Synthesis of Fault-Tolerant Reconfigurable Scan Networks

Sebastian Brandhofer, Michael A. Kochte, Hans-Joachim Wunderlich
email: {brandhofer, kochte}@iti.uni-stuttgart.de, wu@informatik.uni-stuttgart.de

ITI, University of Stuttgart, Pfaffenwaldring 47, D-70569, Stuttgart, Germany

***Abstract*—On-chip instrumentation is mandatory for efficient bring-up, test and diagnosis, post-silicon validation, as well as in-field calibration, maintenance, and fault tolerance. Reconfigurable scan networks (RSNs) provide a scalable and efficient scan-based access mechanism to such instruments. The correct operation of this access mechanism is crucial for all manufacturing, bring-up and debug tasks as well as for in-field operation, but it can be affected by faults and design errors.**

This work develops for the first time fault-tolerant RSNs such that the resulting scan network still provides access to as many instruments as possible in presence of a fault. The work contributes a model and an algorithm to compute scan paths in faulty RSNs, a metric to quantify its fault tolerance and a synthesis algorithm that is based on graph connectivity and selective hardening of control logic in the scan network. Experimental results demonstrate that fault-tolerant RSNs can be synthesized with only moderate hardware overhead.

***Index Terms*—Fault Tolerance, Reconfigurable Scan Network, IEEE Std 1687, iJTAG, On-Chip Infrastructure, DFT**

I. Introduction

On-chip instruments, such as health monitors, test wrappers, and trace & debug structures, are essential for complex runtime objectives [1, 2], bring-up and post-silicon validation including test, diagnosis, and debug [3].

To access such instruments, reconfigurable scan networks (RSNs) have been proposed and standardized in IEEE Std 1149-2013 or IEEE Std 1687-2014. In RSNs, the scan-based dataflow between the primary input, the on-chip instruments, and the primary output is *reconfigurable*. This enables cost-efficient and scalable communication for a wide range of on-chip instruments [4–6].

Scan networks themselves must also be robust since they connect critical on-chip instruments and are affected by a large percentage of defects in a chip [7, 8]. A defect in an RSN can disturb the communication to on-chip instruments by corrupting the scan data or the scan-based dataflow. In the first case, a short, an open or a hold time violation in RSN registers can set a part of the scan data to an erroneous value [9]. In the second case, a set of on-chip instruments is permanently disconnected from the remainder of the chip. This behavior is typically modeled by stuck-at and transition faults [10].

Empirical data shows that 50% of scan network failures lead to chip failure [11]. Even if a scan network failure does not directly cause a malfunction of the mission logic, it may severely impact the post-silicon validation of the complete system. A fault in the RSN may prevent programming required values into on-chip instruments or reading out the status of monitors. In consequence, the testability and diagnosability of the complete chip is significantly impeded. Debugging the chip or improving the yield, especially during bring-up, requires an efficient and effective diagnosis that guides e.g. physical failure analysis for root cause analysis [12].

Effective post-silicon validation therefore requires fault-tolerant RSNs that keep the scan-based dataflow intact, avoid complete chip failure and enable to extract a maximum of information from the on-chip instruments in presence of faults. Since scan networks already require up to 30% of the area [8], or 50% of the transistors in chips [13], it is equally important to minimize the incurred hardware-overhead of fault-tolerant RSNs.

If an RSN tolerates stuck-at faults, it can tolerate a wide range of defects including those that lead to transition faults [14]. We therefore focus on synthesizing RSNs that are tolerant against stuck-at faults.

The state of the art describes functional and structure-oriented testing of RSNs [15–17], as well as diagnostic pattern generation specifically for RSNs [18]. These algorithms are crucial to detect and pinpoint to defects in an RSN.

However, these prior works neither address how to access on-chip instruments in presence of a fault nor how to synthesize a fault-tolerant RSN. The work at hand solves both aspects by developing:

- A formal model for computing access patterns to circumvent stuck-at faults in an RSN.
- A metric to quantify the tolerance of an RSN with respect to stuck-at faults.
- A synthesis algorithm that adds a minimum amount of redundancy to a given RSN such that the resulting RSN is fault-tolerant.

The proposed synthesis method consists of the steps in fig. 1:

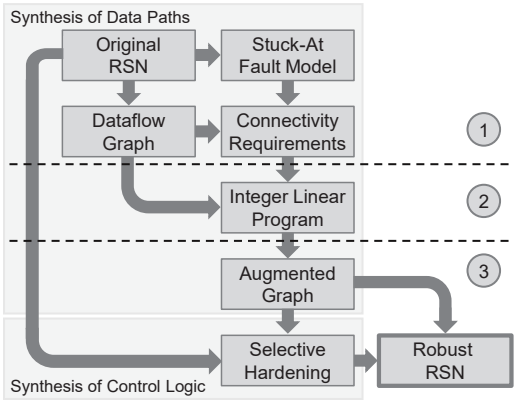

Fig. 1. Overview of the developed synthesis method

1) A dataflow graph is constructed from the given RSN

structure. Connectivity requirements for the RSN components are derived based on the stuck-at fault model and the given graph.

2) An integer linear program (ILP) is constructed from the graph and the connectivity requirements. The solution to this ILP is an augmented graph that incorporates redundant edges.
3) The augmented graph is processed together with the selective hardening of control signals to synthesize a fault-tolerant RSN.

The remainder of this paper is organized as follows. Section II describes reconfigurable scan networks and their formal modeling that is used as a basis for the quantification of fault tolerance. Section III describes the RSN model extensions for the fault tolerance metric and the developed synthesis method for fault tolerance. Lastly, the fault tolerance improvement of the developed method and incurred costs are evaluated based on benchmark circuits in section IV.

II. Reconfigurable Scan Networks (RSNs)

This section explains the structure and behavior of RSNs. The used terminology is based on IEEE Std 1687 and previous works [16, 19–22] .

A. Structure and Operation

An RSN consists of scan segments, scan multiplexers, and control logic, as shown in fig. 2. The RSN provides scan-based access to those scan segments that are part of a reconfigurable scan path. Access to an RSN is typically provided through a JTAG test access port (TAP).

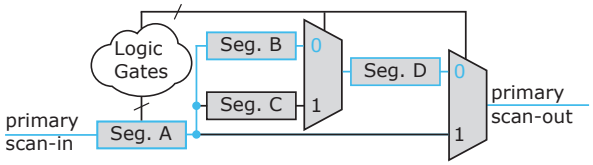

Fig. 2. A reconfigurable scan network with scan segments A, B, C, D. Segments A, B, D are on the active path (light blue) in the initial state.

A *scan segment* contains registers that are used to communicate with instruments and to drive control logic including the address inputs of scan multiplexers.

Moreover, a segment has a *select* control signal that determines whether it participates in RSN accesses (*select*=1) or remains passive (*select*=0). The path from the primary scan input of the RSN through selected segments and scan multiplexers to the primary scan output is called the *active scan path*. Scan segments that are not part of the active scan path are *deselected*.

The interface of a scan segment, as shown in fig. 3, consists of control signals, a scan input (scan-in), scan output (scan-out), data input and data output ports. A scan segment has one shift register, placed between the scan input port and scan output port. It must have a *shadow* register if the segment provides write access to an attached instrument or drives control signals. Shift and shadow registers can contain one or multiple flip-flops. The state of all shadow registers and primary input ports of an RSN is called *scan configuration*. A scan configuration is *valid* if it contains only one active scan path. The set of scan segments, scan multiplexers and interconnects is called *scan elements*. Global control signals in

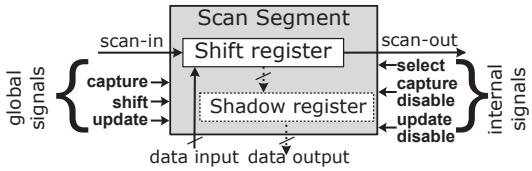

Fig. 3. Scan segment with a shift register and an optional shadow register. Clock and reset signals omitted for clarity.

the RSN trigger three operations on shift registers of selected segments: The *capture* operation (C) captures a scan segment's data input into its shift register. The *shift* operation (S) stores a scan segment's scan input value in its shift register. The *update* operation (U) latches the shift register state into the shadow register. A read/write access to the selected segments in an RSN is implemented by a CSU operation, which comprises a capture cycle, multiple shift cycles (typically corresponding to the length of the active scan path), and a final update cycle.

The capture (and update) operation of individual scan segments on the active scan path can be disabled by asserting their capture (update) disable signal.

B. Formal Modeling for Fault-Free Behavior

The formal RSN model proposed in [23] is used as basis to check the accessibility of scan segments in presence of faults. This model computes a time-optimal series of CSU operations that set up the required scan configuration and shift data through a scan segment for every scan segment access. The latency of any access is then given by the sum of the number of cycles of each CSU operation in the computed series. The model is only able to handle fault-free RSNs. An extension for stuck-at faults is described in section III-A. The formal RSN model is constructed from a structural description of the RSN and defined as the set $\mathcal{M} := \{S, H, I, V, C, c_0, \texttt{Select}, \texttt{Updis}, \texttt{Capdis}, \texttt{Active}\}$, where:

- S is the set of all scan segments in the RSN and I is the set of primary (data/control) inputs of the RSN.
- H is the set of shadow registers. Each shadow register is mapped to its segment via the function $\mathcal{S} : H \rightarrow S$.
- V is the set of binary variables modeling the state of primary inputs and shadow registers $D := H \cup I$.
- $C := \{0,1\}^{|D|}$ is the set of scan configurations. The function $c : D \rightarrow \{0,1\}$ returns the assignment to element d in a configuration c. $c_0 \subset C$ is the set of initial configurations (reset states).
- $\texttt{Select}, \texttt{Updis}, \texttt{Capdis}$ are predicates that define the state of the select, capture and update disable port of a segment s in configuration c.
- $\texttt{Active} : C \times S \rightarrow \{0,1\}$ defines the active scan path. It only evaluates to *true* if the corresponding scan segment is selected in a valid configuration, i.e., there exists exactly one active scan path.

The transition relation $T \subseteq C \times C$ includes all pairs (c_i, c_j) of scan configurations $c_i, c_j \in C$ where c_j can be reached from c_i within a single CSU operation. The characteristic function of T is defined as:

$$T(c_i, c_j) := \bigwedge_{h \in H} [(\neg \texttt{Active}(c_i, \mathcal{S}(h)) \vee \texttt{Updis}(c_i, \mathcal{S}(h))) \Rightarrow (c_i(h) = c_j(h))]. \quad (1)$$

T defines the conditions under which a scan configuration may change in the RSN.

Bounded model checking (BMC) [24] is used to decide the accessibility of a scan segment by unrolling the transition relation $n+1$ times and checking if certain access conditions are satisfied after the n-th CSU operation.

III. Synthesis of Fault-Tolerant RSNs

The fault-tolerant synthesis method consists of the steps in fig. 1. The method expects the structural description of an RSN as input and models its dataflow as graph. The graph is augmented by additional edges. Integer linear programming (ILP) is used to compute a minimal set of augmenting edges that satisfy the connectivity requirements of fault-tolerant RSNs. Finally, the control logic in the resulting fault tolerant RSN is constructed and hardened. All access scenarios in the given RSN are preserved: scan paths that were configurable in the original RSN are still configurable in the fault-tolerant RSN.

A. RSN Model Extensions for the Fault Tolerance Metric

A metric is developed that quantifies the fault tolerance of an RSN as the fraction of accessible scan segments in presence of a fault. For this metric, the stuck-at 0/1 fault model is used and faults at all scan segment, register and multiplexer ports and at all logic gates that fan out into multiple ports are considered. The RSN model is extended to model the effect of stuck-at faults on the accessibility of scan segments by:

- Stuck-at constraints: registers and signals are set to the stuck-at value if they are the fault site.
- Adapted transition relation: if the fault site is part of the active scan path, the stuck-at value of the fault propagates to subsequent updatable registers on the active scan path.
- Access conditions: the location and propagation of the fault site must be considered when reasoning about the accessibility of a scan segment.

These conditions are modeled formally in constraints and predicates and added to the RSN model. Then, access is checked (as shown in section II-B) for every scan segment and every considered fault. The results of these checks are aggregated to obtain the average and worst case accessibility of scan segments in presence of stuck-at faults.

B. RSN Dataflow Graph

Essentially, RSNs are unidirectional scan-based communication networks and can be modeled as directed graphs. In this graph $G = (V, E)$, vertices V represent scan segments and primary input/output ports. The primary scan-in port vertex is the unique root of the graph and the primary scan-out port vertex is the unique sink of the graph. Graph edges E model the interconnects between these components. Control logic is not included in the graph. Only the dataflow in an RSN is modeled as graph.

According to IEEE Std 1687, structural cycles may exist in an RSN if and only if they cannot be sensitized, i.e. an active scan path may not form a cycle. Therefore, such cycles can be broken and the dataflow in the RSN can be modeled as a directed acyclic graph.

C. Connectivity Requirements of Fault-Tolerant RSNs

Accessing a scan segment s in an RSN requires the existence of a path p in its dataflow graph that connects the primary scan-in through segment s to the primary scan-out port. Furthermore, the path p must be configurable, i.e. CSU operations must exist that set p as the active scan path.

A fault-tolerant RSN requires both of these access conditions to hold in presence of any stuck-at fault for as many scan segments as possible. The first access condition requires the connectivity of the dataflow graph to tolerate stuck-at faults. For the second access condition, control signals must be hardened. This is addressed during the final synthesis described in section III-E.

When a stuck-at fault affects a scan element, the dataflow along this scan element is corrupted. Hence, the dataflow must be realized along a secondary path that does not use the vertex or edge corresponding to the faulty scan element.

The dataflow graph of an RSN contains such a secondary path for every scan segment s and every stuck-at fault, if there are at least two *vertex-independent* paths from the primary scan-in vertex to s and from s to the primary scan-out vertex. Two paths are vertex-independent iff they have no vertex in common except possibly their start or end vertex.

If there are no such paths for one scan segment s, all paths connecting s have at least one vertex or edge in common. The scan element corresponding to this vertex or edge is a single point of failure, i.e. s becomes inaccessible when the scan element is faulty.

The existence of such vertex-independent paths is ensured by augmenting the connectivity of the graph as described in the following section.

D. Connectivity Augmentation

To satisfy the vertex independence requirements, the connectivity of the dataflow graph $G = (V, E)$ is augmented by a minimal edge set E_A that is determined by ILP-based optimization. The augmented graph $G_A = (V, E_A)$ is then processed in the final synthesis step. An ILP is defined by a set of variables, an optimization objective and a set of constraints that acts on the variables. An optimal solution to an ILP is an assignment to the variables that is optimal according to the optimization objective and that satisfies all constraints. The variables of the proposed ILP represent the set of potential edges E_P of G, defined as:

$$E_P := \{e = (i, j) \mid \forall i, \forall j \in V : level(j) \geq level(i)\}$$

where $level(i)$ returns the topological level of vertex i. The set of potential edges defines every potential way in which the dataflow graph of the original RSN can be extended to yield an augmented graph that satisfies the vertex independence requirement.

The optimization objective is defined by a cost function, here given as matrix c^T, which is used to determine the minimal set of augmenting edges. An arbitrary linear cost function can be used, but the cost of an edge should reflect the cost of integrating the edge into an RSN. In this work, the cost of an edge is defined as a function of the level difference of the vertices connected by the edge. The edge cost is zero if it is part of the dataflow graph G. This cost function, when minimized, reduces the number of additional long signal lines that are integrated into the RSN in the final synthesis step. The edge set $E_A \supseteq E$ always contains all zero cost edges, since the original RSN is used as a basis.

The constraints of the proposed ILP are defined such that the augmented graph G_A:

1) satisfies the vertex independence requirement of fault-tolerant RSNs.
2) does not contain any cycles.

To ensure vertex independence in graph G_A, it is sufficient to require that each vertex of the graph has two incoming edges from and two outgoing edges to distinct vertices [25, 26]. The acyclicity constraints are modeled after the subtour elimination problem known from the traveling salesman problem.

The integer decision variables x of the ILP are defined as: $x \in \mathbb{R}^{|E_P|}$, where $x_e = 1$ iff edge $e = (i, j)$ is part of the augmenting edge set $E_A \subseteq E_P$. The vertex degree, acyclicity and the binary constraints on the variables x are defined as:

$$\text{minimize} \quad c^T x$$

$$\text{subject to}$$

$$\forall t \in V: \quad \sum_{e=(\cdot,t)\in E_P} x_e \geq 2, \tag{2}$$

$$\forall t \in V: \quad \sum_{e=(t,\cdot)\in E_P} x_e \geq 2, \tag{3}$$

$$\forall W \in V_{lv}, \forall W' \subseteq W: \quad \sum_{e=(i,j):i,j\in W'} x_e \leq |W'| - 1, \tag{4}$$

$$\forall e \in E_P: \quad x_e \in \{0, 1\}. \tag{5}$$

1) Degree Constraints: The first two constraints enforce the indegree respectively outdegree of every vertex in G_A to be at least two by requiring the set of incoming respectively outgoing edges of every vertex t to have at least two elements each. Thus, in the augmenting edge set E_A, every vertex t has at least two incoming and two outgoing edges. These constraints are only enforced for a vertex if it can satisfy the constraint in principle: for instance, the indegree of the root vertex is not enforced to be at least two.

2) Acyclicity Constraints: Let W' be a set of vertices, then $|W'|$ is the minimum number of edges that can constitute a cycle containing every vertex in W'. Therefore, enforcing the number of incident edges in every subset $W' \subseteq W$ to be less than $|W'|$ ensures that there are no cycles in W. Let V_{lv} be the sets of vertices with the same level: $V_{lv} = \{W_0, W_1, ..., W_n \mid W_i$ contains vertices v with level$(v) = i\}$.

The solution to this ILP is an assignment to the integer decision variables x such that the degree and acyclicity constraints are satisfied at minimal cost. A minimal cost solution for the RSN from fig. 2 is shown in fig. 4.

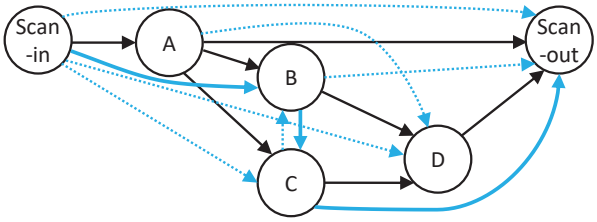

Fig. 4. Potential edges (dotted and solid) and vertices of the RSN in fig. 2. Edges of the dataflow graph are colored black. The edges of the minimal augmenting edge set E_A computed by the ILP have solid lines (black and light blue).

E. Final Synthesis

This section explains the steps that synthesize the fault-tolerant RSN based on the computed augmenting edge set.

1) Integration of the Augmenting Edge Set: For every edge $e = (i, j)$ in the augmenting edge set E_A that does not represent an interconnect in the original RSN, a multiplexer m is added to the RSN. The component represented by vertex i and the original predecessor of j are the inputs of m. The output of m is connected to the component represented by vertex j. The address signal of m is connected in step 3).

2) Hardening of Select Signals: If a faulty select signal fans out, it can affect multiple scan segments. To limit the impact of such signals, select signals are synthesized such that for any given segment there are at least two independent ways of asserting its select signal.

The last scan element (primary scan-out) is always selected when the RSN is enabled - its select signal is directly derived by the primary select port of the RSN. The assertion of the select signal of a scan element u is recursively derived by the following rules that depend on the type and number of the successors of u:

- Let u fan out to multiple scan elements, then any of the direct successors of u must be selected.
- Let the direct successor of u be a multiplexer, then this multiplexer must be selected and must be configured to forward u to its output.
- Let u have one direct successor, then this successor must be selected.

Fig. 5 shows the vicinity of scan segment B after completing the first two steps of the final synthesis. Let $\texttt{Select}(s)$ be the state of the select port of a scan segment s. The state of the select port of scan segment B is defined as:

$$\texttt{Select}(B) := (\texttt{Select}(D) \wedge \neg a) \vee (\texttt{Select}(C) \wedge \neg b).$$

where $\texttt{Select}(D)$ and $\texttt{Select}(C)$ are defined as above.

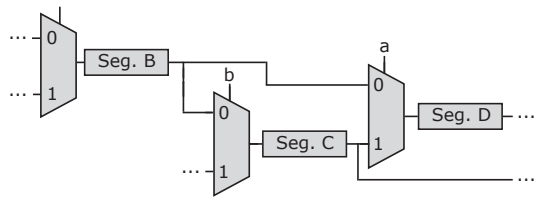

Fig. 5. Vicinity of scan segment B from fig. 2 after the first two steps of the final synthesis. The multiplexer address signals, such as a and b, are yet to be connected.

If the select signal in one fan-out stem of a scan segment is affected by a stuck-at 0 fault, the scan elements of the other fan-out stem can be used to set the select signal of the segment. If the select signal is affected by a stuck-at 1 fault, access is still possible through the resulting active scan path. The select signals of the original RSN are not used in the fault-tolerant RSN.

3) Multiplexer Address Signals: Faulty multiplexer address signals can impact multiple scan segments, since they often fan out and drive multiple multiplexers (see e.g. fig. 2). In this work, this is prevented by hardening address signals using triple modular redundancy.

4) Additional Primary Scan Ports: The primary scan-in and scan-out ports are still single points of failure. Duplicating these ports introduces a structural redundancy that can tolerate single stuck-at faults at the primary scan ports.

After adding a secondary scan-in port, it is connected to each successor of the original primary scan-in port via multiplexers. Likewise, each predecessor of the original primary scan-out port is connected to an additional secondary scan-out port via multiplexers. The address signals of these multiplexers are hardened as explained above.

IV. Evaluation

This section shows the improvement in fault tolerance of the original and fault-tolerant RSN as well as the area impact of the method on the RSN. Since all scan paths of the original RSN are still configurable in the fault-tolerant RSN, the number of cycles to access a scan segment in an active scan path is not increased by the synthesis.

A. Experiment Setup

For the evaluation, *SIB*-based RSNs [27] are generated from the ITC'02 system-on-chip (SoC) benchmarks. From these RSNs, fault-tolerant RSNs are synthesized and evaluated.

In SIB-based RSNs, segment insertion bits (SIB), consisting of one 1-bit register and a scan multiplexer, are used for a configurable bypass of hierarchies of scan segments. Depending on the value of the SIB register, the scan multiplexer either connects a lower hierarchy of scan segments to a higher hierarchy of scan segments or bypasses it.

Table I shows the characteristics of the original RSNs, the accessibility in SIB-based and fault-tolerant RSNs, and the area increase of the RSNs incurred by the proposed synthesis. The characteristics of the original RSN contain the number of modules, levels, multiplexers, segments and bits. The column "modules" gives the number of hardware modules of the target SoC that are connected via the RSN. The "level" column specifies the hierarchical depth of the RSN. The number of multiplexers, scan segments and scan bits in the corresponding RSN are given in the "mux", "segments" and "bits" columns.

B. Fault Tolerance Improvement

The developed fault tolerance metric is used to quantify the fault tolerance of the original and fault-tolerant RSNs: for each single stuck-at 0/1 fault in the RSN, the metric quantifies the fraction of scan segments and bits that are accessible in presence of the fault. Faults in global control signals such as the reset signal are not considered.

Column "worst" in table I states the worst case, i.e. the lowest fraction of accessible scan segments or scan bits in presence of a stuck-at fault. The worst case accessibility of SIB-based RSNs is a complete disconnection of every scan segment and scan bit, denoted by a value of 0.00. The worst case accessibility of scan segments in fault-tolerant RSNs improves significantly: from 95% up to 99.9% of scan segments are still accessible even for the worst fault. This high percentage is equivalent to an accessibility of all but one scan segments in a faulty RSN. To achieve full fault tolerance, all actual scan cells in the scan segments and connections to instruments must also be fault tolerant, e.g. by triplication, which is beyond the scope of this paper.

The average fault tolerance is computed over all single stuck-at faults and reported in column "avg". On average, 67% to 93% of all scan segments remain accessible in presence of a stuck-at fault in the *original* SIB-based RSNs. In fault-tolerant RSNs, over 99% of bits and scan segments remain accessible in presence of faults. These results demonstrate that the developed method effectively limits the impact of stuck-at faults on the accessibility of scan segments and scan bits in an RSN. This means that tasks such as post-silicon debug and diagnosis have access to almost all of the on-chip instruments in a faulty RSN.

The ILP solver of the proposed synthesis finished in less than 8 minutes and required less than 6.5GB of RAM for the largest instance p93791.

C. Area Overhead

The area overhead of the fault-tolerant RSN is examined by comparing the area of the original and fault-tolerant RSN as reported by a commercial logic synthesis tool. In addition to the area (last column), the number of multiplexers (column "mux"), bits (column "bits") and interconnects (column "nets") were extracted from the logic synthesis report of every original and fault-tolerant RSN. They are displayed as ratios in table I where the parameter of the fault-tolerant RSN is divided by the corresponding parameter of the original RSN.

The synthesis method increases the number of scan bits by 1% to 38% and the number of interconnects by 1% to 54%. The number of multiplexers increases by 198% up to 281%. Since the vertex independence constraint in the ILP requires at least two incoming edges for every vertex, every scan segment in the fault-tolerant RSN has at least one additional multiplexer

TABLE I. Characteristics and Accessibility in presence of faults of evaluated RSNs for ITC'02 SoCs, Area Overhead of Fault-Tolerant RSNs

SoC	RSN Characteristics					Accessibility in SIB-RSNs				Accessibility in Fault-Tolerant RSNs				RSN Area Overhead			
						Bits		Segments		Bits		Segments					
	modules	levels	mux	segments	bits	worst	avg	worst	avg	worst	avg	worst	avg	mux	bits	nets	area
u226	10	2	49	89	1465	0.00	0.71	0.00	0.76	0.93	***0.994***	0.975	***0.994***	3.67	1.38	1.54	***1.56***
d281	9	2	58	108	3871	0.00	0.81	0.00	0.83	0.79	***0.995***	0.980	***0.995***	3.62	1.17	1.24	***1.25***
d695	11	2	167	324	8396	0.00	0.90	0.00	0.90	0.96	***0.998***	0.994	***0.998***	3.54	1.21	1.32	***1.32***
h953	9	2	54	100	5640	0.00	0.85	0.00	0.85	0.94	***0.995***	0.978	***0.995***	3.59	1.10	1.15	***1.16***
g1023	15	2	79	144	5385	0.00	0.86	0.00	0.86	0.93	***0.997***	0.985	***0.996***	3.53	1.16	1.23	***1.24***
x1331	7	4	31	56	4023	0.00	0.75	0.00	0.78	0.86	***0.991***	0.960	***0.991***	3.81	1.09	1.13	***1.14***
f2126	5	2	40	76	15829	0.00	0.78	0.00	0.78	0.94	***0.993***	0.972	***0.993***	3.60	1.03	1.04	***1.04***
q12710	5	2	25	46	26183	0.00	0.80	0.00	0.80	0.86	***0.988***	0.952	***0.988***	3.56	1.01	1.02	***1.02***
t512505	31	2	159	287	77005	0.00	0.85	0.00	0.87	0.98	***0.998***	0.992	***0.998***	3.58	1.02	1.03	***1.03***
a586710	8	3	39	71	41674	0.00	0.78	0.00	0.79	0.94	***0.993***	0.969	***0.993***	3.72	1.01	1.02	***1.02***
p22081	29	3	282	536	30110	0.00	0.92	0.00	0.93	0.99	***0.999***	0.996	***0.999***	3.54	1.10	1.15	***1.15***
p34392	20	3	122	225	23241	0.00	0.87	0.00	0.86	0.97	***0.998***	0.990	***0.998***	3.68	1.06	1.09	***1.09***
p93791	33	3	620	1208	98604	0.00	0.66	0.00	0.67	0.99	***0.999***	0.999	***0.999***	3.55	1.07	1.11	***1.10***

at its scan-in port. This amounts roughly to a factor of 3x for the RSNs in table I since each RSN contains about twice as many scan segments as multiplexers.

Although the number of multiplexers increases a lot, the *overall* increase of the RSN area ranges only from 2% to 56%. The weighted average of the area increase is only 8.2% (weighted by number of scan bits).

V. Conclusion

Fault tolerance in reconfigurable scan networks (RSNs) is crucial for post-silicon debug, early bring-up and in-field maintenance. In this work, a novel fault tolerance metric and synthesis method for fault-tolerant RSNs have been developed. The synthesis method is based on graph augmentation and selective hardening of control signals.

Results show that the fault tolerance of an RSN significantly improves by the proposed method. On average, at least 99% of the scan segments in an RSN are still accessible in presence of single stuck-at faults. The incurred area overhead w.r.t. the original RSN is on average only 8.2%. The fault-tolerant RSN preserves all scan paths that are configurable in the original RSN.

Acknowledgments

This work was supported by the German Research Foundation (DFG) under grant WU 245/17-2 (ACCESS).

References

[1] N. Dutt and A. Jantsch, "Guest Editorial: Special Issue on Self-Aware Systems on Chip," *IEEE Design & Test*, vol. 35, no. 5, pp. 5–6, 2018.

[2] M. A. Kochte and H.-J. Wunderlich, "Self-Test and Diagnosis for Self-Aware Systems," *IEEE Design & Test*, vol. 35, no. 5, pp. 7–18, 2017.

[3] S. Mitra, S. A. Seshia, and N. Nicolici, "Post-Silicon Validation Opportunities, Challenges and Recent Advances," in *Proc. Design Automation Conference (DAC)*, 2010, pp. 12–17.

[4] F. G. Zadegan, D. Nikolov, and E. Larsson, "In-Field System-Health Monitoring Based on IEEE 1687," in *Proc. IEEE Int'l System-on-Chip Conf. (SOCC)*, 2016, pp. 69–74.

[5] A. Ibrahim and H. G. Kerkhoff, "A Cost-Efficient Dependability Management Framework For Self-Aware System-On-Chips Based on IEEE 1687," in *Proc. IEEE Int'l Symp. On-Line Testing and Robust System Design (IOLTS)*, 2017, pp. 1–2.

[6] A. Tsertov, A. Jutman, K. Shibin, and S. Devadze, "IEEE 1687 Compliant Ecosystem for Embedded Instrumentation Access and In-Field Health Monitoring," in *Proc. IEEE AUTOTESTCON*, 2018, pp. 1–9.

[7] C. Schuermyer, B. Benware, G. Rhodes, D. Appello, V. Tancorre, and O. Riewer, "Device Selection for Failure Analysis of Chain Fails Using Diagnosis Driven Yield Analysis," in *Proc. Int'l Symp. for Testing and Failure Analysis (ISTFA)*, 2011, pp. 91–97.

[8] R. Guo and S. Venkataraman, "A Technique for Fault Diagnosis of Defects in Scan Chains," in *Proc. IEEE Int'l Test Conf. (ITC)*, 2001, pp. 268–277.

[9] Y. Wu, "Diagnosis of Scan Chain Failures," in *Proc. IEEE Int'l Symp. on Defect and Fault Tolerance in VLSI Systems*, 1998, pp. 217–222.

[10] Y. Huang, R. Guo, W.-T. Cheng, and J. C.-M. Li, "Survey of Scan Chain Diagnosis," *IEEE Design & Test of Computers*, vol. 25, no. 3, 2008.

[11] Jheng-Syun Yang and Shi-Yu Huang, "Quick Scan Chain Diagnosis Using Signal Profiling," in *Int'l. Conf. on Computer Design (ICCD)*, 2005, pp. 157–160.

[12] D. P. Vallett, "IC Failure Analysis: The Importance of Test and Diagnostics," *IEEE Design & Test of Computers*, vol. 14, no. 3, pp. 76–82, 1997.

[13] F. Yang, S. Chakravarty, N. Devta-Prasanna, S. M. Reddy, and I. Pomeranz, "On the Detectability of Scan Chain Internal Faults an Industrial Case Study," in *IEEE VLSI Test Symp. (VTS)*, 2008, pp. 79–84.

[14] E. J. McCluskey and Chao-Wen Tseng, "Stuck-Fault Tests vs. Actual Defects," in *Proc. IEEE Int'l Test Conf. (ITC)*, 2000, pp. 336–342.

[15] R. Cantoro, M. Montazeri, M. S. Reorda, F. G. Zadegan, and E. Larsson, "On The Testability Of IEEE 1687 Networks," in *Proc. IEEE Asian Test Symp. (ATS)*, 2015, pp. 211–216.

[16] M. A. Kochte, R. Baranowski, M. Schaal, and H.-J. Wunderlich, "Test Strategies for Reconfigurable Scan Networks," in *Proc. IEEE Asian Test Symp. (ATS)*, 2016, pp. 113–118.

[17] D. Ull, M. A. Kochte, and H.-J. Wunderlich, "Structure-Oriented Test of Reconfigurable Scan Networks," in *Proc. IEEE Asian Test Symp. (ATS)*, 2017, pp. 127–132.

[18] R. Cantoro, M. Montazeri, M. Sonza, F. G. Zadegan, and E. Larsson, "Automatic Generation of Stimuli for Fault Diagnosis in IEEE 1687 Networks," in *Proc. IEEE Intl. Symp. On-Line Testing and Robust System Design (IOLTS)*, 2016, pp. 167–172.

[19] R. Baranowski, M. A. Kochte, and H.-J. Wunderlich, "Modeling, Verification and Pattern Generation for Reconfigurable Scan Networks," in *Proc. IEEE Int'l Test Conf. (ITC)*, 2012, pp. 1–9.

[20] R. Baranowski, M. A. Kochte, and H.-J. Wunderlich, "Scan Pattern Retargeting and Merging with Reduced Access Time," in *Proc. IEEE European Test Symp. (ETS)*, 2013, pp. 39–45.

[21] R. Baranowski, M. A. Kochte, and H.-J. Wunderlich, "Reconfigurable Scan Networks: Modeling, Verification, and Optimal Pattern Generation," *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, vol. 20, no. 2, p. 30, 2015.

[22] M. A. Kochte and H.-J. Wunderlich, "Dependable On-Chip Infrastructure for Dependable MPSOCs," in *Proc. IEEE Latin-American Test Symp. (LATS)*, 2016, pp. 183–188.

[23] R. Baranowski, "Reconfigurable Scan Networks: Formal Verification, Access Optimization, and Protection," Ph.D. dissertation, University of Stuttgart, 2014.

[24] A. Biere, A. Cimatti, E. M. Clarke, M. Fujita, and Y. Zhu, "Symbolic Model Checking Using SAT Procedures Instead of BDDs," in *Proc. Design Automation Conference (DAC)*, 1999, pp. 317–320.

[25] S. Brandhofer, "Synthesis of Robust Reconfigurable Scan Networks," Master's thesis, ITI - University of Stuttgart, 2017.

[26] G. Dahl, "Directed Steiner Problems With Connectivity Constraints," *Discrete Applied Mathematics*, vol. 47, no. 2, pp. 109–128, 1993.

[27] F. G. Zadegan, U. Ingelsson, G. Carlsson, and E. Larsson, "Design Automation for IEEE P1687," in *Proc. Conf. Design, Automation & Test in Europe (DATE)*, 2011, pp. 1–6.