Test Strategies for Reconfigurable Scan Networks

Kochte, Michael A.; Baranowski, Rafal; Schaal, Marcel; Wunderlich, Hans-Joachim

Proceedings of the 25th IEEE Asian Test Symposium (ATS'16) Hiroshima, Japan, 21-24 November 2016

doi: http://dx.doi.org/10.1109/ATS.2016.35

Abstract: On-chip infrastructure is an essential part of today's complex designs and enables their cost-efficient manufacturing and operation. The diversity and high number of infrastructure elements demands flexible and low-latency access mechanisms, such as reconfigurable scan networks (RSNs). The correct operation of the infrastructure access itself is highly important for the test of the system logic, its diagnosis, debug and bring-up, as well as post-silicon validation. Ensuring correct operation requires the thorough testing of the RSN. Because of sequential and combinational dependencies in RSN accesses, test generation for general RSNs is computationally very difficult and requires dedicated test strategies. This paper explores different test strategies for general RSNs and discusses the achieved structural fault coverage. Experimental results show that the combination of functional test heuristics together with a dedicated RSN test pattern generation approach significantly outperforms the test quality of a standard ATPG tool.

Test Strategies for Reconfigurable Scan Networks

Michael A. Kochte, Rafal Baranowski, Marcel Schaal, Hans-Joachim Wunderlich
email: {kochte, baranowski}@iti.uni-stuttgart.de, wu@informatik.uni-stuttgart.de
ITI, University of Stuttgart, Pfaffenwaldring 47, D-70569, Stuttgart, Germany

***Abstract*—On-chip infrastructure is an essential part of today's complex designs and enables their cost-efficient manufacturing and operation. The diversity and high number of infrastructure elements demands flexible and low-latency access mechanisms, such as reconfigurable scan networks (RSNs). The correct operation of the infrastructure access itself is highly important for the test of the system logic, its diagnosis, debug and bring-up, as well as post-silicon validation. Ensuring correct operation requires the thorough testing of the RSN.**

Because of sequential and combinational dependencies in RSN accesses, test generation for general RSNs is computationally very difficult and requires dedicated test strategies. This paper explores different test strategies for general RSNs and discusses the achieved structural fault coverage. Experimental results show that the combination of functional test heuristics together with a dedicated RSN test pattern generation approach significantly outperforms the test quality of a standard ATPG tool.

***Keywords*-Test generation, reconfigurable scan network, design-for-test, on-chip infrastructure, IEEE Std 1687, iJTAG**

I. Introduction

Today's and even more so future integrated systems contain a large diversity of on-chip instrumentation for efficient manufacturing, test, debug, and operation. Examples of this on-chip infrastructure include conventional scan-design and design-for-test structures, design-for-debug, repair structures, or online sensors and monitors for calibration and maintenance during runtime [1, 2].

In the past, such infrastructure has been accessed by standardized or proprietary bus- or serial scan-based access mechanisms. Standardized scan-based access networks are a low-cost, general purpose solution that has been widely used for accessing the on-chip instrumentation [1, 3, 4]. JTAG (Joint Test Action Group, IEEE Std 1149.1) has been a popular choice for scan-based access. For the scan-based access, the status or control registers of instruments are made scannable via *scan-in* and *scan-out* terminals. Multiple scannable registers are then connected to the JTAG circuitry either as ”Data Registers” or in series as a single register.

The variety and high number of instruments in complex designs require more flexible scan network architectures for low-cost and low-latency access. *Reconfigurable Scan Networks* (RSNs) are recently proposed scan architectures, in which the path through which data is shifted can be flexibly changed and minimized in length [4, 5]. In RSNs, the address signals of multiplexers on the scan path and other control signals can be external or generated internally in the network.

RSNs have been recently standardized by IEEE in two standards: IEEE Std 1149.1-2013 (JTAG-2013) allows RSNs with excludable and selectable shift registers for the efficient access to systems with power-gated modules [6]. IEEE Std 1687-2014 (*Standard for Access and Control of Instrumentation Embedded within a Semiconductor Device*) [7], also known as Internal JTAG (iJTAG), targets very flexible and generic scan-based integration and access to *arbitrary* on-chip instrumentation.

After manufacturing of a circuit, its infrastructure and in particular the scan-based access mechanism has to be tested first to ensure that test, diagnosis, and bring-up features are accessible and functional. Since scan-based infrastructure access mechanisms may already amount to up to 30% of the chip area or 50% of the transistors [8, 9], they may cause test failures and constitute a significant fraction of the overall yield loss, up to 50% as reported in [10, 11]. The thorough test of the infrastructure access is also relevant for safety critical systems, as monitoring, repair, or fault management features need to be accessed reliably during the whole system lifetime [12].

For the test of conventional, not reconfigurable scan chains, special bit sequences or flush tests, e.g., ”00110011”, can be applied to check the integrity of the scan cells and their interconnection [13, 11]. Such a basic test can already detect certain stuck-at and transition-delay faults. However, the effects of defects in scan cells and related control signals can be quite different from those of stuck-at faults, making already the test for non-reconfigurable scan chains highly challenging. Thus, dedicated test methods have been developed for more complex defects [14, 15, 16, 17, 18]. Correspondingly, the diagnosis of scan chains demands tailored approaches for localization of permanent and intermittent faulty behavior [19]. These methods, however, cannot be readily be applied to RSNs since their much more complex control logic and shadow elements also need to be exercised.

A fault in an RSN may affect the sequential elements and their shadow registers, scan multiplexers, control logic, or the interconnect. The fault effects may be observable only for a certain RSN state, i.e., for a certain active scan path. For such faults, the generation of test patterns in a gate-level RSN model may require a justification over a very high number of cycles, which is beyond the capabilities of existing tools for sequential test pattern generation. Consequently, even for the simplified stuck-at fault model assumption, the test of RSNs is an open problem.

This paper targets the test of structural faults in general RSNs and investigates the test quality of different test strategies. The next section briefly introduces RSNs addressed in this work. Section III discusses the related work. Section IV presents four different test strategies. The implementation is outlined in Section V. Experimental results are discussed in Section VI, followed by the conclusion.

II. Reconfigurable Scan Networks

This section summarizes the structure and terminology of RSNs relevant for this work, following the description in [2]: Reconfigurable scan networks (RSNs) are usually accessed through a JTAG Test Access Port (TAP). Compared to a static JTAG data register, an RSN behaves like a data register of *variable length*. The logic state of the RSN determines which scan registers in the RSN are currently accessible and this state can be changed by writing to the accessible registers.

The components of an RSN include scan multiplexers, scan segments or registers, and combinational logic to generate internal control signals. Scan registers may contain a shadow register for bidirectional communication with attached instruments or to generate control signals in the RSN, which must be stable during shifting. An example of an RSN is shown in Figure 1. The one-bit scan registers S1 and S3 control the access to two multi-bit scan registers S2 and S4, respectively. For instance, the scan-in data is shifted through register S2 only if a previous access set S1 to 1. Control signals can also depend on external inputs and combinational logic, as shown for the second multiplexer. The path through which the scan-in data is shifted is called the *active scan path*.

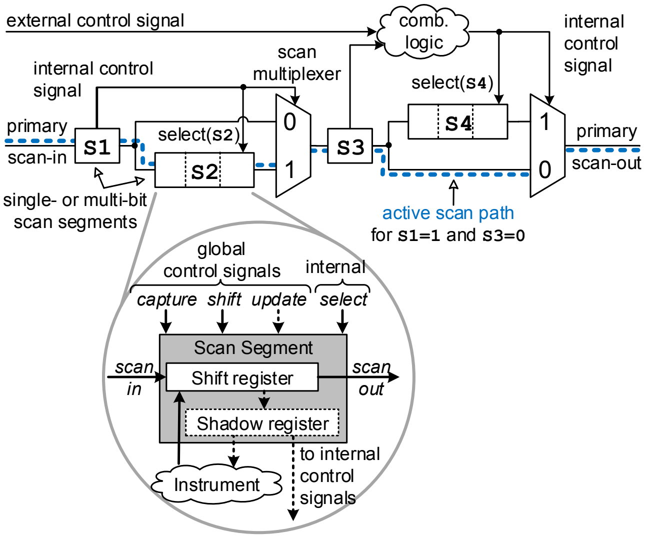

Figure 1. Example of a reconfigurable scan network [2]

The basic access to the RSN is an atomic operation consisting of the Capture, Shift, and Update phases (CSU), managed by the JTAG TAP controller and the corresponding control signals of scan segments. During *capture*, the scan segments on the active scan path may latch new data. This data is shifted out during the *shift* phase, while new data is shifted in. During the *update* phase, the shifted-in data is latched in the *shadow registers* of the scan segments on the active scan path, which may change the active scan path. A scan segment participates in these phases if and only if its *select* signal is asserted.

The state of all scan segments in the RSN is referred to as *scan configuration*. A scan configuration is valid if and only if an active scan path is formed and all elements that are not part of it are passive.

A read or write access to a scan segment in the network requires that the accessed segment is part of the active scan path in the current scan configuration. A *scan access* is a sequence of CSU operations, which reconfigure the RSN such that the target register(s) become part of the active scan path and which read or write the target register(s).

IEEE Std 1687 allows both hierarchical structures based on bypassing scan registers using Segment Insertion Bits (SIBs) as well as more flexible and irregular RSNs with arbitrary control signals and distributed configuration. This can cause combinational and sequential dependencies between accesses [20, 21], which makes access and test generation for general RSNs an NP-hard problem.

III. Related Work

The test of the RSN interface, i.e., the JTAG test access port (TAP), can be performed by functional tests that exercise its state machine [22]. To test the connectivity of scan cells in a conventional non-reconfigurable scan chain, a bit sequence called flush test or chain pattern is applied [13]. A flush bit sequence of alternating pairs of 0s and 1s ”00110011” applies all possible transitions in two cycles. This, however, is insufficient to cover all scan cell internal defects [18], even in conventional static scan chains. Dedicated tests for stuck-open latch-internal faults are derived in [14], or for bridges in [17]. The coverage of scan cell stuck-at and stuck-open faults is increased when the flush test is applied at reduced clock frequency [15]. In [16], test patterns for latch or flip-flop based scan chains are generated using ATPG targeting all scan cell internal combinational defects. The test of scan paths through functional (mission) logic is considered in [23], where combinational test generation is used to compute a test pattern for a fault in a scan path through mission logic, which is then validated by sequential fault simulation. This illustrates the challenges in test generation even for conventional scan chains. The much more complex structure of RSNs makes the test of them an even greater challenge.

Even the simple flush test cannot be directly applied to RSNs since the accessibility of scan cells in the scan registers of the RSN may already require complex reconfiguration. Using the scan access generation algorithms in [24, 5], it is possible to generate functional access sequences to scan registers with low test time. Such functional accesses to RSN elements can be generated with low computational effort to perform write and read operations by shifting in and out values, and can serve as a first basic test for RSNs.

However, to completely test a general RSN functionally, an exponentially high number of accesses may be required to cover all possible combinations of control signals in the RSN. Also, all faults in the structure should be targeted. As shown in Figure 2, a simple one-bit scan segment with shadow register is already synthesized into a complex netlist with feedback paths, flip-flops with inverted clocks and multiple multiplexers. Validating generated tests by fault simulation is mandatory to assess the fault quality in the actual hardware structure.

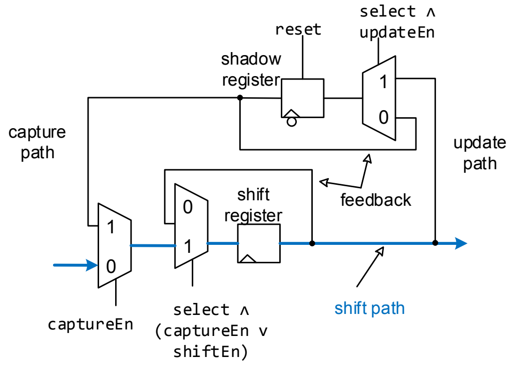

Figure 2. Structure of a 1-bit scan segment with shadow register after synthesis (shift path marked in blue)

In [25], a functional test method for a subset of IEEE Std 1687 scan networks has been proposed. The generated tests exercise scan multiplexers, bypass structures, and scan registers by multiple write and read accesses in different test sessions. Each test session constitutes a particular configuration of the RSN. While this method is applicable to certain RSN architectures, it does not consider more complex scan networks with combinational logic in control signals. This, however, is relevant if secure RSN architectures such as [26] need to be tested, or if legacy instruments with complex interfaces are connected to the RSN. In both cases, a functional test of a subset of scan paths in the RSN does not guarantee that the control logic is free of faults, which may compromise the security or limit accessibility. For the test generated in [25], the actually achieved structural fault coverage is not given.

IV. Test of Reconfigurable Scan Networks

This section discusses four test strategies targeting stuck-at faults in RSNs. The first discussed method is based on sequential test generation at gate level. The following two methods are functional test heuristics that access each scan segment in the RSN. The last method maps the test generation problem at an abstract level to an instance of the Boolean satisfiability problem to generate a scan access that activates the fault and allows its observation at the primary scan output.

A. Sequential Test Generation at Gate Level (TGGL)

In general, an RSN is a sequential circuit with a clock, reset, external control and primary scan-in signal as inputs, and a primary scan-out signal as output. The externally generated control signals include the capture, shift, and update enable signals for the corresponding access operations. The structural netlist of such a circuit can be provided to a sequential ATPG tool for test pattern generation targeting for instance stuck-at faults at all internal signals and cells.

In principle, this approach can generate a test pattern (corresponding to a sequence of capture, shift and update operations) for each testable fault, or prove the fault untestable otherwise. However, since sequential test generation is an NP-hard problem and even a single access to the RSN may require thousands of cycles (depending on the length of the active scan path), it is not apparent how well sequential ATPG tools can handle RSN designs.

To reduce the complexity of sequential test generation, it is possible to include the sequential elements of the RSN in an additional *non-reconfigurable* scan chain such that the control logic and access hierarchy in the RSN is bypassed. Because of the wiring overhead and incurred design overhead of such an additional design-for-test, it is desirable to avoid such a modification.

B. Simple Flush Test (SFT)

The basic idea in the simple flush test is to test the accessibility of each scan segment in the RSN at least once by making it part of an active scan path. Once the active scan path is formed, data can be shifted through all scan segments on that path. The data to be shifted can be a proven flush test for conventional scan chains. Here, the sequence ”00110011” of alternating pairs of 0s and 1s is used.

This simple functional test heuristic promises a good trade-off between the number of test patterns or CSU operations and fault coverage. The test time is minimized by merging the accesses to the tested scan segments by searching for a minimal number of scan accesses such that the flush sequence is shifted through each scan segment at least once. For general RSNs, this is achieved by the method of [5].

Implicitly, the capture and update paths are partially exercised in scan segments driving internal control signals if these signals are required to setup a particular active scan path. However, there may be faults whose effect does not propagate to the active scan path.

C. Test of Capture and Update Path (CUPT)

The simple flush test does not systematically exercise the capture or update paths in scan segments with shadow registers. These paths comprise a significant amount of logic that also needs to be tested. Figure 2 shows the netlist of a 1-bit scan segment with a shadow register obtained after logic synthesis. The figure shows the shift path, capture and update paths to the shadow register, and feedback loops. These structures also need to be tested to ensure correct communication to attached instruments and error-free generation of internal control signals. The method proposed in [25] functionally tests shadow registers and the corresponding paths only for segment insertion bits (SIBs) and scan segments that directly control scan multiplexers.

To test the capture and update circuitry in scan segments with shadow registers, write and read operations with opposite values are performed for each scan segment on the active scan path. Algorithm 1 below outlines the CSU operations to write a scan segment under test by the update operation and the check of updated values in the shadow register of s by the capture operation in the following CSU access. Since multiple segments can be placed on the active scan path and since the third CSU operation can already be used to reconfigure the RSN for the test of the next set of segments, the overall test time can be minimized by access merging.

Algorithm 1 Test for Capture and Update Paths

```
for all scan segments $s$ under test do
    Place $s$ on the active scan path
    // 1st CSU operation for the write-1 access:
    $s \leftarrow$ "1...1" // Update
    // 2nd CSU operation for the read-1 / write-0 access:
    Check $s =$ "1...1" after Capture and Shift-out
    $s \leftarrow$ "0...0" // Update
    // 3rd CSU operation for the read-0 access:
    Check $s =$ "0...0" after Capture and Shift-out
end for
```

D. Test Generation for Scan Path Alternation (TGSPA)

The previous two functional approaches do not exercise all the possible configurations of the RSN. Consequently, some faults may remain untested. A fourth, deterministic test generation method is developed to generate a scan access that activates a fault and propagates its effect such that the active scan path in the faulty circuit is affected. This may allow fault detection at the primary scan output when known data is shifted through the scan path.

In principle, it is possible to break the active scan path, so that the values at the scan output are independent of the values shifted in, for instance constant. Alternatively, the scan segments on the path and the length of the active scan path can be changed. This however can increase the test generation effort if the scan path length is encoded in the problem instance and the lengths of the active scan paths in the fault-free and faulty circuit need to be compared.

In this work, test patterns to break the scan path are generated using the transition relation of the RSN at transactional level in a SAT instance based on the model of [21]. A *scan path segment* is defined as the elements on the scan path between two adjacent scan segments. It may consist of a wire, multiplexers, or combinational gates (after synthesis). For each scan path segment p, the Boolean conditions for its sensitization are extracted and represented as function $\text{sens}(p)$. If a path segment has reconvergences or contains complex cells without controlling value, the Boolean difference $\partial p_o / \partial p_i$ of the path segment with input p_i and output p_o is modeled.

For an active scan path $\mathcal{P}$ from the primary scan input to the primary scan output, the condition to break the scan path in presence of fault f is given in Eq. (1), which is satisfied if there is one scan path segment p on the active scan path $\mathcal{P}$ that is not sensitized under fault f.

$$\exists p \in \mathcal{P}\ \forall q \in \mathcal{P} : \big((p = q) \rightarrow (\text{sens}^G(p) \land \neg\text{sens}^f(p))\big) \\ \land \big((p \neq q) \rightarrow (\text{sens}^G(q) \land \text{sens}^f(q))\big) \quad (1)$$

The term requires that the sensitization of the other scan path segments q is not affected by f. The term becomes part of the transition relation of the whole RSN and allows to search for a solution, i.e., a scan access, that activates the fault and causes a breakup of the scan path.

V. Implementation

The test generation methods have been implemented in C++. A Boolean satisfiability solver is used to analyze the sensitization conditions described in Section IV-D.

To process an RSN, its model at transaction level specified in instrument connection language (ICL, [7]) and its gate netlist is read. The transition relation of the RSN is extracted. The list of stuck-at faults is generated from the netlist. Test patterns for methods SFT and CUPT are then generated using transition relation of the RSN. For the structural test generation targeting to break the scan path (TGSPA), undetected faults are processed one at a time.

For the extracted test patterns, a sequential fault simulation at gate level is performed to validate the fault detection and to check if other faults can be dropped. This fault simulation is mapped to sequential logic simulation and serial fault injection in a commercial tool. Since a high number of cycles are simulated and compared in the testbench and the simulator needs to be reset intermittently, this causes very high runtimes for test validation. However, sequential fault simulation algorithms are beyond the scope of this paper.

VI. Evaluation

The fault coverage of functional and structure-oriented test sets is evaluated for stuck-at faults at the gate-level implementation of RSNs. The used circuits are described in Section VI-A. A standard tool for sequential test generation is used with high backtrack limit to generate patterns for the RSN netlists. Then, the achieved fault coverage is discussed.

A. RSN Circuits

For the evaluation, RSNs are constructed from the ITC'02 SOC benchmark set. Details on the construction is given in [20, 21]. All generated RSNs are synthesized for the generic lsi10k gate library. Table I lists the number of scan segments and stuck-at faults for the circuits.

Both hierarchical bypass (SIB) based RSNs as well as remotely controlled or multiplexer-based RSNs are considered. The multiplexer-based RSNs support two access modes, a configuration access to the scan segments that drive internal control signals, and a data access to the scan segments of attached instruments that have been put on the active scan path by a previous configuration access.

A third type of synthetic RSNs (called *ctrl* in the table) is generated to explore the limits of the different test generation methods when complex control logic is included. As illustrated in Figure 3, in this type of RSNs, the address signals of scan multiplexers are driven by a combinational circuit whose inputs are set by scan segments. Here, the ISCAS'85 circuits are used as combinational circuit. Faults in the upper scan segments and the control logic can limit the accessibility of scan segments in the lower part of the figure. A subset of these faults can only be detected by a change of the length of the active scan path.

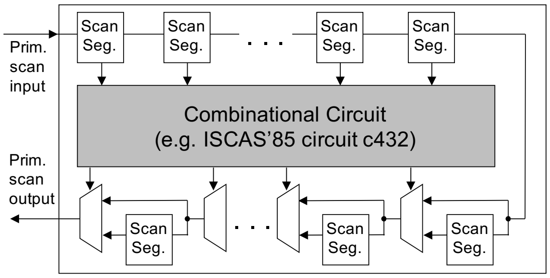

Figure 3. Structure of generated RSN with complex control logic

B. Fault Coverage Results

Table I tabulates the results for the four test methods of Section IV (TGGL: Test gen. at gate level; SFT: Simple flush test; CUPT: Test of Capture and update paths; TGSPA: Test gen. scan path alternation). For each method, the achieved stuck-at fault coverage ('FC') is given in percent. Column 'RT' shows the runtime in minutes, excluding the time for sequential fault simulation. For method SFT, the runtime is below one second for all circuits and thus omitted. Column 'NT' shows the number of generated scan accesses.

The last three columns of the table, titled 'TGSPA+CUPT', give the fault coverage and runtime of using the TGSPA method after test generation with CUPT and fault simulation. Column 'ΔNT' shows the additional scan accesses generated by the TGSPA method after CUPT.

For smaller RSNs, a high fault coverage can be achieved by sequential test generation at gate level (TGGL) since only very few cycles need to be considered. For larger RSNs, the coverage drops significantly and can be as low as 12%. The runtime of this approach is very high.

The fault coverage of the simple flush test (SFT) ranges from 39% to 88%, with an average fault coverage of 68%. Only very few tests are generated and the runtime of the test generation task is negligible. The more complex test generation method for capture and update paths (CUPT) increases the fault coverage to an average of 77%, ranging from 47% up to 89%. The number of tests increases, but is still low. Also the runtime for this method is very low with a maximum of 50 seconds for circuit p22810_mux.

Using the test generation approach targeting the alternation of the scan path (TGSPA), the fault coverage further increases. It ranges from 73% to 88%. Since the test generation is more complicated compared to the functional approaches SFT or CUPT, the runtime increases and can be high for larger circuits. The number of generated tests reaches 5272 tests for the largest multiplexer-based circuit. For this high number of tests, the sequential fault simulation, for which a separate commercial tool was used, becomes a bottleneck. Accurate fault coverage could not be computed for the largest circuits (marked as '—' in the table).

Finally, the methods CUPT and TGSPA are combined. The fault coverage increases for every RSN and reaches a maximum of 92% for circuit c432_ctrl. The runtime is significantly reduced compared to the exclusive application of TGSPA. For the majority of RSNs, the number of additional scan accesses is much lower than using TGSPA alone. However, for a few RSNs, e.g., c432_ctrl, the number is higher because of a different order of fault processing.

VII. Conclusion

The scalable access to on-chip infrastructure has been addressed by reconfigurable scan networks. Such serial scan-based access architectures pose novel challenges beyond the complexities of the test of conventional non-reconfigurable scan chains. For the first time, this paper discusses different test strategies for general reconfigurable scan networks and analyzes the achieved test quality for the stuck-at fault model. The combination of a functional and a structure-oriented deterministic test generation yields the most promising results and achieves high fault coverage even in synthetic, difficult to test circuits.

Acknowledgments

This work was supported by the German Research Foundation (DFG) under grant WU 245/17-1 (ACCESS).

References

[1] N. Stollon, *On-Chip Instrumentation: Design and Debug for Systems on Chip*. Springer US, 2011.

[2] 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.

[3] J. Rearick, B. Eklow *et al.*, "IJTAG (Internal JTAG): A Step Toward a DFT Standard," in *Proc. IEEE International Test Conference (ITC)*, 2005, paper 32.4.

[4] E. Larsson and F. Ghani Zadegan, "Accessing Embedded DfT Instruments with IEEE P1687," in *Proc. IEEE Asian Test Symposium (ATS)*, 2012, pp. 71–76.

[5] 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.

[6] "IEEE Standard for Test Access Port and Boundary-Scan Architecture 1149.1-2013," 2013, TTTC, IEEE Comp. Society.

[7] "IEEE Std 1687-2014 – IEEE Standard for Access and Control of Instrumentation Embedded within a Semiconductor Device," Dec. 2014, IEEE Computer Society.

[8] R. Guo and S. Venkataraman, "A technique for fault diagnosis of defects in scan chains," in *Proc. IEEE International Test Conference (ITC)*, 2001, pp. 268–277.

[9] F. Yang, S. Chakravarty *et al.*, "On the Detectability of Scan Chain Internal Faults - An Industrial Case Study," in *Proc. IEEE VLSI Test Symposium (VTS)*, 2008, pp. 79–84.

Table I: Fault coverage (FC), runtime (RT), and number of tests (NT) for the RSN test methods (TGGL: Test gen. at gate level; SFT: Simple flush test; CUPT: Test of Capture and update paths; TGSPA: Test gen. scan path alternation)

Name	Scan	Faults	TGGL		SFT		CUPT			TGSPA			TGSPA+CUPT		
	segm.		FC [%]	RT [min]	FC [%]	NT	FC [%]	NT	RT [min]	FC [%]	NT	RT [min]	FC [%]	ΔNT	RT [min]
c432_ctrl	43	2334	85	22.1	59	2	76	8	0.0	88	269	0.7	92	327	0.2
c1908_ctrl	58	4918	83	56.4	53	2	63	8	0.0	82	145	16.8	83	103	5.5
c880_ctrl	86	5584	12	200.9	59	2	70	8	0.0	81	209	13.0	88	221	5.0
c499_ctrl	73	5968	30	264.8	40	1	51	8	0.0	85	141	35.4	87	163	17.2
c1355_ctrl	73	5970	29	265.3	39	1	47	8	0.0	78	95	35.5	81	153	20.2
c3540_ctrl	72	6888	14	432.4	56	3	71	12	0.0	82	93	11.9	88	123	3.4
q12710_mux	51	2878	94	22.3	70	5	79	18	0.0	81	269	1.4	86	67	0.3
x1331_mux	63	3644	52	75.5	73	9	82	32	0.0	85	574	1.7	88	116	0.3
a586710_mux	79	4804	96	26.5	67	7	75	25	0.0	83	748	5.9	89	473	1.2
f2126_mux	81	5034	94	2244.3	62	5	71	18	0.0	82	731	8.0	89	578	2.0
DS9_mux	89	5448	23	70.3	74	19	81	67	0.0	84	2687	5.9	89	684	1.0
u226_mux	99	6092	45	133.9	65	5	72	18	0.0	84	915	12.7	90	619	4.1
h953_mux	109	6188	69	80.0	70	5	78	18	0.0	82	630	10.5	87	269	2.7
d281_mux	117	6952	90	68.1	66	5	75	18	0.0	84	888	15.3	90	504	3.2
g1023_mux	159	9814	88	131.8	67	5	75	18	0.0	82	1122	62.7	88	614	16.5
p34392_mux	245	14856	74	340.7	64	7	72	25	0.1	—	3122	166.9	—	1798	49.0
t512505_mux	319	19880	55	628.3	67	5	74	18	0.1	—	3016	428.8	—	1311	94.8
d695_mux	335	20312	62	309.2	67	5	70	18	0.1	—	4121	312.4	—	3057	101.3
p22810_mux	565	32302	22	1058.8	66	7	74	25	0.8	—	5272	1548.4	—	2714	378.5
q12710_sib	46	2106	95	23.9	76	3	86	12	0.0	76	42	0.5	87	1	0.0
x1331_sib	56	2552	48	98.2	83	5	89	20	0.0	83	86	0.8	89	1	0.0
a586710_sib	71	3142	58	198.0	78	4	87	16	0.0	77	24	1.0	88	3	0.0
f2126_sib	76	3316	75	143.8	77	3	87	12	0.0	77	24	1.3	87	2	0.0
DS9_sib	80	3702	16	401.1	75	10	84	40	0.0	73	358	2.0	85	67	0.3
u226_sib	89	3992	38	359.0	79	3	88	12	0.0	79	88	1.6	88	0	0.1
h953_sib	100	4412	37	352.9	78	3	88	12	0.0	78	20	2.2	88	2	0.1
d281_sib	108	4754	33	384.0	77	3	87	12	0.0	78	44	2.3	88	0	0.0
g1023_sib	144	6440	17	676.4	78	3	88	12	0.0	76	101	4.6	88	0	0.3
p34392_sib	225	9920	46	843.3	78	4	87	16	0.0	77	163	12.2	88	0	0.2
t512505_sib	287	12914	17	707.4	88	3	88	12	0.0	77	267	19.8	88	2	1.6
d695_sib	324	13950	31	1360.9	76	3	87	12	0.0	75	50	26.5	88	0	0.2
p22810_sib	536	23278	12	2870.9	77	4	87	16	0.0	—	101	82.8	88	5	4.8

[10] J.-S. Yang and S.-Y. Huang, "Quick scan chain diagnosis using signal profiling," in *Proc. International Conference on Computer Design (ICCD)*, Oct 2005, pp. 157–160.

[11] S.-Y. Huang, "Ch. 7.3 scan chain diagnosis," in *VLSI test principles and architectures*, L.-T. Wang, C.-W. Wu, and X. Wen, Eds. Morgan Kaufmann, 2006.

[12] F. G. Zadegan, D. Nikolov, and E. Larsson, "A self-reconfiguring IEEE 1687 network for fault monitoring," in *Proc. IEEE European Test Symp. (ETS)*, 2016, pp. 1–6.

[13] K.-J. Lee and M. Breuer, "A Universal Test Sequence for CMOS Scan Registers," in *Proc. IEEE Custom Integrated Circuits Conference (CICC)*, 1990, pp. 28.5.1–28.5.4.

[14] M. Reddy and S. Reddy, "Detecting FET Stuck-Open Faults in CMOS Latches and Flip-Flops," *IEEE Design & Test of Computers*, vol. 3, no. 5, pp. 17–26, Oct. 1986.

[15] F. Yang, S. Chakravarty *et al.*, "Detection of Internal Stuck-open Faults in Scan Chains," in *Proc. IEEE International Test Conference (ITC)*, 2008, paper 10.3.

[16] S. Maka and E. McCluskey, "ATPG For Scan Chain Latches and Flip-Flops," in *Proc. IEEE VLSI Test Symposium (VTS)*, Monterey, CA, USA, 1997, pp. 364–369.

[17] F. Yang, S. Chakravarty *et al.*, "Detectability of internal bridging faults in scan chains," in *Asia and South Pacific Design Autom. Conf. (ASPDAC)*, Jan 2009, pp. 678–683.

[18] R. Guo, L. Lai *et al.*, "Detection and diagnosis of static scan cell internal defect," in *Proc. IEEE International Test Conference (ITC)*, Oct 2008, pp. 1–10.

[19] Y. Huang, R. Guo *et al.*, "Survey of scan chain diagnosis," *IEEE Design and Test*, vol. 25, pp. 240–248, May/June 2008.

[20] R. Baranowski, M. A. Kochte, and H.-J. Wunderlich, "Modeling, Verification and Pattern Generation for Reconfigurable Scan Networks," in *Proc. IEEE International Test Conference (ITC)*, 2012, paper 8.2.

[21] R. Baranowski, M. A. Kochte, and H.-J. Wunderlich, "Reconfigurable Scan Networks: Modeling, Verification, and Optimal Pattern Generation," *ACM Trans. Design Automation of Electronic Systems (TODAES)*, vol. 20, no. 2, pp. 30:1–30:27, Feb. 2015.

[22] A. Dahbura, M. Uyar, and C. W. Yau, "An Optimal Test Sequence for the JTAG/IEEE P1149.1 Test Access Port Controller," in *Proc. IEEE International Test Conf. (ITC)*, 1989, pp. 55–62.

[23] D. Chang, M. T. C. Lee *et al.*, "Functional scan chain testing," in *Proc. Design, Automation and Test in Europe (DATE)*, Feb 1998, pp. 278–283.

[24] F. Ghani Zadegan, U. Ingelsson *et al.*, "Design Automation for IEEE P1687," in *Proc. Design, Automation Test in Europe Conference (DATE)*, 2011, pp. 1412–1417.

[25] R. Cantoro, M. Montazeri *et al.*, "On the Testability of IEEE 1687 Networks," in *Proc. IEEE Asian Test Symposium (ATS)*, 2015, pp. 211–216.

[26] R. Baranowski, M. A. Kochte, and H.-J. Wunderlich, "Fine-grained access management in reconfigurable scan networks," *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, vol. 34, no. 6, pp. 937–946, June 2015.