Structure-oriented Test of Reconfigurable Scan Networks

Ull, Dominik; Kochte, Michael A.; Wunderlich, Hans-Joachim

Proceedings of the 26th IEEE Asian Test Symposium (ATS'17) Taipei, Taiwan, 27-30 November 2017

doi: http://dx.doi.org/10.1109/ATS.2017.34

Abstract: Design, production and operation of modern system-on-chips rely on integrated instruments, which range from simple sensors to complex debug interfaces and design-for-test (DfT) structures. Reconfigurable scan networks (RSNs) as defined in IEEE Std. 1687-2014 provide an efficient access mechanism to such instruments. It is essential to test the access mechanism itself before it can be used for test, diagnosis, validation, calibration or runtime monitoring. Realistic fault mechanisms in RSNs are hard to test due to their high sequential depth and limited controllability and observability via serial scan ports. We present a novel low-cost DfT modification specifically designed for RSNs that enhances the observability of shadow registers. Furthermore, we present different test methods for stuck-at and more realistic gate-level fault models like flip-flop- internal and bridge faults. Experimental results demonstrate the effectiveness of the presented DfT modification and test methods.

Structure-Oriented Test of Reconfigurable Scan Networks

Dominik Ull, Michael A. Kochte, Hans-Joachim Wunderlich
email: {ull, kochte}@iti.uni-stuttgart.de, wu@informatik.uni-stuttgart.de
ITI, University of Stuttgart, Pfaffenwaldring 47, D-70569, Stuttgart, Germany

***Abstract*—Design, production and operation of modern system-on-chips rely on integrated instruments, which range from simple sensors to complex debug interfaces and design-for-test (DfT) structures. Reconfigurable scan networks (RSNs) as defined in IEEE Std. 1687-2014 provide an efficient access mechanism to such instruments. It is essential to test the access mechanism itself before it can be used for test, diagnosis, validation, calibration or runtime monitoring. Realistic fault mechanisms in RSNs are hard to test due to their high sequential depth and limited controllability and observability via serial scan ports.**

We present a novel low-cost DfT modification specifically designed for RSNs that enhances the observability of shadow registers. Furthermore, we present different test methods for stuck-at and more realistic gate-level fault models like flip-flop-internal and bridge faults. Experimental results demonstrate the effectiveness of the presented DfT modification and test methods.

***Index Terms*—Design-for-test, test generation, reconfigurable scan network, on-chip infrastructure, IEEE Std. 1687, iJTAG**

I. Introduction

Embedded instrumentation and their access has become essential in the whole product lifecycle of integrated circuits [1]. DfT structures, e.g. built-in self-test (BIST) or IEEE 1500 test wrapper, support manufacturing test, diagnosis, bring-up, post-silicon validation, and in-field maintenance. Debug interfaces are integrated for software development and debugging. Recent architectures for online fault management and fault tolerance provide transparent access to sensors and monitors or even self-reconfiguration [2, 3]. Design requirements, minimization of test costs and runtime performance constraints mandate efficient, low latency access to such instruments, which may be distributed over the complete design. With a growing number of instruments, the original JTAG access mechanism (IEEE Std. 1149.1) exhibits flexibility and scalability problems: Fixed-size instrument registers are either concatenated in serial scan chains or multiplexed in parallel, leading to high access time or expensive wiring effort. Recently, scan-based access architectures with *reconfigurable* scan paths have been standardized in IEEE Std. 1149.1-2013 and IEEE Std. 1687-2014. The scan network can be (re)configured such that the targeted instrument registers become part of the active scan path, while other registers have no impact on the scan path length and access time. This reconfigurability is provided by additional control logic and shadow registers, which control the active scan path. The flexibility of such reconfigurable scan networks (RSNs) allows for different optimized topologies, including hierarchical architectures. Formal methods for minimization of access latencies have been developed in [4, 5].

As RSNs are employed in a variety of critical run-time and cost-related production applications [6], it is especially important to test their integrity. In comparison to non-reconfigurable scan chains, the test of RSNs poses additional challenges. In functional scan chain testing [7], so-called flush sequences of alternating bits are shifted through the entire scan path. This approach detects stuck-at faults on the scan path, broken chains and non-shifting scan cells, but is not directly applicable for RSNs. In RSNs there is a significantly higher number of possible scan paths (possibly exponential in the number of registers), and selecting a specific scan path requires complex reconfiguration sequences [8]. Also, the scan path represents only a part of a reconfigurable scan cell's structure and functionality. Flush tests neither target shadow registers nor control and reset logic, which are both essential parts in RSNs. In addition, faults in the reset logic may only be detectable in very specific configurations. RSN testing becomes even more challenging by uncertainties that stem from instruments. Sensors provide indeterministic data, and intellectual property (IP) restricts the tester's knowledge on instrument behaviour and structure.

Experiments on scan chains have shown that functional or stuck-at faults only cover a fraction of realistic flip-flop-internal faults [7, 9–12], which may account for half of all scan chain failures [13]. For instance, an internal latch-stuck-at fault makes the flip-flop transparent, i.e. behaving like a combinational buffer. This faulty behaviour is especially relevant in RSNs as scan paths lengths are altered and scan-based test and diagnosis methods depend on the scan path length [14–17]. In general, such faults are hard to test as they may introduce combinational loops to the circuit.

In this work, we present a novel DfT structure for RSNs that is necessary to observe shadow registers and the update logic. Based on this modification, we introduce test methods for stuck-at, flip-flop-transparency and bridge faults in the whole RSN structure. To the authors' best knowledge, so far only RT-level functional faults in a subset of IEEE Std. 1687 networks [17] and gate-level stuck-at faults in general RSNs [16] have been investigated. We evaluate our work with gate-level fault simulation and achieve an average coverage of 99% for stuck-at, 100% for flip-flop transparency, and 83% for bridge faults. When test generation targets multiple segments at once, a test speedup over 2x and test generation speedup of 34x is achieved.

The next section briefly describes the structure and behaviour of RSNs, followed by a review of related work on scan network testing. Section IV describes the novel DfT modification for RSNs. Section V defines test generation methods for complex faults in RSNs. Sections VI and VII explain the implementation and report on experimental results.

II. Reconfigurable Scan Networks (RSNs)

This section explains the structure and behaviour of RSNs, following the terminology of previous work [4, 6, 8, 16, 18] and IEEE Std. 1687 [19]. An RSN consists of scan segments, scan multiplexers, and control logic. It provides a scan interface and control inputs. RSNs are either driven by automatic test equipment (ATE) as in production test or through a JTAG test access port (TAP) [20].

A *scan segment* is either a data or scan register. Data registers are used to communicate with instruments. Scan registers drive control logic including the address inputs of scan multiplexers. Multiplexers determine the shift path depending on the state of scan registers.

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 scan input of the RSN through selected segments and scan multiplexers to the scan output is called the *active scan path*. Scan segments that are not part of the active scan path are *de*selected. Shadow flip-flops have distinct reset states, which determine the initial active scan path.

A register consists of one or multiple scan cells with common control signals and a scan input (scanIn), scan output (scanOut), data input (D) and data output (Q). A scan cell, as shown in Fig. 2, has a scan (shift) flip-flop, placed between scanIn and scanOut. It may have an optional *shadow* flip-flop which is implemented if the cell provides write access to an attached instrument or drives control signals. The state of all shadow flip-flops in an RSN is called its *scan configuration*.

Global control signals in the RSN trigger three operations on all scan cells of selected segments: The *capture* operation (C) captures a scan cell's data input D into its scan flip-flop. The *shift* operation (S) stores a scan cell's scan input value in its scan flip-flop. The *update* operation (U) latches the scan flip-flop state into the shadow flip-flop. A read/write access to the selected segments is called a CSU operation, comprising a capture cycle, multiple shift cycles (corresponding to the length of the active scan path), and a final update cycle. The initial capture loads instrument data to the scan path, which is then serially shifted out. In the final update cycle, newly shifted-in data is written to the shadow register. Accesses to multiple segments can be merged and mapped to a sequence of CSU operations with minimal access time [4].

RSNs allow to construct different topologies, for instance hierarchies in which access to lower levels can be dynamically en- or disabled. In IEEE Std. 1687, hierarchies are either based on scan multiplexers or Segment Insertion Bits (SIBs). A SIB contains a one-bit scan register and a scan multiplexer which either bypasses or selects a lower level (when set to 1).

Fig. 1 depicts an exemplary RSN with 3 instruments ($INSTR_{1-3}$) accessible via data registers TDR_{1-3}. Shadow registers are shown in grey. Further control logic and signals are omitted. Scan register SR_1 controls the scan multiplexers MUX_1 and MUX_2 through its shadow register. The first multiplexer either asserts TDR_1 or selects a bypass. The data inputs of TDR_1 are connected to outputs of $INSTR_1$. Hence, $INSTR_1$ is a read-only instrument and requires no shadow flip-flops. The second multiplexer either asserts TDR_2 or SIB_1. TDR_2 communicates with the instrument $INSTR_2$ via its shadow register. Data inputs of TDR_2 capture instrument outputs. When SIB_1 is asserted and set to 1, the data register TDR_3 becomes accessible. It provides bi-directional access to $INSTR_3$. Suppose the shadow flip-flop of SR_1 has the reset value 0. The corresponding active scan path is (SR_1, TDR_1, TDR_2). Access to $INSTR_3$ requires 2 CSU operations. The first one, CSU(1XXXXX), sets SR_1 to 1. This changes the active scan path to (SR_1, SIB_1). A second operation CSU(11) sets SR_1 and SIB_1 to 1, and the active scan path becomes (SR_1, SIB_1, $INSTR_3$). A final CSU operation is now able to read or write TDR_3 resp. $INSTR_3$.

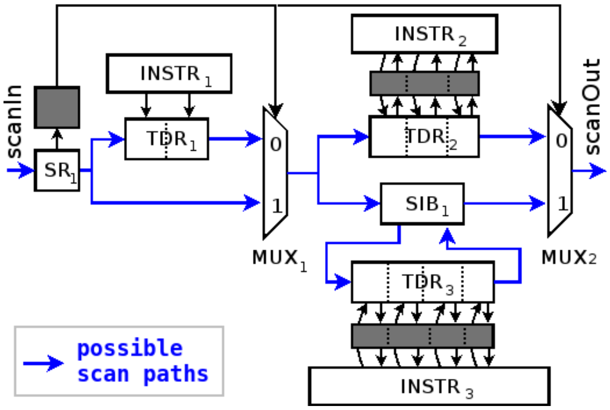

Fig. 1. Example of a reconfigurable scan network (with shadow registers)

III. Related Work

The test of a JTAG TAP can be performed by a functional test sequence [21]. The connectivity of cells in a static scan chain can be tested by a chain pattern or flush test [22]. The flush sequence "00110011..." applies all possible transition in two cycles. Dedicated tests for cell-internal and bridge faults can be generated deterministically [7, 23].

In [17] and [24], a graph-based method to select a set of RSN configurations for flush tests with minimized test time is presented. This method targets a functional fault model. Since the access graph does not model control logic, it is only suitable for a subset of IEEE Std. 1687 networks, but not for general networks allowed by the standard. Also, the reset functionality in RSNs is not considered.

In [16], different RSN test strategies are introduced and compared w.r.t. stuck-at fault coverage. Functional test generation as well as structural test generation tailored for general RSNs are developed. The highest coverage was achieved by a hybrid approach that performs structure-oriented tests on top of functional accesses. Sequential stuck-at fault simulation was a bottleneck for larger RSNs.

A challenge of RSN tests are the observation of shadow flip-flops. The data output of a scan cell is only visible if it drives RSN control logic and affects the scan path. If an instrument is connected to the data output, it is unobservable unless the state of the instrument can be extracted during the test of the RSN. This coordination of RSN test, e.g. via the TAP, and interaction with the instrument leads to a more complex test sequence and may require costly protocol-aware test equipment.

In this work, we develop a DfT modification for RSNs for improved testability and test generation approaches for complex fault models. Both are applicable to general RSNs including control and reset circuity, which may account for a significant part of an RSN.

IV. Design-for-Test Modification for RSNs

In this section we describe a DfT modification for RSNs that solves the problem of limited observability of shadow flip-flops and decouples the test of the RSN structure from attached instruments. Our DfT modification adds a feedback path from the shadow flip-flop to the scan flip-flop as detailed below:

Fig. 2 shows a reconfigurable scan cell. It consists of shift and update circuitry. The scan flip-flop keeps its previous value (when $\neg Select \vee (\neg ShiftEn \wedge \neg CaptureEn)$) or latches new data either from scanIn ($Select \wedge ShiftEn$) or from an instrument through data input D ($Select \wedge CaptureEn$). The shadow flip-flop on the right either keeps its previous value ($\neg Select \vee \neg UpdateEn$) or the value from the shift flip-flop ($Select \wedge UpdateEn$). It drives the data output Q, which may be connected to an instrument or drive RSN-internal control signals. In general, a reconfigurable scan cell includes the following paths:

- The *scan path* starts at the scanIn port and ends at the scanOut port of a cell. In Fig. 2, it contains the two multiplexers controlled by CaptureEn and ShiftEn, and includes the internal data path of the scan flip-flop.
- The *update path* starts at the scan flip-flop's output and leads to the data output Q. It comprises the multiplexer controlled by UpdateEn and the data path of the shadow flip-flop.

The introduced DfT feedback path (highlighted in Fig. 2) connects the shadow flip-flop's output to the scan flip-flop's input when the control signal FeedbackEn is 1. Hence, it provides direct visibility of the shadow stage without requiring knowledge about or control over the connected instrument. Only the actual interconnection to the instrument in not included in this test.

This modification is compliant with IEEE Std. 1687 and can be described in the Instrument Connectivity Language (ICL). Therefore it can be readily handled by EDA tools supporting this standard.

There are several possibilities to control the feedback: FeedbackEn can be declared as global wire controlled by the ATE or JTAG TAP. However, this trivial approach may incur expensive wiring effort. Secondly, FeedbackEn can be controlled with additional scan cells, either globally or locally. Using a single global scan cell exhibits comparable wiring effort like the first solution. Using segment- or hierarchy-specific scan cells avoids the global wiring effort at the cost of additional scan cells. In both cases, existing access patterns and access generation tools can be easily adapted to the new network topology with pattern retargeting [4]. FeedbackEn can also be controlled by using previously unused assignments to existing control signals. This solution avoids global wiring and requires no additional scan cells. For example, when using the assignment ($ShiftEn \wedge CaptureEn$) to enable the feedback, the overhead of feedback control is only one AND-gate per scan cell. The wiring effort is neglectable since the re-used control signals are already locally available.

V. RSN Test Generation

This section describes the test generation methods for stuck-at and flip-flop transparency faults.

A. Test of stuck-at faults

Let us consider the scan cell from Fig. 2. Stuck-at faults on the scan path are detected by preceeding the shift part of a CSU operation with a flush sequence of alternating

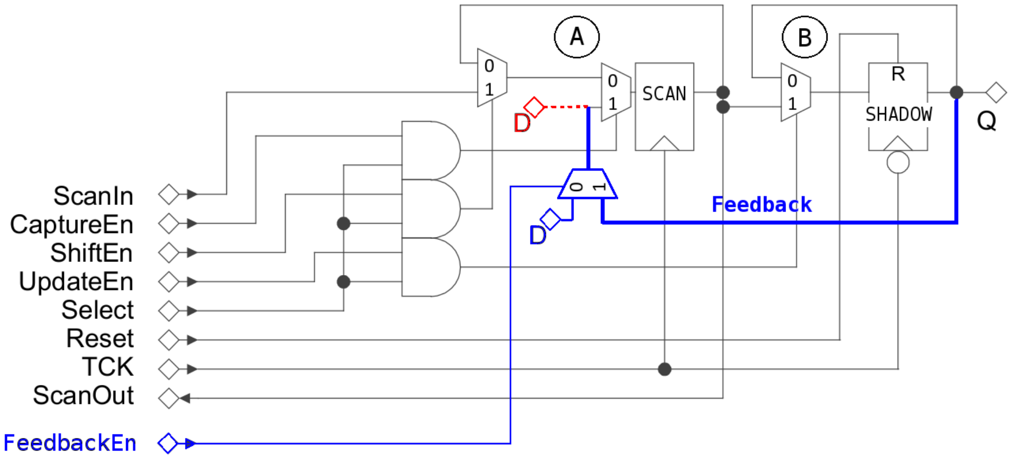

Fig. 2. Reconfigurable scan cell according to IEEE Std. 1687, and with DfT feedback path (in blue).

values. The update path can be exercised by writing arbitrary complementary values to the shadow flip-flop. Note that this is only possible for data registers — writing to scan registers that control the scan path may leave the RSN in a non-functional state or restrict further access.

Fault effects stored in the shadow flip-flop can be moved to the scan path using the DfT feedback path. The feedback path is therefore testable for the same stuck-at faults as the shadow flip-flop's in-/output. Faults in the DfT additions (multiplexer / AND-gate) are testable, with the exception of faults at the FeedbackEn signal which are possibly detectable. Instrument data port D is unknown, so FeedbackEn cannot be deterministically propagated through the DfT multiplexer. The remaining paths (labelled A and B in Fig. 2) enable the flip-flops to keep their values. They are exercised by idle cycles. Path A is in use when $\neg select \vee (\neg CaptureEn \wedge \neg ShiftEn)$. Path B is exercised when no update occurs $(\neg select \vee \neg UpdateEn)$. We define the following two kinds of idle cycles:

- Control *idle cycle*:

$$(\neg CaptureEn \wedge \neg ShiftEn \wedge \neg UpdateEn)$$

- Select *idle cycle*: $(ShiftEn \wedge \neg select)$

These two variants of idle cycles exercise different control logic and should therefore be both applied. To observe faults on paths A and B, they must be propagated through the scan path to the scan output. This is satisfied when applying the *control idle cycle* prior to capture operations (followed by shifts) and the *select idle cycle* directly before shifting. Activating further control faults requires control signal assignments that do not occur in CSU operations.

In order to test the reset functionality of shadow flip-flops, an RSN must first be brought to a known state which differs from its reset state. Subsequently, a global reset is applied for fault activation. Finally, the shadow registers are read via the feedback path in order to observe captured fault effects. The following procedure summarizes this so-called reset test:

1) Set shadow flip-flops to their inverted reset state.
2) Apply a reset.
3) Read shadow flip-flops via feedback path and shift-out.
4) Apply a reset.

Note that the read accesses in step 3 typically require reconfigurations of the RSN and therefore can overwrite control-related shadow flip-flops, making their reset functionality only potentially testable. This problem depends on the network topology and can be solved with network re-design. Steps 1 and 3 can be realized with merged accesses to all segments at once. The formal solver in the RSN access generator [4] is asked for a CSU sequence with minimal access time, so that all segments are accessed accordingly. This increases the optimization potential of the solver and leads to CSU operations with an overall shorter access (and thus test) time. We use merged accesses to apply the following test access patterns to all segments at once:
write(data=01...01), read(), write(data=10..10), read().

B. Test of flip-flop transparency faults

A flip-flop transparency fault in a scan flip-flop can be detected with a flush sequence, since it shortens the active scan path by one bit. However, the test of shadow flip-flops for transparency is challenging. In general, a flip-flop transparency fault in a negative-edge-triggered shadow flip-flop is tested when three conditions hold:

- *Condition 1*: The signal value of its data input D differs from the currently stored state. Note that a faulty flip-flop will still hold its value due to the combinational feedback from Q to D when no Update operation is performed (UpdateEn=0).
- *Condition 2*: The Q output is captured at the same point in time when condition (1) holds.
- *Condition 3*: The captured value is propagated to an observable primary output (i.e. shifted to scanOut).

The following test procedure satisfies at least the first two conditions for all update flip-flops on the active scan path:

1) Fill the active scan path with alternating values.
2) Apply an update operation to transfer the alternating sequence to the update flip-flops.
3) Apply a single shift to satisfy condition (1) when a subsequent update operation occurs.
4) Apply an update and capture operation simulaneously at rising edge (UpdateEn=1 & CaptureEn=1). This step satisfies condition (1) and (2).
5) Shift out the complete scan path to satisfy condition (3).

VI. Implementation and Experimental Setup

The test generation methods are implemented in C++ and work on RT-level Instrument Connectivity Language (ICL) descriptions [19] of RSNs. For the generation of CSU sequences, the method of [4] has been extended. The gate-level fault coverage is determined with a commercial sequential stuck-at fault simulator. For complex fault models (flip-flop transparency and bridging faults), the faults are mapped to stuck-at faults using fault injection cells in the netlist that model the faulty behaviour.

The implemented test methods are evaluated on gate-level descriptions of benchmark RSNs similar to the ones used in [16]. In this work, we utilize fault sampling [25] to overcome simulation runtime problems. The first set of benchmarks (ending in *_ctrl*) represents synthetic mux-based RSNs with complex control logic. The second set is constructed from ITC'02 System-on-Chip (SoC) benchmarks. Corresponding to the hierarchy of each SoC benchmark, we construct a corresponding SIB-based RSN structure. Details of this construction can be found in [16, 18]. All benchmark circuits are generated as ICL descriptions (for test generation) and Verilog descriptions for gate-level synthesis. In synthesis, the library lsi_10k with basic gate types (supporting up to 4 inputs) is used. The previously used multiplexer-based RSNs [16] are omitted here due to limited space. The synthetic RSNs ending in *_ctrl* are more difficult to test than these multiplexer RSNs because of their much more complex control logic.

For the synthesized benchmark circuits we generate uncollapsed fault lists. We generate uncollapsed stuck-at faults, flip-flop transparency for every flip-flop and the following kinds of bridges between signals of gate distance 1: byzantive (4-way), dominated, AND-dominated, OR-dominated, wired-AND and wired-OR bridges.

The following pattern sets are generated and evaluated:

- *H*: Serial access heuristic with flush sequence and idle cycles before shift.
- *RST+H*: Reset test followed by pattern set *H*.
- *RST+M*: Reset test followed by merged accesses with flush sequence and idle shifts.
- *RST+M+F*: Pattern set *RST+M* extended by the flip-flop transparency test for shadow flip-flops.

VII. Evaluation

The generated patterns are evaluated in altogether 456 fault sampling experiments (19 benchmarks * 4 pattern sets * 3 fault models with / without feedback). 30 simulation jobs were executed in parallel on 4 Intel Xeon E5-2650 CPUs (with 8 cores each). This evaluation took approximately 11 days.

Table I shows characteristics of the utilized benchmark RSNs (gate count, number of flip-flops) as well as fault coverage results for stuck-at, flip-flop transparency and bridge faults. For each fault model, the following figures are provided: the total number of faults, a fault coverage confidence interval and the rate of possibly detected faults as reported by the simulation tool. Fault sampling [25] has been used to determine fault coverage intervals with a confidence niveau of 99%. A sample contains the recommended number of 1000 faults. When the total number of faults is below 1000, i.e. flip-flop faults of c17_ctrl and c432_ctrl, the exact fault coverage is given.

A fault detection confidence interval $c_{99\%}$ is determined by solving equation (2) from [25] with c being the sample fault coverage, $\alpha = 2.6$ and $k = 1 - (\frac{N_s}{N_t})$ for N_t total faults and N_s sampled faults:

$$c_{99\%} = c \pm \frac{\alpha^2*k}{2*N_s}\sqrt{\frac{1+4*N_s*c*(1-c)}{\alpha^2*k}}$$

The achieved average stuck-at coverage is 99%. Some remaining faults are classified as possibly detected. This includes stuck-at faults on the clock signal, and faults on the control line FeedbackEn. For flip-flop transparency faults, the reported average coverage for pattern set $RST + M + F$ is 100%. For bridge faults, $RST + M + F$ attains an average coverage of 83% with 15% being possibly detected. Please note that [7] has shown that approximately 12% of bridge faults cannot be detected by logic testing. The high number of possibly detected bridges stems from several facts: Hard-to-test bridges may be located between scan and control logic or concern clock/reset signals, instrument data is unknown, and the initial network state as well. Some clock-related faults are reported by the tool as possibly detected although implication methods show they are in fact detected. Table II presents generation time, test length (in number of cycles) and fault coverage numbers averaged over all benchmark circuits for the generated test patterns and investigated fault models. On average, adding the DfT feedback increases fault coverage by 52 percent points for stuck-at faults, by 50 percent points for flip-flop transparency faults, and by 37 percent points for bridge faults.

The comparison of pattern sets *H* and *RST+H* shows that the *reset test* requires approx. 22% of the serial heuristic's test time and increases coverage by 4 percent point for stuck-at, by 43 percent points for flip-flop transparency, and by 9 percent points for bridge faults. When generating merged (*RST+M*) instead of serial (*RST+H*) test accesses, test time decreases by a factor of 2.2x. At the same time, test generation time is reduced significantly since only a single (yet large) access problem is solved instead of a high number of simpler instances. The comparison of *RST+M* and *RST+M+F* shows that the flip-flop transparency test increases the flip-flop transparency coverage by 4 percent points. This test also improves bridge fault coverage by 3 percent points.

TABLE I. Circuit characteristics and fault coverage results for pattern set *RST+M+F* and RSNs with DfT feedback.

circuit	# gates	# flip-flops	# test cycles	stuck-at faults			FF transparency faults			bridge faults		
				#	$c_{99\%}$	pos.det.	#	$c_{99\%}$	pos.det.	#	$c_{99\%}$	pos.det.
c17_ctrl	526	138	598	3136	98.7±0.8%	0.2%	138	100.0%	0.0%	16146	84.1±2.9%	13.5%
c432_ctrl	1938	520	4809	12078	96.2±1.5%	0.0%	520	100.0%	0.0%	63585	86.9±2.8%	10.5%
c3540_ctrl	5786	1508	37362	36544	96.8±1.5%	0.0%	1508	98.6±0.6%	0.0%	193329	85.1±2.9%	11.9%
c1908_ctrl	6279	1666	13914	37670	97.8±1.2%	0.0%	1666	99.4±0.4%	0.0%	194796	82.3±3.1%	15.4%
c880_ctrl	6741	1784	32433	40252	99.0±0.9%	0.0%	1784	99.5±0.4%	0.0%	209097	83.2±3.2%	15.5%
c6288_ctrl	9876	2112	9345	57526	99.4±0.7%	0.0%	2112	100.0±0.2%	0.0%	293589	83.2±3.0%	13.7%
c499_ctrl	7994	2130	9632	47646	90.0±0.9%	0.0%	2130	100.0±0.2%	0.0%	245934	84.9±3.0%	12.8%
c1355_ctrl	7994	2130	9632	47646	99.3±0.7%	0.0%	2130	100.0±0.2%	0.0%	245934	81.6±3.2%	16.3%
u226	6235	2930	15536	64308	99.0±0.9%	0.1%	2930	99.9±0.3%	0.1%	333864	82.9±3.1%	14.9%
d281	15921	7742	32863	165872	99.7±0.6%	0.0%	7742	99.0±0.8%	1.0%	857439	82.1±3.2%	16.1%
x1331	16344	8046	31033	170812	99.8±0.5%	0.0%	8046	99.5±0.6%	0.5%	880533	81.6±3.2%	17.0%
g1023	22137	10770	46443	230602	99.9±0.4%	0.0%	10770	99.4±0.7%	0.6%	1191897	80.7±3.3%	17.4%
h953	22951	11280	44296	239868	99.9±0.4%	0.0%	11280	99.4±0.7%	0.6%	1237923	83.1±3.1%	14.1%
d695	34927	16792	84026	362346	99.3±0.7%	0.1%	16792	99.0±0.9%	1.0%	1873269	77.8±3.4%	20.0%
f2126	63624	31658	114563	667106	100.0±0.3%	0.0%	31658	100.0±0.3%	0.0%	3433986	81.2±3.2%	16.1%
p34392	93915	46482	181591	983214	100.0±0.3%	0.0%	46482	99.4±0.7%	0.6%	5064426	83.3±3.1%	15.6%
q12710	104961	52366	184971	1101316	99.9±0.4%	0.0%	52366	99.9±0.4%	0.1%	5665581	84.8±3.0%	13.2%

TABLE II. Average pattern set characteristics and fault coverage over all circuits.

pattern set	gen. time [sec.]	test time [# cycles]	stuck-at faults f.cov.	stuck-at faults pos.det.	FF transp. faults f.cov.	FF transp. faults pos.det.	bridge faults f.cov.	bridge faults pos.det.
H (with feedback)	34	53686	95%	4%	53%	43%	72%	25%
H (w/o feedback)			47%	1%	50%	2%	43%	13%
RST+H (with feedback)	34	65707	99%	0%	96%	0%	81%	17%
RST+H (w/o feedback)			47%	1%	50%	0%	47%	10%
RST+M (with feedback)	1	30022	99%	0%	96%	0%	80%	17%
RST+M (w/o feedback)			46%	0%	50%	0%	45%	10%
RST+M+F (with feedback)	7	50179	99%	0%	100%	0%	83%	15%
RST+M+F (w/o feedback)			47%	1%	50%	0%	46%	10%

VIII. Conclusion

This paper presents a novel DfT modification for reconfigurable scan networks to improve observability of its shadow flip-flops. Its effectiveness is demonstrated with tests for stuck-at faults and more complex flip-flop transparency and bridge faults, that are relevant for scan infrastructure. We developed an efficient test generation method for the discussed fault models. By merging test accesses, the test time is significantly reduced. The fault coverage has been estimated using gate-level fault sampling in sequential fault simulation. We achieved an average fault coverage of 99% for stuck-at, 100% for flip-flop transparency and 83% for different bridge faults.

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] A. Jutman, S. Devadze, and K. Shibin, "Effective scalable IEEE 1687 instrumentation network for fault management," *IEEE Design & Test*, vol. 30, no. 5, pp. 26–35, 2013.

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

[4] R. Baranowski, M. A. Kochte, and H.-J. Wunderlich, "Scan pattern retargeting and merging with reduced access time," in *Proc. IEEE European Test Symposium (ETS)*, 2013, pp. 1–6.

[5] R. Krenz-Baath, F. G. Zadegan, and E. Larsson, "Access time minimization in IEEE 1687 networks," in *Proc. IEEE International Test Conf. (ITC)*, 2015, pp. 1–10.

[6] M. A. Kochte and H.-J. Wunderlich, "Dependable on-chip infrastructure for dependable MPSOCs," in *Proc. IEEE Latin-American Test Symposium (LATS)*, 2016, pp. 183–188.

[7] S. Makar and E. J. McCluskey, "ATPG for scan chain latches and flip-flops," in *Proc. IEEE VLSI Test Symposium (VTS)*, 1997, pp. 364–369.

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

[9] S. R. Makar and E. J. McCluskey, "Checking experiments to test latches," in *Proc. IEEE VLSI Test Symposium (VTS)*, 1995, pp. 196–201.

[10] S. R. Makar and E. J. McCluskey, "Iddq test pattern generation for scan chain latches and flip-flops," in *Proc. IEEE International Workshop on IDDQ Testing*, 1997, pp. 2–6.

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

[12] F. Yang, S. Chakravarty *et al.*, "Improving the detectability of resistive open faults in scan cells," in *Proc. IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems (DFT)*, 2009, pp. 383–391.

[13] F. Yang, S. Chakravarty *et al.*, "Detection of internal stuck-open faults in scan chains," in *Proc. IEEE International Test Conf. (ITC)*, 2008, pp. 1–10.

[14] Y. Huang, R. Guo *et al.*, "Survey of scan chain diagnosis," *IEEE Design & Test of Computers*, vol. 25, no. 3, 2008.

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

[16] M. A. Kochte, R. Baranowski *et al.*, "Test strategies for reconfigurable scan networks," in *Proc. IEEE Asian Test Symposium (ATS)*, 2016, pp. 113–118.

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

[18] 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.

[19] "IEEE Std 1687-2014 IEEE standard for access and control of instrumentation embedded within a semiconductor device," 2014, IEEE Computer Society.

[20] E. Larsson and F. G. Zadegan, "Accessing embedded DfT instruments with IEEE P1687," in *Proc. IEEE Asian Test Symposium (ATS)*, 2012, pp. 71–76.

[21] 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.

[22] K.-J. Lee and M. Breuer, "A universal test sequence for CMOS scan registers," in *Proc. IEEE Custom Integrated Circuits Conf. (CICC)*, 1990, pp. 28.5.1–28.5.4.

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

[24] R. Cantoro, M. Palena *et al.*, "Test time minimization in reconfigurable scan networks," in *Proc. IEEE Asian Test Symposium (ATS)*, 2016, pp. 119–124.

[25] V. D. Agrawal, "Fault sampling revisited," *IEEE Design & Test of Computers*, vol. 7, no. 4, pp. 32–35, 1990.