that bridges two nets on the occurrence of a stuck-closed fault in it, as a *spanning* switch. Also, we will combinedly refer to all faults except switch stuck-closed faults (SSC) as *non-SSC* faults.

THEOREM 1 *Any number of faults in each net tested by the pair-wise adjacent-net comparison approach discussed above will be detected. Furthermore, a stuck-closed fault in any spanning switch between every pair of mirror nets compared by the mirror-pair approach will also be detected.*

Proof Outline: In the case of no faults in the two nets in the WUT pair being compared by an ORA, the ORA will output $(1, 1, 0)$ for the three test vectors 01, 10 and 00. It follows from the construction of our testing technique for the WUT pair that if there are faults in either net in it and/or a stuck-closed fault in any spanning switch between the two nets in the WUT pair, the ORA output will be different than $(1, 1, 0)$. Note that due to the application of identical-bit as well complementary-bit test vectors, there are no masking (aliasing) faults between the two nets in the WUT pair across all test vectors. $\diamondsuit$

Thus, the basic mechanism of the I-BIST technique, discussed above, is able to detect all types of faults in a pair of nets and requires only three test vectors $\{01, 10, 00\}$. We next discuss how this mechanism is used in the global and detailed

[3] If enough PLB resources are available, the top and bottom TPGs can operate simultaneously, synchronously generating the same test vectors; this reduces the number of configurations needed by half.

[4] The reason for the ORAs fixed position will be clear when we discuss global testing in detail in the next section.

testing phases of the I-BIST technique.

3 Global Testing

The aim of the global testing phase is to first detect faults, and then isolate the possible fault locations to a small number of *suspect* interconnects, thus making the task of the detailed testing phase simpler. Also this aim needs to be achieved in a minimum number of configurations as mentioned in objective 2 in Sec. 2.2. We next discuss our *Simultaneous adjacent-&-mirror pair comparison (SAMP)* technique, based on the mechanisms discussed in Sec. 2.2, that allows detection of faults on all the wire-segments and switches under the ROTE area in just 5 configurations.

3.1 Simultaneous adjacent-&-mirror pair comparison technique

Figure 5 shows one of the five configurations of the global testing phase, that we will use to explain our technique.

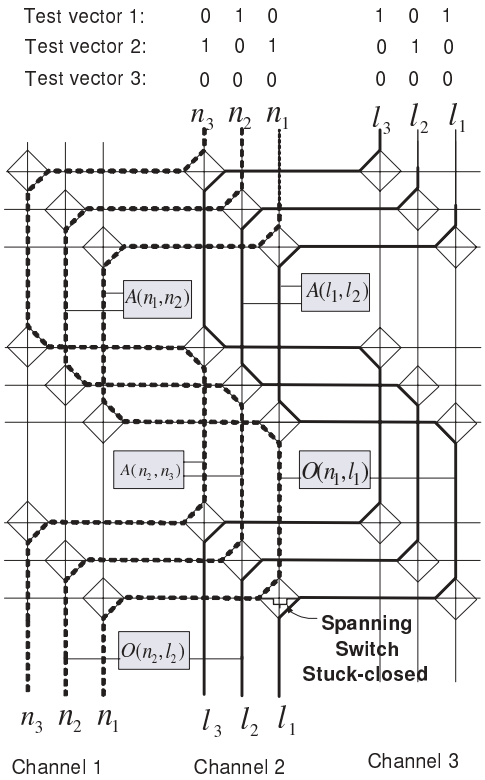

Figure 5: *Configuration* 1 *of the global testing phase.*

As shown in Fig. 5, the basic mechanism discussed in Sec. 2.2 for a pair of adjacent and mirror nets, is simultaneously applied to all adjacent and mirror net pairs in the ROTE area using the same TPG to generate the replicated test vectors that are applied to all these compared net pairs but using distinct ORAs to evaluate their outputs. We call a set of similarly configured nets in each channel a *net-set*. Hence in Fig. 5, the nets n_1, n_2 and n_3 shown by dashed lines form one net-set while the nets l_1, l_2 and l_3 shown by continuous lines form another net-set. In order to detect interconnect faults, two types of net comparisons are performed:

1. **Intra-net-set comparison:** Each pair of adjacent nets in a net-set is compared by an ORA configured as a 2-bit XOR. E.g., in Fig. 5, A(n_1, n_2) represents ORA comparing nets n_1 and n_2; A(n_2, n_3) represents ORA comparing nets 2 and 3 of the net-set represented by the dashed lines. Similarly, all adjacent pairs of nets of the net-set represented by continuous lines are also compared. This comparison is targeted to detect non-SSC faults. E.g., a switch-open fault or wire-segment fault(s) on net n_2 of the dashed net-set in Fig. 5 will be detected by ORAs A(n_1, n_2) and A(n_2, n_3). We will refer to the ORAs used in intra-net-set comparison as *intra-net-set ORAs*.

2. **Inter-net-set comparison:** Mirror nets i of the two net-sets are compared to each other by an ORA configured as a 2-bit XOR. E.g., O(n_1, l_1) represents ORA comparing mirror nets n_1 and l_1 of the two net-sets. This comparison is targeted to detect SSC faults, i.e., switch stuck-closed faults among the spanning switches between the nets of the two net-sets. E.g., a switch stuck-closed fault in the spanning switch S_{ew} of track 1 of channel 2, shown in Fig. 5, will result in a short between nets n_1 and l_1 of the two net-sets; hence this fault will be detected by the ORA O(n_1,l_1). We will refer to the ORAs used in inter-net-set comparison as *inter-net-set ORAs*.

With regards to the configuration details discussed above, we next describe how the design of the five configurations is derived on the basis of the possible interconnect faults. Note that, as discussed earlier, to detect non-SSC faults on a particular interconnect, the necessary and sufficient condition is to include this interconnect on a net in at least one configuration, while to detect SSC faults in a spanning switch, the necessary and sufficient condition is that each of the two compared nets from each of the two net-sets should contain exactly one of the two wire-segments on either side of the spanning switch.

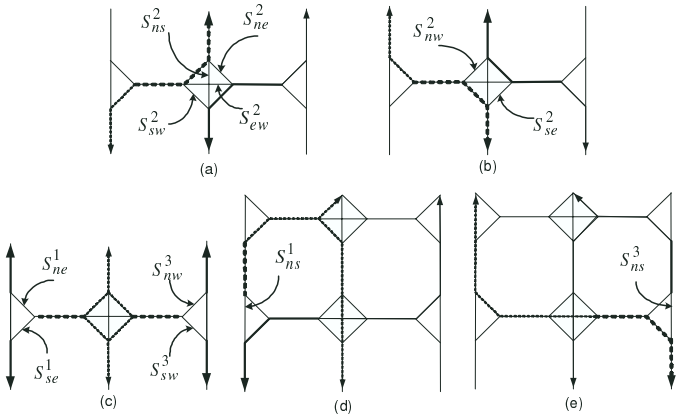

Figure 6: *Local views of configurations 1 to 5 of the global testing phase are given in (a) to (e), respectively. Dashed and continous dense lines of each net indicate how the nets are designed to detect stuck-closed faults in the spanning switches of the relevant local switch-boxes while the lighter lines indicate how the nets are extended in the rest of the ROTE area.*

Fig. 6 shows the localized view of the five configurations. As in Fig. 5, in this figure too the dashed and continuous lines indicate nets of two different net-sets. The dense lines in the figure indicate the localized view of the net design required for detecting switch stuck-closed fault(s) in the spanning switches of local switch-box(es), while the lighter (arrowed) lines indicate how the dense lines are extended on either side in order to span the entire ROTE area. We explain how all the switch faults are detected in these configurations. In order to detect stuck-closed faults in switches S^2_{ns} of channel 2, the wire-segments on

the north and the south side of the switch should be contained in different nets[5]; similarly for detecting stuck-closed faults in switches S^2_{ew} of channel 2, the wire-segments on the east and the west side of the switch should be contained in different nets. The design in which both these requirements can be met is the one corresponding to configuration 1 and is indicated by the dense lines in Fig. 6(a). As switches S^2_{sw} and S^2_{ne} of channel 2 also have the wire-segments they span in different nets of configuration 1, hence stuck-closed faults in these switches will also be detected.

In order to detect stuck-closed faults in the remaining switches of channel 2, i.e., switches S^2_{nw} and S^2_{se}, a mirror-image design of configuration 1 is required. This design corresponds to configuration 2 shown in Fig. 6(b). Thus, configurations 1 and 2 together can detect all switch faults of channel 2.

In order to detect stuck-closed faults in switches S^1_{ne} and S^1_{se} of channel 1 and switches S^3_{nw} and S^3_{sw} of channel 3, configuration 3 (Fig. 6(c)) is required.

Stuck-closed faults in switches S^1_{ns} and S^3_{ns} of channels 1 and 3 are respectively detected in configurations 4 and 5, as indicated by the continuous lines in Figs. 6(d) and (e).

Once the nets required to detect switch stuck-closed faults are designed (shown by dense lines in Fig. 6(a) - (e)), the remaining task is to extend the nets in each of the configurations in such a way that it passes through each interconnect in at least one configuration, as per the necessary and sufficient condition of detecting non-SSC faults. This is shown by the lighter (arrowed) lines in Fig. 6(a) - (e), and fully shown for configuration 1 in Fig. 5.

THEOREM 2 *Any number of fault(s) in the V-set (i.e., set of interconnects under a V-ROTE area) will be detected by the SAMP technique across the five configurations of the global testing phase.*

Proof: From the above discussion, the five configurations of the global testing phase include all the wire-segments as well as all switch-box switches in both connected and spanning modes in the ROTE area. Thus all wire segments and swtitches are tested for all possible faults, and from Theorem 1 they will be detected irrespective of their numbers. $\diamondsuit$

4 Detailed Testing

Detailed testing is the second phase of interconnect testing (I-BIST) wherein we diagnose faults in the nets detected as faulty in either of the five configurations of the global testing phase.

For further reference, with respect to a particular track-set, we will refer to a configuration of the global testing phase as a *failed configuration* if at least one of the intra-net or inter-net ORAs in that configuration has an output of fail. The global testing phase with one or more failed configurations for a particular track-set will be referred to as failed global testing phase, else it will be refer to as passed global testing phase for that track-set.

One simple approach to performing detailed testing is to form a suspect set that consists of all the interconnects in the ROTE area minus the interconnects of the nets corresponding to the passed global configurations. Each element of this suspect set is then diagnosed by comparing it with a fault-free element. This approach, however, is inefficient in terms of the fault latency, as the size of the suspect set may be large. E.g., a single stuck-at fault on a horizontal wire-segment will lead to a failure in all the five configurations and all the interconnect components of the corresponding track-sets will be the elements of the suspect set. We will refer to this approach of diagnosing the elements of the suspect set as the *flat intersection* approach.

[5]Superscript 2 in S^2_{ns} indicates the channel number, while subscript ns (for north-south) indicates the wire-segments' locations (at the north and south ends of the switch) that the switch can connect.

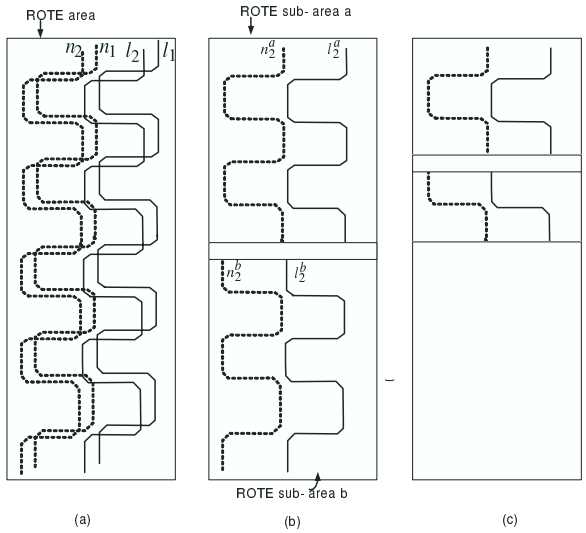

Figure 7: *High-level view of the divide-and-conquer approach of the detailed testing phase. (a) A configuration in the global testing phase.(b) The sub-divided configurations correspond to the configuration in part [a] in the detailed testing of the failed nets n_2 and l_2. Both these nets are broken into two parts, n_2^a and n_2^b for net n_2, and l_2^a and l_2^b for net l_2, and the corresponding failed global test configurations are independently and simultaneously applied on these two nets. (c) The second iteration of the D&C approach; as the ROTE sub-area b is found to be fault-free in the earlier iteration, it is not required to be tested further.*

In order to overcome this problem of high detailed test time (or fault latency), we propose another approach based on the *divide-and-conquer* (D&C) paradigm. In this approach, in case of a failed global testing phase corresponding to a particular track-set, we iteratively divide the ROTE area into two halves and apply the failed global test configuration(s) on the two subsets of the failed track-set located in these two ROTE sub-areas. E.g., as shown in Fig. 7(a), track-set 2 containing nets n_2 and l_2, that failed in the global testing phase in cofiguration 1 (because of a fault in net n_2 or l_2 or the spanning switches between them), is divided into two sub-track-sets each belonging to one of the two ROTE sub-areas a and b, and then configuration 1 along with the SAMP technique are applied simultaneously to the two sub-track-sets (Fig. 7(b)). The sub-track-set(s) that fail are further divided into two halves and the above procedure is repeated (Fig. 7(c)) until the size of the ROTE sub-area spans two rows, the minimum area required to apply the global test configurations. Also, as was implicit in the above example, a sub-track-set in a particular sub-area needs to be tested for only those configurations that failed in the global testing of its root track-set. E.g., in Fig. 7(a), if track-set 2 failed for configurations 1 and 2 only, then the two sub-track-sets need to be tested for configurations 1 and 2 only. This is because the components of track-set 2 that correspond to passed configurations (3-5) of the global testing phase are guaranteed to be fault-free.

The nets in each sub-track-set in the D&C approach are also tested in the same way, using intra-net-set and inter-net-set ORAs, as in the global testing phase. However, if in a failed

Techniques	*Fault detection efficiency*	*Fault diagnosis efficiency*	*No. of Test vectors required*	*Applicable to online testing ?*	*No. of ROTE configurations in online testing*	*No. of WUT configurations in online testing*
Previous best	$\approx 100\%$ [1, 8, 9]	$\geq 1/k \times 100\%$ [1]	$2k$ [8]	Yes [1]	n^2 [1]	$O(log\ k)$ [1]
I-BIST	100%	$\approx 100\%$	3	Yes	$2n$	$O(1)$

Table 2: *Comparison of previous best interconnect BIST techniques with our I-BIST technique. n is the number of rows/columns in an FPGA, while k is the number of nets in a WUT (wires-under-test) and is a function of the # of tracks/channel.*

global testing configuration, for any adjacent track-sets, i and $(i+1)$, the intra-net testing of nets (n_i, n_{i+1}) of these track-sets results in a pass output, then it implies that there is no wire-segment bridge fault between these two nets or any other interconnect fault in any of these two nets. Hence in detailed testing (i.e., in the D&C approach) the nets formed in each ROTE sub-area are tested only with inter-net-set ORAs for testing inter-net pairs (n_i, l_i) and (n_{i+1}, l_{i+1}) of track-sets i and $(i+1)$; intra-net-set ORAs are not required in this case.

After the fault(s) are confined to the minimum possible ROTE sub-areas (i.e., 2-row-length sub-areas), the interconnect components in each failed configurations of these sub-areas constitute the final suspect sets. Note that the switches that lie between any two sub-areas do not get diagnosed while diagnosing the interconnects of the two sub-areas, and hence these switches are also included in the final suspect sets. The elements of the suspect sets of each of the minimal sub-areas are then tested by comparison with a fault-free element. However, since the number of components in the minimal sub-area is far less compared to that of the entire ROTE area, the size of the suspect set is small. Thus the fault latency of D&C approach is much smaller compared to that of the flat intersection approach.

5 Simulation and Analytical Results

A 32×32 FPGA array with 16 tracks/channel, each track formed of single-length wire-segments, was functionally simulated in C++. Different types of interconnect faults were inserted randomly in the wire-segments and switches. Table 3 gives the fault coverage (% of faults that are correctly diagnosed) results for different fault densities (% of interconnect blocks that are faulty. The internal fault density of each interconnect block was randomly selected from $\approx 5\%$ to $\approx 20\%$). Table 3 affirms the high-diagnosability of our I-BIST technique.

Fault density (%)	1 to 4	5	6	7	8	9	10
Fault coverage (%)	100	99.8	99.7	99.7	99.6	99.6	99.3

Table 3: *Fault coverage vs fault density results for I-BIST.*

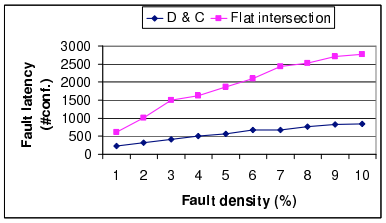

Figure 8: *Fault latency results for I-BIST for divide and conquer (D&C) approach and flat intersection approach. The unit of latency is the number of configurations.*

Fig. 8 compares the fault latencies of the two approaches to detailed testing, divide and conquer (D&C), and the flat intersection approach. The results clearly indicate that D&C is a better approach in terms of fault latency; fault coverages are the same for both techniques.

Table 2 compares different parameters of our technique with the previous best parameters of techniques that were in Table 1. As demonstrated in the table, the fault latency (i.e., test time) of our technique is much lower than those of previous interconnect BIST techniques. Our technique requires only 5 configurations per ROTE area for detection of faults (i.e., during global testing) and only 3 test vectors to diagnose faults. Compared to this, the previous best interconnect BIST technique requires $O(\log k)$ configurations per ROTE area [1], and $2k$ test vectors [8], respectively, where k is the number of nets in a WUT and is a function of the number of tracks/channel. Furthermore, for on-line testing, we need only $2n$ ROTE configurations in an $n \times n$ FPGA, while the previous best on-line interconnect BIST method [1] requires n^2 configurations—all these add up to much lower test times as well as power consumption for I-BIST compared to the technique in [1].

6 Conclusions

We presented a new BIST technique, I-BIST, for testing interconnects of an FPGA, which was theoretically proven to to be able to detect any number of mutiple faults, and empirically shown to have a very high-diagnosability even at high fault densities of up to 10%. Our BIST technique is applicable to both offline as well as online testing (testing concurrently with the circuit operation). To the best of our knowledge, this is the first BIST technique that alleviates the limited diagnosability of only one fault per WUT (wires-under-test) of previous interconnect BIST techniques. I-BIST is thus a very effective interconnect testing technique for current as well as emerging nano-scale FPGAs that are expected to have high fault densities. The fault latency or test time of our technique is also much lower than those of previous interconnect BIST techniques as demonstrated in Table 2, and should be scalable with the large FPGA device sizes expected in emerging technologies.

References

[1] M. Abramovici et al., "On-line BIST & Diag. of FPGA Interconnect Using Roving STARs", Proc. IEEE Int'l On-line Test. Workshop, p. 27-33, 2001.

[2] M. Abramovici, C. Stroud, B. Skaggs and J. Emmert, "Improving on-line BIST-based diagnosis for roving STARs", Proc. 6th IEEE Int'l On-line Testing Workshop, pp. 31-39, 2000.

[3] M. Abramovici, C. Stroud, S. Wijesuriya and V. Verma, "Using Roving STARs for On-Line Testing and Diagnosis of FPGAs in Fault-Tolerant Applications", *Proc. IEEE Int'l Test Conf.*, Sept'99.

[4] C. Hamilton et al., "Enhanced BIST-based Diagnosis of FPGAs via Boundary Scan Access", Proc. IEEE VLSI Test Symp., pp. 413-418, 1999.

[5] W.K. Huang and F.Lombardi, "An approach to testing of Programmable/Configurable FPGAs", Proc. IEEE VLSI Test Symp. , pp. 450-455, 1996

[6] W.K. Huang, F.J.Meyer, X. Chen and F. Lombardi, "Testing configurable LUT-Based FPGAs", IEEE Trans. VLSI systems, Vol. 6, No. 2, pp 276-283

[7] T.Inoue and H. Fujiwara, "Universal Fault Diagnosis for lookup table FPGAs", IEEE Design & Test of Compt.,Vol 15, No.1, Jan 1998

[8] J. Liu and S. Simmons, "BIST-Diagnosis of Interconnect Fault Locations in FPGAs", CCECE 2003.

[9] M.Y. Niamat et. al., "A BIST scheme for testing interconnects of SRAM-Based FPGAs", Midwest Symposium, Vol 2, Aug 2003. vspace*-0.1in

[10] N. R. Shnidman, W. H. Mangione-Smith, and M. Potkonjak, "On-line Fault Detection for Bus-Based Field Programmable Gate Arrays," IEEE Trans. On VLSI systems, Vol. 6, No. 4, pp 656-666, Dec 1998 .

[11] A.Steininger and C. Scherrer, "On the Necessity of On-Line BIST in Safety-Critical Applications", Proc. 29th Fault Tolerant Computing Symp., pp. 208-215, 1999.

[12] C. Stroud, E. Lee and M. Abramovici, "BIST-based diagnostics of FPGA Logic Blocks", Proc. IEEE Int'l Test Conf., pp. 539-547, 1997

[13] C. Stroud et al., "Built-In Self-Test for Programmable Logic Blocks in FPGAs (Finally a Free Lunch: BIST Without Overhead!)", Proc. IEEE VLSI TEST Symp., pp. 387-392, 1996.

[14] X. Sun, S. Xu, J. Xu and P. Troubort, "Design and Implementation of a Parity-Based BIST Scheme for FPGA Global Interconnects", CCECE 2001.

[15] X. Sun, S. Xu, B. Chan and P. Troubort, "Novel Technique for Built-In Self-Test of FPGA Interconnects", Proc. IEEE Int'l Test Conf., pp. 795-803, 2000

[16] V. Suthar and S. Dutt, "High-Diagnosability Online Built-In Self-Test of FPGAs via Iterative Bootstrapping", GLSVLSI, pp.78-83, 2005.

[17] V. Verma, S. Dutt and V. Suthar, "Efficient On-line Testing of FPGAs with Provable Diagnosabilities", Proc. DAC, pp. 498-503, 2004.