Efficient Built-in Self Test of Regular Logic Characterization Vehicles

Ben Niewenhuis and R. D. (Shawn) Blanton
Department of Electrical and Computer Engineering
Carnegie Mellon University, Pittsburgh, PA 15213
http://www.ece.cmu.edu/~actl/

Abstract—Fast and efficient analysis of test chips is crucial for effective yield learning. Prior work proposed the Carnegie-Mellon logic characterization vehicle (CM-LCV) as an improved test chip for yield learning. The highly regular nature of the CM-LCV test chip is particularly appealing for BIST; the current work describes a BIST scheme that achieves 100% input-pattern fault coverage with an 86.9% reduction in test time for a reference design. Furthermore, all of these properties are achieved with a minimal hardware overhead.

I. INTRODUCTION

Test chips broadly encompass any design fabricated for purposes other than use or sale. Prior work has established a distinct category of product-like test chips, that is, integrated circuits that share features and some functionality with commercial products, but are fabricated to gather information about the design and manufacturing process. This current work builds on the Carnegie Mellon logic characterization vehicle (CM-LCV), which proposed removing the product functionality from product-like test chips while maintaining a similar physical structure [1]. This approach allows the CM-LCV to be designed with optimal test and diagnosis properties without compromising its ability to extract relevant information about the semiconductor design and manufacturing process.

In this paper the unique characteristics of the CM-LCV are used to create an efficient built-in self test (BIST) system based on the circular BIST concept [2]. Several proofs are supplied for existence of the crucial properties of the CM-LCV. Additionally, a reference design is constructed and simulated to support the claims made concerning fault coverage and test set size.

The remainder of the paper is organized as follows: Section II provides background on LCVs and BIST. Section III discusses the properties of the CM-LCV relevant to this paper and describes the implementation of the BIST system. Section IV describes in detail the experiments used to evaluate this new BIST architecture while Section V concludes the paper.

II. BACKGROUND

This section presents the background relevant to the current work. Specifically, Section II-A describes the current state-of-the-art for test chips. Section II-B then provides an overview of built-in self-test techniques, with particular emphasis on circular BIST.

A. Logic Characterization Vehicle

The concept of a test chip encompasses any design fabricated with the primary goal of gathering actionable feedback about the manufacturing process as opposed to commercial sale. The traditional test chip is a collection of specialized test structures that are optimized for gathering specific information with high precision. Examples include comb- and serpentine-shaped interconnect used for measuring defect size and density distributions [3][4]. However, the increasing complexity of the IC design and manufacturing process in the nanoscale regimes has resulted in the emergence of systematic defects. Systematic defects are failures caused by complex interactions between the design and the fabrication process which result in significant yield loss whenever specific features exist in a design. Traditional test chips are ill-suited for uncovering systematic defects because they do not include many of the design features of an actual product. This has led to the development of the product-like test chip, which is composed of portions of actual product designs, utilizes cells from standard-cell libraries, and is created using typical design flows (synthesis, place-and-route, etc.). While many of these product-like test chip designs are ad hoc in nature, Hess et. al. introduced a more systematic approach in their Logic Characterization Vehicle (LCV) [5]. Their methodology is primarily focused on a jig (essentially a ring of logic) that applies logical and parametric tests to any combinational circuit placed inside the jig. Their jig can be considered to be a form of DFT since it provides the tester different access modes to the circuit under test. The flexibility and extensibility of this jig enables easy creation of an LCV using any product sub-circuit.

The CM-LCV is centered on the concept of removing the product functionality from the test chip design; the key insight is that the manufacturing process is sensitive only to the physical features of a design, not its functionality. Although the functionality (logic) and physical features (layout) are related, they can be sufficiently separated to maximize testability and diagnosability through careful design of the logic function while at the same time ensuring that the layout mimics the targeted product or family of products. To achieve the goals of maximizing testability and diagnosability the following test-chip characteristics are adopted:

- Regularity - A set of functional unit blocks (FUBs) with regular connections can be designed to be C-testable
Fig. 1. VH-bijectivity demonstrated in the presence of a defective FUB in a two-dimensional array. Note that signal propagation moves left to right, top to bottom.

[6][7][8], a property which provides strong guarantees about the test-set size and fault coverage over the entire array regardless of its size.

- Two-dimensional array - Arranging FUBs in a two-dimensional array with vertical and horizontal connections enables propagation of fault effects in two directions, allowing for better localization of defects within the array and within a defective FUB.
- Bijective FUBs - Constraining the FUB function to be bijective guarantees propagation of fault effects and simplifies test set construction.

In addition to these characteristics, the FUBs are designed to exhibit VH-bijectivity, a more constrained form of bijectivity that propagates an error residing exclusively on either the horizontal or vertical input of the FUB to both the vertical and horizontal FUB outputs. Figure 1 demonstrates how VH-bijectivity controls error propagation within a FUB array.

Assume some FUB instance in the array of Figure 1 is defective (marked with the red “X”), and produces an erroneous value (represented by the red-crossed connection) for some test along its horizontal output. All of the FUBs in the same row as the defective module (marked with an “A”) fall under the horizontal case of the VH-bijective property: an error only present at a horizontal input will propagate to both outputs. Furthermore, all FUBs in the first downstream column (marked with a “B”) fall under the vertical case of the VH-bijective property: an error that appears only at a vertical input will propagate to both outputs. The remaining downstream FUBs may experience errors on both inputs; because they are bijective the error will propagate to at least one of the outputs, but this case is not covered by VH-bijectivity and thus the error propagation is unpredictable without precise knowledge of the defect behavior (represented by the blue-dotted connections). Regardless, the array location indicated by the first row/column errors observed must be either (a) the site of the defect, (b) horizontally adjacent to the defect, or (c) vertically adjacent to the defect, resulting in near-perfect diagnostic resolution for a single defective FUB.

Fig. 2. Notation used for (a) the inputs and outputs of a single FUB and (b) the inputs and outputs of an entire array of FUBs.

B. Built-In Self Test

BIST is a circuit design methodology that seeks to reduce the cost of test by building some portion of the test flow into the circuit under test itself. BIST is thus a balancing act between the savings generated on the testing side with the incurred hardware and performance overhead. It is important to note that the output produced by most BIST schemes is not ideal for actual diagnosis; instead, BIST is typically used as a simple pass/fail indicator. Regular designs (e.g. memories [9], adders [10], multipliers [11], etc.) have always been appealing targets for BIST, since their regular test patterns simplify the hardware required to implement BIST.

The two-dimensional FUB array used in the CM-LCV is particularly appealing in this regard as it possesses an additional property: given careful test-vector selection, the output of the fault-free array is equivalent to another (related) test vector. This makes the CM-LCV a perfect candidate for a circular feedback approach for BIST wherein the circuit inputs are directly driven by the circuit outputs. Previous BIST architectures with circular feedback include the Circular Self-Test Path (CTSP)[2] and the Circular Cellular BIST (C²BIST)[12]. However, both of these methods require additional circuitry along the feedback path, which is not necessary for the CM-LCV due to the aforementioned property.

III. BUILT-IN SELF TEST DESIGN

This section describes the BIST architecture. Specifically, Section III-A defines and formally establishes the relevant properties of the CM-LCV. Section III-B then details the changes to the circuit architecture required for circular BIST.

A. Theory

The core assumption of this work is that the output of the logic array used in the CM-LCV is equivalent to another test for some high-quality set of test vectors. This property is proven through three theorems. First, however, it is necessary to establish the notation used in these theorems.

Figure 2a summarizes the relevant notation for a single FUB. A FUB is a combinational circuit that implements function $F$ with vertical and horizontal inputs, denoted as $h$ and $v$, respectively, and horizontal and vertical outputs,
denoted as $\hat{h}$ and $\hat{v}$, respectively. The input pattern of a FUB is denoted as $p = (h, v)$. Similarly, the output response of a FUB is represented as $r = (\hat{h}, \hat{v})$. Finally, a cycle $C$ of function $F$ is defined as a sequence of input patterns $\{p^{(i)}, p^{(1)}, \ldots, p^{(|C|-1)}\}$ such that $F(p^{(i)}) = r^{(i)} = p^{((i+1) \mod |C|)}$ and $p^{(i)} \neq p^{(j)} \forall p^{(i)}, p^{(j)} \in C$.

Figure 2b summarizes the relevant notation for an array of FUBs. An array of FUBs is denoted as $A$ and consists of $M$ rows and $N$ columns. The two tuple $(i, j)$ is used to refer to the FUB location in the array, and is also used as a subscript to refer to the specific FUB inputs/outputs at that location, i.e., $h_{(i,j)}, v_{(i,j)}, \hat{h}_{(i,j)}, \hat{v}_{(i,j)}$. An input array pattern includes all of the logic values applied to the primary inputs of the array and is denoted as $P = (H, V)$, where $H = \{h_{(1,1)}, h_{(2,1)}, \ldots, h_{(M,1)}\}$, $V = \{v_{(1,1)}, v_{(1,2)}, \ldots, v_{(1,N)}\}$. Similarly an array output response is defined as $R = (H, V)$ where $H = \{\hat{h}_{(1,N)}, \hat{h}_{(2,N)}, \ldots, \hat{h}_{(M,N)}\}$, $V = \{\hat{v}_{(1,1)}, \hat{v}_{(1,2)}, \ldots, \hat{v}_{(M,N)}\}$.

Given these definitions, Theorem 1 describes a method for constructing a test set for an array of FUBs from a cycle that exists within the FUB function $F$:

**Theorem 1.** Given an $M \times N$ array of FUBs implementing bijective function $F$ with cycle $C$ of length $k$, there exists an array test set $T$ of size $k$ constructed from $C$ such that all patterns in $C$ are applied to all FUBs in the array when the tests of $T$ are applied.

**Proof:** Suppose a cycle $C$ in $F$ is of length $k$, that is, $C = \{p^{(1)}, p^{(2)}, \ldots, p^{(k)}\}$, where $p^{(i)} = (h^{(i)}, v^{(i)})$. Let $T$ be composed of $k$ array input patterns, that is, $T = \{P_1, P_2, \ldots, P_k\}$, and let each $P_s \in T$ be constructed according to the following:

$$H_s = \{h^{(s)}_{(1,1)} \mod k, \ldots, h^{(s)}_{(M,1)} \mod k\}$$
$$V_s = \{v^{(s)}_{(1,1)} \mod k, \ldots, v^{(s)}_{(1,N)} \mod k\}$$

Now consider all of the FUBs along the diagonal of the array, that is, all $F_{(i,j)}$ where $i + j = \lambda$. (see Figure 3 for an illustration.)

- For $\lambda = 2$ only $F_{(1,1)}$ meets the constraint. Observe that for test $P_s$, $h_{(1,1)} = h^{(s)}$ (determined by $H_s$) and $v_{(1,1)} = v^{(s)}$ (determined by $V_s$); thus $r_{(1,1)} = F(p^{(s)}) = p^{(s+1) \mod k}$.
- For $\lambda = 3$ both $F_{(2,1)}$ and $F_{(1,2)}$ meet the constraint. Observe that for test $P_s$, $p_{(2,1)} = p^{(s+1) \mod k}$ (determined by $H_s$ and $\hat{v}_{(1,1)}$); thus $r_{(2,1)} = F(p^{(s+1) \mod k})$. Similarly, for test $P_s$, $p_{(1,2)} = p^{(s+1) \mod k}$ (determined by $V_s$ and $\hat{h}_{(1,1)}$); thus $r_{(1,2)} = F(p^{(s+2) \mod k})$.
- For $\lambda > 3$, $h_{(i,j)}$ will be determined by either $H_s$ or a FUB on diagonal $\lambda - 1$; in both cases $h_{(i,j)} = h^{(s+\lambda-2) \mod k}$. Similarly $v_{(i,j)}$ will be determined by either $V_s$ or a FUB on diagonal $\lambda - 1$; in both cases $v_{(i,j)} = v^{(s+\lambda-2) \mod k}$. Thus $p_{(i,j)} = p^{(s+\lambda-2) \mod k}$, and therefore $r_{(i,j)} = F(p^{(s+\lambda-2) \mod k}) = p^{(s+\lambda-1) \mod k}$.

Thus for any test $P_s$, all FUBs along the diagonal defined by $\lambda$ have input pattern $p^{(s+\lambda-2) \mod k}$ applied. Given that $s$ ranges from 0 to $(k - 1)$ over the $k$ tests in $T$, $p^{(s+\lambda-2) \mod k}$ will cover all $p \in C$ irrespective of $\lambda$; thus, all FUBs along all diagonals will experience all input patterns in $C$ over the application of the $k$ tests in $T$.

Theorem 1 is particularly useful given that error propagation is guaranteed if the FUBs are bijective, as is the case in the CM-LCV; thus, all input pattern faults [13] in a cycle $C$ can be effectively tested across the entire array using a fixed test set of size $|C|$. Theorem 2 builds on this result by showing that the output of the FUB array is equivalent to another test constructed from this same cycle if the array is square (i.e., $M = N$).

**Theorem 2.** Given an $N \times N$ square array of FUBs implementing bijective function $F$ with cycle $C$ of length $k$ and the test set $T$ constructed from $C$ according to Theorem 1, the array output $R_s \in T$ for all test $P_s \in T$.

**Proof:** Consider test $P_s \in T$ derived from cycle $C$. The proof for Theorem 1 demonstrated that for a fixed $\lambda = i + j$, $p_{(i,j)} = p^{(s+\lambda-2) \mod k}$. Thus by definition $\hat{p}_{(i,j)} = p^{(s+\lambda-1) \mod k}$. However, recall that $R_s = (H_s, \hat{V}_s)$ where in this case:

$$H_s = \{h_{(1,N)}, h_{(2,N)}, \ldots, h_{(N,N)}\}$$
$$\hat{V}_s = \{\hat{v}_{(N,1)}, \hat{v}_{(N,2)}, \ldots, \hat{v}_{(N,N)}\}$$

Substituting according to $\lambda$ these become:

$$H_s = \{h^{(s+N)} \mod k, h^{(s+N+1)} \mod k, \ldots, h^{(s+2N-1)} \mod k\}$$
$$\hat{V}_s = \{v^{(s+N)} \mod k, v^{(s+N+1)} \mod k, \ldots, v^{(s+2N-1)} \mod k\}$$

This $R_s$ is identical to $P_q \in T$ where $P_q$ is constructed starting at $p^{(s+N) \mod k}$; thus $R_s \in T$.

Thus a cycle-based test set constructed according to Theorem 1 can be applied to a square array of bijective FUBs.
by driving the array input with the array output. While this result is useful, its utility is limited by the length of the cycles present in the bijective function. To circumvent this limitation, Theorem 3 is introduced.

**Theorem 3.** An $M \times N$ array of FUBs implementing bijective function $F$ is equivalent to a single bijective function $G$ with inputs and outputs corresponding to the array inputs and outputs, respectively.

*Proof:* Suppose two array inputs $P_i \neq P_j$ both evaluate to the same array output $R_k$. The difference between $P_i$ and $P_j$ can be represented by a set of errors on the array inputs. Because every FUB in the array is bijective, these errors are guaranteed to propagate through the array to some set of array outputs. Thus $G(P_i) \neq G(P_j)$ for all $P_i \neq P_j$, that is, $G$ is one-to-one. Furthermore, because the array input space is the same size as the array output space (same number of array inputs/outputs), $G$ must be onto. Therefore $G$ is bijective. □

Thus, according to Theorem 3, any $m \times n$ sub-array within a FUB array can be represented as a single bijective function. Furthermore, if the dimensions of a FUB array are a scalar multiple of $m \times n$, the overall array can be considered a square array of FUBs implementing this sub-array bijective function. Thus a test set for this array can be derived using Theorem 1 and the cycles of the sub-array bijective function. Assuming some limited design freedom for the FUB array dimensions, it is expected that this process can be used to derive optimal test sets for the CM-LCV array by examining the cycles of various sub-array sizes. Exploiting Theorem 3 is further examined in Section IV.

**B. Circuit Design**

Implementation of circular BIST requires very little modification to the original CM-LCV design. The main components consist of the two dimensional array of FUBs and a single scan chain around its periphery. Circular feedback is added to connect the array outputs to the normal-mode inputs of the scan chains that drive the array input. Thus, for normal-mode operation (i.e., when the scan enable signal is not asserted), the scan chains feeding the array are updated with the array output values. Execution of the circular BIST test cycle is achieved through three steps: first, a seed vector is loaded into the scan chain. Second, a series of normal-mode clocks are applied to run the test cycle. Finally, the resulting values are unloaded from the scan chain and compared to the expected, fault-free signature.

Two variants of this circular BIST architecture based on the length of the scan chain are explored: Figure 4a uses the minimum scan chain length required, while Figure 4b extends the scan chain to capture the outputs in a second set of flip-flops. While the short-chain variant requires less hardware overhead and has a shorter scan chain length, the long-chain variant has several advantages:

- **Diagnosability** - the additional flip-flops in the long-chain implementation allow the feedback connections to be tested independent of the logic array. This results in enhanced resolution because the short-chain implementation cannot distinguish between a failure in the FUBs located on the output edge of the array and a failure due to the feedback connections.
- **Test independence** - the long-chain implementation allows for two independent test vectors to be applied during BIST. Specifically, one vector can be applied to the logic array by the input scan chain while a second vector is simultaneously transferred from the output scan chain to the input scan chain via the feedback connections.

**IV. Experiment**

This work continues to use the “gamma” 4-input VH-bijective function identified in the CM-LCV work [1]. The truth table for this function is given in Table I. A test set and a fault model are needed to properly evaluate the effectiveness.
of the proposed circular BIST architecture. First, a high-quality test set is derived according to Theorem 1. Section III-A stated that an array can be considered a square array of FUBs implementing the sub-array bijective function (assuming the array dimensions are a multiple of the sub-array dimensions). Thus, given freedom over the size of the FUB array, the cycles of various sub-array sizes are all of interest as potential sources for a test set for the overall FUB array. Table II lists the cycle lengths for the sub-array bijective functions for various sub-array sizes; each entry under the column labeled “Cycle lengths” denotes a specific cycle length present in the sub-array function as well as the number of such cycles shown as an integer in parentheses. Note that the number of BIST cycles required for a single execution of BIST is at most the maximum cycle length for the corresponding sub-array.

The data shown in Table II support the previous assertion that test cycles of varying length can be found by considering various sub-array sizes. Of note are the single cycle of length 889 present in the $2 \times 3$ sub-array bijective function, and the three cycles of length 5,461 present in the $4 \times 3$ sub-array bijective function. Recalling Theorem 1, the existence of these cycles indicates that it is possible to create, for example, a design with a circular BIST test cycle of length 5,461 can be constructed if the array dimensions are a multiple of $4 \times 3$. The remainder of this section focuses on test sets derived from the 889 test cycle derived from the $2 \times 3$ sub-array bijective function for a $6 \times 9$ array.

Quantifying fault coverage for the proposed circular BIST architecture using fault simulation is computationally expensive since fault effects (i.e., errors) may be masked when errors are repeatedly fed back through a defective array. To mitigate this issue, fault coverage can instead be equated to the combination of (a) fault activation, and (b) the probability of error masking.

First the fault activation achieved by a test set is examined. A single-pattern input pattern (IP) fault model [13] is assumed for each FUB in the array. Thus, in order to achieve 100% fault activation, the test set must apply all $2^d$ possible input patterns to each FUB in the array. Figure 5 shows how fault activation evolves as more tests are applied from the 889 tests derived from the 889-length cycle. Figure 5 indicates that the test set achieves 100% fault activation for all locations in the array by the 89th test.

However, the fault activation results of Figure 5 is not indicative of all the capabilities of this test set. Activation for each $2 \times 2$ sub-array within the overall array translates to all $2^8$ possible input patterns applied to the 8-input sub-array. From another perspective, fault activation for each $2 \times 2$ sub-array is equivalent to a 16-detect (or, rather, 16-activate) test set for each individual FUB. Figure 6 shows how fault activation evolves for all $2 \times 2$ sub-arrays for the same test set and $6 \times 9$ array. The test set achieves 100% IP fault activation for all $2 \times 2$ sub-arrays by the 835th test.

Activation of all faults is necessary but not sufficient for 100% fault detection; one way to check sufficiency is to consider the probability of error masking within the circular BIST architecture. Recalling the VH-bijectivity property, it is guaranteed that an error will propagate to both the vertical and horizontal outputs of the array for some test that activates the fault. These erroneous signals are guaranteed to be propagated by the non-faulty FUBs in the array during subsequent BIST cycles; the only way for an activated fault to escape detection is for these erroneous signals to converge on the faulty FUB to be masked before the final signature is observed. The probability of this form of masking is expected to be extremely

![Fig. 5. Evolution of IP fault activation over test index for a 6×9 array using a test set derived from the 2×3 sub-array cycle of length 889. Each trace represents the IP fault activation for each FUB in the 6×9 array.](image1)

![Fig. 6. Evolution of IP fault activation for 2×2 sub-arrays over test index for a 6×9 array using a test set derived from the 2×3 sub-array cycle of length 889. Each trace represents the IP fault activation for each unique 2×2 sub-array in the 6×9 array.](image2)
TABLE III
SUMMARY OF SIMULATION RESULTS FOR BOTH A SCAN TEST SET AND CIRCULAR BIST APPLIED TO A 6×9 FUB ARRAY.

<table>
<thead>
<tr>
<th>Metric</th>
<th>Scan test</th>
<th>Circular BIST</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clock cycles</td>
<td>7714</td>
<td>1013</td>
</tr>
<tr>
<td>SSL fault coverage</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td>IP fault coverage</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td>Simulation time (min.)</td>
<td>12.9</td>
<td>261.3</td>
</tr>
</tbody>
</table>

low, and moreover, it is expected to decrease exponentially as the size of the FUB array increases\(^1\). Thus the true fault coverage is expected to be equivalent to the level of fault activation achieved by the applied test set in the case of a single faulty FUB.

Empirical justification of the low likelihood for masking is investigated by simulating an implementation of a 6×9 FUB array using a commercial tool for both single-stuck line (SSL) and IP faults. The results are summarized in Table III for both the circular BIST scheme based on the 2×3 sub-array and a scan test set that achieves a similar level of fault detection. Note that 100% SSL and IP fault coverage is achieved by both the circular BIST and scan test for the given design.

Furthermore, the circular BIST achieves this perfect fault coverage with reduced test time compared to the scan test. The number of test cycles for the scan test is determined by (a) the length of the scan chain, and (b) the number of test vectors that need to be applied to achieve the desired fault coverage on all FUB blocks in the array. Note that in this discussion we are assuming the “long-chain” variant (Figure 4b), in which case only half of the scan chain needs to be observed for each scan test vector. Thus the number of test cycles for scan test can be expressed as: \( TC_{SCAN} = \frac{(vecs + 1) \times sc\_length}{2} + vecs + k \), where \( k \) is a small constant number of test cycles added to ensure the scan chain is functioning properly, and \( vecs \) is determined by the FUB size (in the example design used here, 256 vectors are required to apply all input patterns to the 2×2 sub-arrays of the 4-input FUBs). However, circular BIST only requires two full scan-chain loads/unloads and a constant number of BIST cycles. Thus the number of test cycles required can be expressed as: \( TC_{BIST} = 2 \times sc\_length + BIST\_cycles + k \), where \( k \) is a small constant number of test cycles added to ensure the scan chain is functioning properly, and \( BIST\_cycles \) is the number of BIST cycles. Thus, for the 6×9 FUB array discussed in this section, the circular BIST achieves an 86.9% reduction in the number of test clock cycles. This reduction will asymptotically approach \( 1 - \frac{4}{256} = 98.4\% \) for the 4-input FUBs as the array size increases (due to the dominance of the scan chain length factor in the two expressions). Note that this reduction is only in the number of test cycles, and fails to account for the fact that the circuit can be clocked faster for BIST as compared to scan; thus the actual test-time reduction is likely greater than the 86.9% presented for the 6×9 FUB array.

V. CONCLUSION

This work presented a circular BIST scheme for the CM-LCV design. The CM-LCV is an appealing target for circular BIST for reasons that include: (i) its high degree of regularity and (ii) unique test-set properties. The presented circular BIST implementation achieves 100% SSL and IP fault coverage for the CM-LCV design with an 86.9% reduction in test time. Furthermore, implementation of the circular BIST incurs minimal hardware overhead cost, consisting exclusively of additional routing to/from the scan cells. The significant increase in simulation time (due to the fact that faults cannot be dropped until the BIST execution is complete) is a challenge, but may be addressed through fault sampling. Future work includes developing additional approaches to improving simulation time, proving fault detection properties under the circular BIST scheme, and measuring and assuring similarity between the test chip and customer products.

REFERENCES


---

\(^1\)In a larger FUB array, a single defective FUB has less impact on the overall functionality of the array (which, being composed of bijective FUBs, will continue to propagate errors indefinitely). Furthermore, the state space for the FUB array increases exponentially with the number of array inputs/outputs; if we assume that an injected fault causes the FUB array to output random states, the probability of masking decreases exponentially with the size of the state space. However, at this point the only means of verifying the probability of masking is simulation of the design.