Enhanced DFT for Fortuitous Detection of Transition Faults During Scan Shift

Hui Jiang*, Jennifer Dworak*, Kundan Nepal†, Theodore Manikas*,
* Lyle School of Engineering, Southern Methodist University, Dallas Texas, 75205
†School of Engineering, University of St. Thomas, Saint Paul, Minnesota 55105

2022 IEEE 31st Microelectronics Design & Test Symposium (MDTS) | 978-1-6654-6816-9/22/$31.00 ©2022 IEEE | DOI: 10.1109/MDTS54894.2022.9826976

***Abstract*—Transition fault testing is an important component of modern testing for delay defects. Unfortunately, test pattern sets for delay defects tend to be significantly longer than test pattern sets for static defects. In the past, various approaches have been devised to detect static defects during scan shift to reduce test time and increase defect coverage. In this paper, we propose a DFT (Design-For-Test) enhancement to allow delay defects to be detected by stuck-at test patterns during scan shift as well.**

***Index Terms*—transition faults, Launch on Shift, ATPG, DFT**

I. Introduction

Modern integrated circuits (ICs) must be tested for both static and delay defects. Unfortunately, test sets for delay defects are generally considerably longer than those needed for static defects, and it is more difficult to obtain high fault coverage. This is especially problematic when a circuit must be tested in the field, such as on startup or shut-down, because aging tends to cause circuits to become slower over time—introducing the opportunity for new failures to appear. Thus, a means of testing for delay defects without long test times is needed.

In the past, we have investigated the ability of a MISR (Multiple Input Signature Register) consisting of shadow flip-flops attached to the scan chain to collect data during scan shift that can be used to detect static cell-aware faults. Unlike more traditional fault models (which model faults on the inputs and outputs of gates), cell-aware faults model defects that may occur *within* a logic gate or standard cell. The detection of a cell-aware fault requires the detection of an appropriate stuck-at fault on the output of the gate potentially along with additional conditions that must be satisfied on the gate's inputs. These additional conditions make the cell-aware fault more difficult to detect than the corresponding stuck-at fault—requiring additional patterns for full cell-aware coverage and leading to increased test time. We have previously shown that many cell-aware faults that would otherwise be missed by a stuck-at test set can be detected during scan shift. This reduces the number of patterns above and beyond those needed for stuck-at fault detection that must be applied to fully detect static cell-aware faults.

Unlike static cell-aware faults, transition faults model a large lumped *delay* at a gate input or output. However, like a static cell-aware fault, a transition fault requires the detection of a stuck-at fault at the gate output with an additional condition that must be satisfied—albeit in the previous clock cycle. In circuits with scan chains, transition faults are often detected with the last shift of the chain, where the last shift launches the transition, and the final pattern in the chain is used to detect the stuck-at fault on the gate's output. Intuitively, if a MISR is used to capture scan chain data during shift, it could be possible to detect a transition fault during scan shift as well—provided that the capturing of the data occurs "at speed."

However, using a MISR to detect delay defects during scan shift introduces an additional difficulty. In particular, if the delay from the circuit's logic to the scan cell is very different than the delay from the circuit's logic to the shadow flip-flop that captures data in a MISR, then the results of a delay-based test will be unreliable. Ideally, we would capture the test results in the original scan cells instead because it is the delay to those cells that will actually be seen in functional mode. However, capturing data in those cells will overwrite the pattern being shifted in. Thus, in this paper, we will introduce an alternative structure to allow data to be captured in the original chain on scan shift to detect delay defects while ensuring that the values in the chain at the end of the shift procedure correspond to the pattern that is to be applied. We will show that a high percentage of the detectable transition faults can be fortuitously detected with this approach even when only a static stuck-at test set is used to fill the chains.

II. Previous Work

The transition fault model has been one of the most widely used fault models to obtain high defect coverage. Because exciting a transition fault requires that the correct logic value be assigned to the fault site in a pre-conditioning pattern, the detection of a transition fault is more difficult than the detection of the corresponding stuck-at fault [1]. In some cases, transition faults may be difficult or impossible to detect due to state assignments [2], [3]. In addition, the authors of [4] stated that even when delay faults are testable with structural test patterns, they may not be detectable in the functional mode due to logic or timing constraints or both.

Various Design for Test (DFT) methods have been proposed to improve transition fault coverage. The authors of [5] proposed a method that utilized on-die delay sensing and test point insertion for delay-fault testing. In [3], researchers reduced the number of untestable transition faults by augmenting a controller design with the addition of state transition pairs on invalid states.

This work was supported by NSF CCF1812777 and CCF1814928.

However, other researchers have noted that high delay fault coverage of functionally unsensitizable faults may lead to yield loss and reduced designer productivity if they cause faults that would never fail in functional mode to be targeted and detected during test [6]. To address this, [7] applied test patterns in functional mode while attempting to optimize their tests with hazard-based transitions so that the delay faults detected would not be redundant under real operating conditions.

Some additional algorithmic approaches for transition test generation, simulation, and/or analysis include [8] and [9]. [8] provided several fault collapsing procedures for transition faults based on dominance relations. An efficient method was invented by the authors of [9] for transition delay fault coverage based on a multi-valued algebra, critical path tracing and deductive fault simulation.

In addition to traditional transition fault models, small delay faults and cell transition faults have been proposed to improve delay coverage. The authors of [10] indicated that traditional fault models are insufficient for guaranteeing small delay defect detection. Similarly, in [11] an enhanced transition fault model was proposed to improve CMOS transition stuck-open fault coverage for stuck-open fault detection inside the CMOS cells.

Fault detection during scan shift is another DFT method that can help detect faults (e.g. [12], [13], [14]). More recently, [15] utilized shifting of stuck-at patterns with enhanced scan chains to fortuitously obtain high cell-aware fault coverage while only applying a stuck-at test set. With the same DFT circuitry, [16] studied the n-detection of stuck-at faults. Even with low hardware overhead, most of the least detected stuck-at faults were detected many more times without changing the pattern set. The authors of [17] also targeted detection during scan shift with test point insertion. However, these methods only considered static faults. Thus, in this paper, novel DFT circuitry is proposed to allow fortuitous transition fault detections to occur when a stuck-at test set is shifted through the chains.

III. Transition Fault Detection during Scan Shift

The majority of the test time in scan-based circuits is devoted to scan shift instead of capture—especially when chains are long (e.g. [15], [16].) As a result, only a very small percentage of all testing clock cycles are used to capture defective behavior in such designs. To allow these shift cycles to be used for defect detection, modifications to the traditional scan chain have been proposed. In particular, a scan chain enhancing method that used a MISR structure to capture test data during scan shift achieved high cell-aware coverage for static defects in [15]. Similarly, high *n*-detect coverage of stuck-at faults was explored in [16].

These approaches were highly successful because they allow the number of "effective" patterns applied to increase by a factor approximately equal to the number of scan shift cycles if test data is captured on every shift cycle. Thus, even a relatively short pattern set can be supplemented by pseudo-random "intermediate" patterns that are orders of magnitude larger than the original test set, and each of these intermediate patterns can potentially detect additional faults and defects.

Due to the success of these previously studied approaches, it is reasonable to consider applying "intermediate" scan shift patterns to transition fault detection as well. However, an important complication arises due to the fact that transition faults are modeling extra delays. Technically, transition faults are assumed to model large lumped delay defects, and as a result the path taken through the circuit (and the slack of the associated path) is assumed to be inconsequential. However, in reality, the slack of the path taken often does matter when detecting extra delays in real circuits, and thus ideally delay defects should be detected in the *functional* flip-flops—even during scan shift—instead of *shadow* flip-flops. At the same time, it is necessary to allow the desired ATPG pattern to be present in the chain at the end of scan shift and to shift out the actual values captured by the original ATPG-generated test pattern.

Because [15] and [16] aimed to detect static defects only, capturing test data in *shadow* flip-flops collected into a MISR, instead of in the original *functional* flip-flops in the scan chain, did not affect the detection of those defects. Because this is no longer true in the case of added delays, a modification to the intermediate-pattern based DFT structures proposed in prior work is needed for the approach to be extended to delay faults.

IV. Circuitry Structure for Transition Fault Fortuitous Detection with Intermediate Patterns

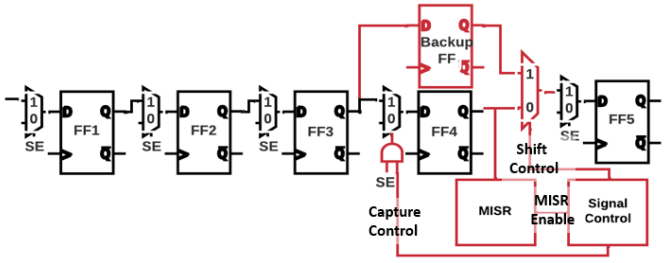

Fig. 1. Selected scan flip-flop in a scan chain is "Backed up" with a bypass flip-flop so that transition faults can be captured in the original scan chain without losing the "normal" shifted or captured value.

Figure 1 shows DFT circuitry designed for transition fault detection with intermediate shift patterns. In this figure, $FF1$ to $FF5$, along with their preceding multiplexers, are shown in black and form five MUX-D scan flip-flops. These scan flip-flops form the original scan chain.

In this figure, $FF4$ is selected to capture test data during scan shift while the other flip-flops will only capture test data once the full ATPG-generated pattern has been shifted in. To accomplish this additional capturing of test data in the original scan flip-flop while preserving the ability to successfully shift in each ATPG pattern, the circuit elements are shown in red are added to the original chain. In particular, a *Backup FF* is added in parallel with $FF4$ to backup the values that are supposed to be shifted into $FF4$. This FF and the other added elements in red work with the rest of the DFT logic as described below.

A. Description of the Added Circuitry

When we wish to apply a launch-on-shift (LoS) transition test using an intermediate pattern during scan shift, we need to allow $FF4$ to capture data from the circuit's logic while the other flip-flops on the scan chain capture data from the previous flip-flop in the chain. At the same time, the *Backup FF* should capture the data that would have been shifted into $FF4$.

Thus, three control signals from another logic block, *Signal Control*, are added (Figure 1). Note that only one *Signal Control* module is needed for all of the scan chain flip-flops that are selected to capture test data with intermediate shift patterns—even when those flip-flops are in different scan chains. *Capture Control* is used to determine whether SE will enable the shifting of data into $FF4$. When *Capture Control* is equal to 1, the SE signal will determine whether $FF4$ captures data from the circuit's functional logic or from the previous flip-flop in the chain. When *Capture Control* is equal to 0, the capture of test data into $FF4$ from the circuit's logic with an intermediate shift pattern is enabled even as the other flip-flops in the chain continue to shift normally.

After test data is captured into $FF4$ with an intermediate shift pattern, two things must happen. 1) The value in $FF4$ must be fed into a MISR so that a single signature can be used to identify whether any of the values captured by the intermediate shift patterns were incorrect. 2) The value which would have been shifted into $FF4$ under normal shifting conditions must be shifted into $FF5$ instead of the value that was captured in $FF4$ from the circuit's logic.

Thus, to shift the correct value into $FF5$, a multiplexer after $FF4$ is used to control which value is shifted into $FF5$. The select input of this multiplexer is set by *Shift Control* from the *Signal Control* logic block. Note that the *MISR Enable* signal is generated by the *Signal Control* module as well to ensure that the MISR value will be updated only when a test result from an intermediate shift pattern is in the corresponding scan flip-flop (here $FF4$)—as opposed to all clock cycles.

B. Example of Operation with the Proposed Enhancement

Figure 2 shows an example of the proposed structure in operation. The example begins as data is captured in all of the scan flip-flops when a regular ATPG test pattern is applied. In this example, an intermediate shift pattern will be used to detect defects, and $FF4$ will capture test data after two bits of the next ATPG pattern are shifted in.

1) Cycle of normal capture: The *SE* signal is set to 0, and all flip-flops on the scan chain capture values from the circuit's logic. These values are specified as $C1$ to $C5$, depending on the flip-flop in which the capture occurs.
2) First cycle after normal capture: The first bit of the next ATPG pattern is shifted in with the SE signal set to 1. The same value $C3$ (from $FF3$) is shifted into $FF4$ and into the *Backup FF*. *Capture Control* is set to 1; *Shift Control* and *MISR Enable* are set to 0.
3) Second cycle after normal capture: The second bit of the next ATPG pattern is shifted into the chain. The value $C2$ is shifted into $FF4$ and the *Backup FF*. The values of *SE*, *Capture Control*, *Shift Control* and *MISR Enable* remain the same as in last clock cycle. This cycle launches the transitions that will be used for detecting transition faults with the intermediate shift pattern.
4) Third cycle after normal capture: The third bit of the next ATPG pattern is shifted into the scan chain. The previously captured value $C1$ is saved in the *Backup FF*. Because we want to capture test data ($M1$) from the circuit logic in response to the transitions launched on the last shift cycle in $FF4$, the value of *Capture Control* is set to 0. To match the delay of the circuit in normal operation, the time between the start of the "2nd cycle after capture" and the start of the "3rd cycle after capture" should be equal to one clock period. Thus, the value of *Capture Control* must transition from 1 to 0 in less than 1 clock. *Shift Control* and *MISR Enable* remain unchanged.
5) Fourth cycle after normal capture: The fourth bit of the next ATPG pattern is shifted into the scan chain. The MISR is enabled by setting *MISR Enable* to 1 so that the value captured in $FF4$ can be used to update the signature in the MISR. *Shift Control* is set to 1, so that the value $C1$ from the *Backup FF* is shifted into $FF5$. *Capture Control* is set to 1 so that both $FF4$ and the *Backup FF* will receive the first bit of the new ATPG test pattern: $S1$.
6) Fifth cycle after normal capture: The last bit of the next ATPG pattern is shifted in. The value shifted into $FF5$ is received from $FF4$ with *Shift Control* as 0. The signature in the MISR does not update as *MISR Enable* is disabled (set to 0). Both $FF4$ and the *Backup FF* capture the value $S2$ of the ATPG test pattern. At this point, the entire ATPG pattern has been shifted in, and the process can begin again.

V. Experimental Procedure and Results

To determine the ability of the proposed approach to fortuitously detect transition faults with stuck-at test patterns, the following procedure was followed:

1) **Stuck-at Fault ATPG Pattern Generation**: Stuck-at ATPG patterns are generated for each circuit. In each case, an on-chip decompressor was used to feed the circuit's scan chains. The values loaded in the scan chains, primary input (PI) values, primary output (PO) values, and good circuit simulation capture values for each ATPG pattern are obtained.
2) **Intermediate Pattern Generation**: Stuck-at intermediate patterns are generated for each shift cycle. Each intermediate pattern is a combination of the ATPG values being shifted in and the good circuit simulation values being shifted out from the previous pattern.
3) **Stuck-at ATPG Pattern Transition Fault Detection**: The last shift of a regular stuck-at ATPG pattern into the scan chain can be used to detect transition faults without requiring the backing up of any flip-flop data or

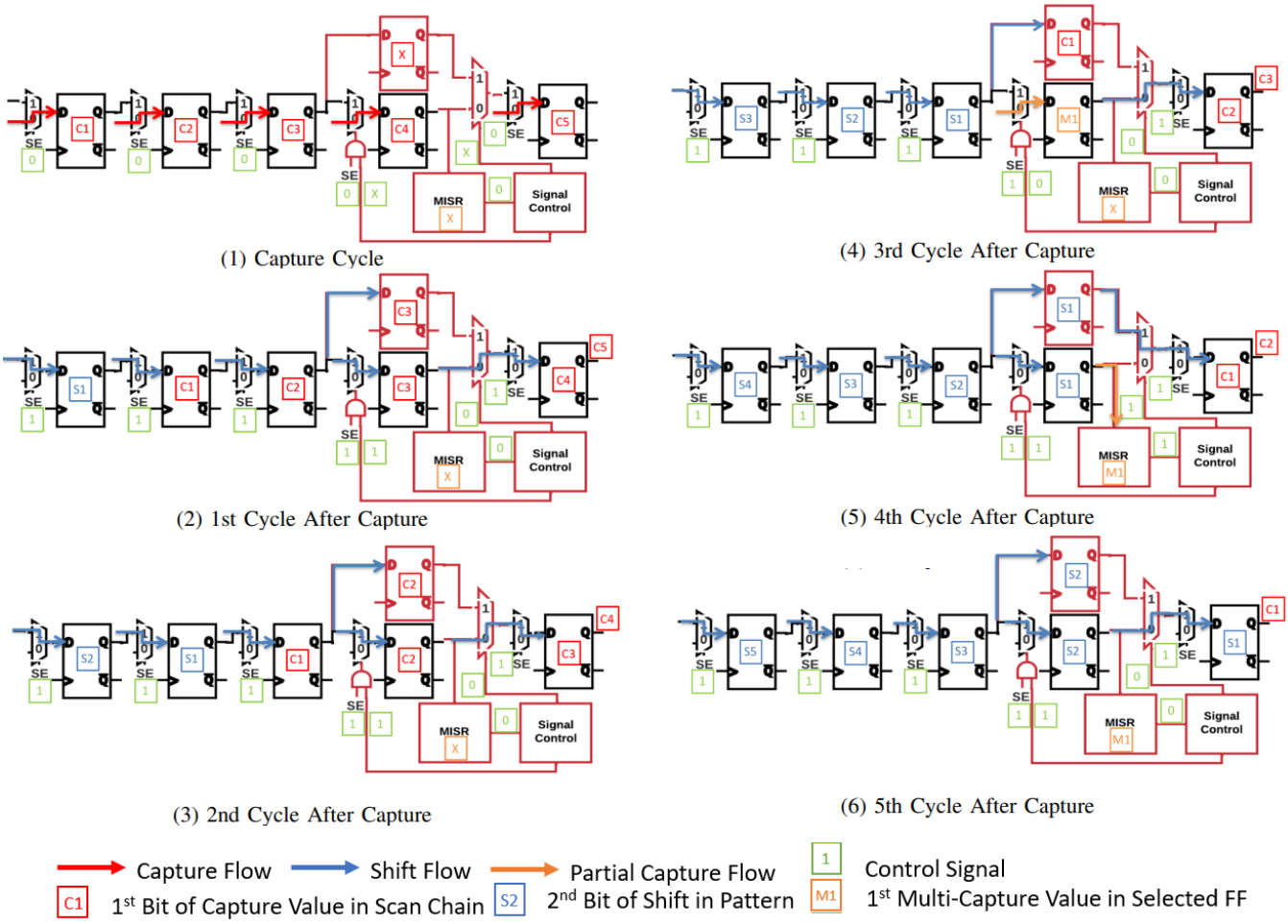

Fig. 2. Transition Fault Scan Chain Enhancing DFT Structure Work Flow

the enabling of the added MISR. In particular, because the last shift of the stuck-at pattern naturally launches transitions into the circuit, transition faults can be detected along with stuck-at faults by capturing data at the circuit's outputs and scan flip-flops as is normally done for a stuck-at test—provided that the capturing can be done at-speed.

Using the final stuck-at test pattern for transition detection is advantageous because stuck-at fault coverage is usually very high, and even stuck-at faults that are detected only once can potentially contribute to fortuitous transition fault coverage. However, some transition faults will still remain undetected; it is these undetected faults that we will target for detection by intermediate patterns on scan shift. Flip-flops will be selected for the additional Back-up and MISR circuitry to improve the detection of these missed faults.

4) **Selecting Shifting Clock Cycles to Perform Extra Captures**: Because we are capturing test data in the original flip-flops when an intermediate shift pattern is applied (e.g. value *M1* is captured in the regular flip-flop in Figure 2 part 5), we must determine those values and include them in our intermediate pattern fault coverage analysis if at-speed capture for intermediate patterns is performed on every shifting clock cycle. This will make further analysis more complex. To avoid this complexity, in this paper, we restrict intermediate pattern fault coverage analysis to only those clock cycles in which values such as *M1* do not appear in the scan chain. Figure 3 shows one example of intermediate pattern sampling to avoid this complexity. Each numbered square corresponds to a shift clock cycle for a chain of length 10. In this example, ten clock cycles are needed to fully shift a stuck-at ATPG test pattern into the chain.

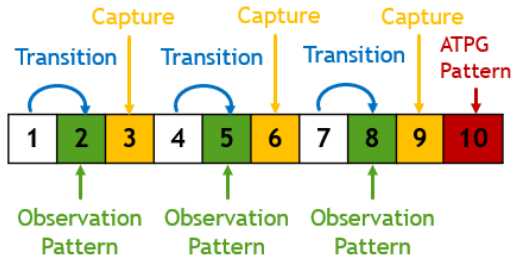

Fig. 3. Intermediate Pattern Sampling Example (Chain Length 10)

In our experiments, the first intermediate pattern pair used for fortuitous detection of transition faults corresponds to intermediate patterns 1 and 2 (i.e. shifting

clock cycles 1 and 2). The transitions in the circuit logic are launched by the second shifting clock cycle, and the intermediate pattern in the chain after the second shift is the pattern that is used to observe the stuck-at faults that correspond to each transition fault. Then, the "extra" capture of the test result occurs with shift cycle 3—where the selected flip-flops capture the circuit's test results, and the backup flip-flops preserve the values that would have been shifted into those selected flip-flops under normal conditions. Thus, the time between the launching of transitions with shift cycle two and the capturing of test results with shift cycle three must be "at speed."

Because the data captured in the selected flip-flops on shift cycle 3 correspond to data generated by the circuit logic, it is not available *a priori* from simple analysis of the stuck-at ATPG patterns in the test set. Instead, good circuit simulation must be done to obtain those values. While that data must be obtained eventually to determine the appropriate MISR signature, if we have not yet finalized which flip-flops will be selected for the extra captures, then there are many possible versions of shift cycle 3 that could be used and analyzed for estimating transition fault coverage. To avoid this complexity, the next intermediate pattern pair used for fortuitous detection of transition faults corresponds to shift cycles 4 and 5.

If we continue in this way, the last intermediate pattern pair would correspond to shift cycles 7 and 8, which would launch the next set of transitions. Intermediate pattern 8 would correspond to the observation pattern, and clock cycle 9 would correspond to the next capture of test results and backing up of test data. Unfortunately, if we do this, then we cannot use intermediate pattern pairs 9 and 10 to detect transition faults for the "Stuck-at ATPG Pattern Detection" from Step (3) because some of the values in the chain on cycle 9 come from the circuit logic instead of only the shifted data.

As a result, because "Stuck-at ATPG Pattern Detection" from Step (3) is usually more valuable for fortuitously detecting transition faults, in our experiments we do *not* use intermediate patterns 7 and 8 for fortuitous transition fault detection. (Ending on a true stuck-at test pattern is more beneficial because the ATPG tool has tried to guarantee at least minimal detections of even the most difficult-to-detect stuck-at faults with the final stuck-at fault test patterns.)

5) **Flip-Flop Selection for Backup and Extra Capture**: To determine which flip-flops should be selected for Backup and extra captures, stuck-at fault coverage data from the observation patterns (e.g. 2, 5, and 10 in Figure 3) are used. Flip-flops are sorted in the order of detected stuck-at faults from highest to lowest, and the top flip-flop is selected. Stuck-at faults detected by that flip-flop are removed from consideration, and the process repeats. The same sorting and selecting steps are repeated until all faults have been detected or all flip-flops have been selected. At the end, the selected flip-flops form a flip-flop list that could potentially detect detected transition faults. Because transition launching conditions are not used in the analysis, the selection procedure is slightly more general than would otherwise be the case. (Future work will look at the differences in flip-flop selections that may occur with other variations of this procedure.)
6) **Transition Fault Detection with Selected FFs**: Transition fault simulation is performed with the selected flip-flops to obtain an accurate value for transition fault coverage. This step also allows for analyzing fault-coverage at different hardware overhead allowances when fewer flip-flops are selected.

Using these steps we performed experiments on five circuits obtained from Opencores.org [18]. Table I shows the circuit characteristics as well as the intermediate patterns and sampled intermediate patterns generated by our method.

TABLE I
CHARACTERISTICS OF CIRCUITS

Circuit	# of Flip -Flops	# of Transition Faults	# of SA ATPG Patterns	# of Intermediate Patterns	# of Sampled Intermediate Patterns
Quadratic	120	8166	36	1044	324
Des56	193	13788	119	2856	833
Fm_rec	501	19888	406	10962	3248
Colorconv	584	38518	98	3136	980
Fpu	5231	297358	538	17216	5380

Figure 4 shows fortuitous transition fault coverage with stuck-at ATPG patterns and intermediate patterns. For each circuit, the first bar (green) shows the LoS transition fault coverage obtained by a commercial ATPG tool for a dedicated LoS test set. The second bar (yellow) shows the fortuitous fault coverage achieved with the stuck-at ATPG and intermediate patterns when all flip-flops can capture test data on intermediate patterns. Most of the transition faults are covered. Finally the third bar (blue) shows the fault coverage when the flip-flop selection algorithm is used to shadow only a sub-set of flip-flops into the MISR. Even though stuck-at fault detection was used for flip-flop selection, the transition coverage remains almost the same.

TABLE II
PERCENT OF FLIP-FLOPS REQUIRED FOR TRANSITION FAULTS DETECTED BY STUCK-AT INTERMEDIATE PATTERNS

	Quadratic	Des56	Fm_rec	Colorconv	Fpu
Selected FF (%)	19.17	98.45	23.15	68.49	82.62

Table II shows the percent of flip-flops shadowed in the MISR to maximize fortuitous detection of transition faults by stuck-at and intermediate patterns. In some cases, these numbers could be further reduced if faults that could be detected by a chain test were not included.

We also ran experiments to see how putting an upper bound on the percentage of flip-flops selected impacted fault-

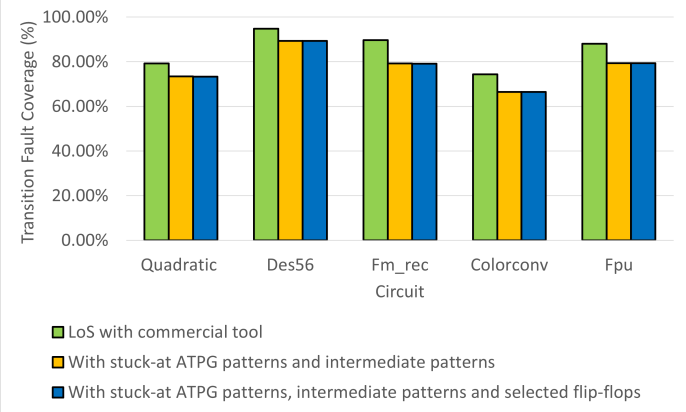

Fig. 4. Transition Fault Coverage with Stuck-at ATPG Patterns, Intermediate Patterns and Selected Flip-flops

coverage. Using a greedy algorithm (as in Step 5), we picked flip-flops up to a certain budget by prioritizing the flip-flops that detect the most faults first. Figure 5 shows the corresponding transition fault coverage. From the figure, we can see that even with a low budget of 1%, significant fault-coverage is possible. For circuit Quadratic, we can get the maximum fault-coverage with only 19% of the flip-flops. Thus, Quadratic was excluded from the 20% and 40% experiments in Figure 5. Similarly circuit Fm_rec also gets to maximum fault-coverage with just 23% of the flip-flops selected. The column for that circuit is excluded from the 40% experiment.

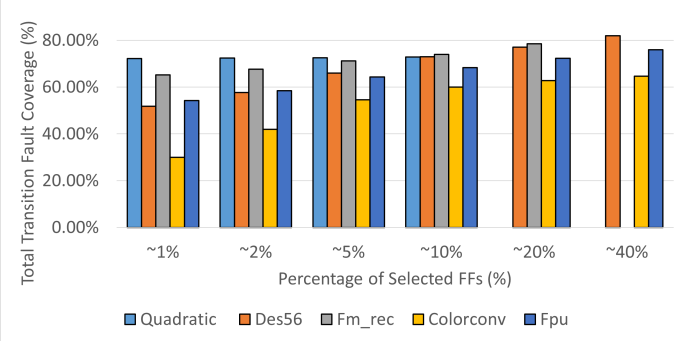

Fig. 5. Transition Fault Coverage with varying overhead of flip-flop

VI. Conclusion and Future Work

Significant transition fault coverage can be fortuitously achieved using a combination of ATPG stuck-at patterns, intermediate patterns, and selective flip-flop shadowing in the MISR. Even higher fault coverage values may be obtained with less overhead if the possible detection of faults during the chain test were considered (11% to 17% of faults may be detectable by chain test for the circuits studied). Future work will explore the impact of including the chain test as well as the impact of adding LoS ATPG top-off patterns to detect any transition faults that still remain undetected.

References

[1] Alfred L Crouch, "Design-for-test for Digital IC's and Embedded Core Systems," 1999, The Rosen Publishing Group

[2] M. Yoshimura, Y. Takeuchi, H. Yamazaki and T. Hosokawa, "A State Assignment Method to Improve Transition Fault Coverage for Controllers," 2019 IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT), 2019, pp. 1-4, doi: 10.1109/DFT.2019.8875322.

[3] Y. Takeuchi, T. Hosokawa, H. Yamazaki and M. Yoshimura, "A Controller Augmentation Method to Improve Transition Fault Coverage for RTL Data-Paths," 2019 IEEE 25th International Symposium on On-Line Testing and Robust System Design (IOLTS), 2019, pp. 293-298, doi: 10.1109/IOLTS.2019.8854445.

[4] A. Krstic, Jing-Jia Liou, Kwang-Ting Cheng and L. -. Wang, "On structural vs. functional testing for delay faults," Fourth International Symposium on Quality Electronic Design, 2003. Proceedings., 2003, pp. 438-441, doi: 10.1109/ISQED.2003.1194772.

[5] S. Ghosh, S. Bhunia, A. Raychowdhury and K. Roy, "A Novel Delay Fault Testing Methodology Using Low-Overhead Built-In Delay Sensor," in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 25, no. 12, pp. 2934-2943, Dec. 2006, doi: 10.1109/TCAD.2006.882523.

[6] J. Rearick, "Too much delay fault coverage is a bad thing," Proceedings International Test Conference 2001 (Cat. No.01CH37260), 2001, pp. 624-633, doi: 10.1109/TEST.2001.966682.

[7] I. Pomeranz and S. M. Reddy, "Hazard-Based Detection Conditions for Improved Transition Fault Coverage of Functional Test Sequences," 2009 24th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems, 2009, pp. 358-366, doi: 10.1109/DFT.2009.11.

[8] I. Pomeranz and S. M. Reddy, "Fault Collapsing for Transition Faults Using Extended Transition Faults," Eleventh IEEE European Test Symposium (ETS'06), 2006, pp. 173-178, doi: 10.1109/ETS.2006.22.

[9] A. Bosio, P. Girard, S. Pravossoudovich, P. Bernardi and M. S. Reorda, "An efficient fault simulation technique for transition faults in non-scan sequential circuits," 2009 12th International Symposium on Design and Diagnostics of Electronic Circuits & Systems, 2009, pp. 50-55, doi: 10.1109/DDECS.2009.5012098.

[10] Y. Sato, S. Hamada, T. Maeda, A. Takatori, Y. Nozuyama and S. Kajihara, "Invisible delay quality - SDQM model lights up what could not be seen," IEEE International Conference on Test, 2005., 2005, pp. 9 pp.-1210, doi: 10.1109/TEST.2005.1584088.

[11] X. Lin, W. Cheng and J. Rajski, "On Improving Transition Test Set Quality to Detect CMOS Transistor Stuck-Open Faults," 2015 IEEE 24th Asian Test Symposium (ATS), 2015, pp. 97-102, doi: 10.1109/ATS.2015.24.

[12] A. S. Abu-Issa, I. K. Tumar and W. T. Ghanem, "SR-TPG: A low transition test pattern generator for test-per-clock and test-per-scan BIST," 2015 10th International Design & Test Symposium (IDT), 2015, pp. 124-128, doi: 10.1109/IDT.2015.7396748.

[13] G. Mrugalski, J. Rajski, J. Solecki, J. Tyszer and C. Wang, "TestExpress - New Time-Effective Scan-Based Deterministic Test Paradigm," 2015 IEEE 24th Asian Test Symposium (ATS), 2015, pp. 19-24, doi: 10.1109/ATS.2015.11.

[14] O. Novak and H. Nosek, "Test-per-clock testing of the circuits with scan," Proceedings Seventh International On-Line Testing Workshop, 2001, pp. 90-92, doi: 10.1109/OLT.2001.937825.

[15] Fanchen Zhang, Daphne Hwong, Yi Sun, Allison Garcia, Soha Alhelaly, Geoff Shofner, LeRoy Winemberg and Jennifer Dworak, "Putting Wasted Clock Cycles to Use: Enhancing Fortuitous Cell-Aware Fault Detection with Scan Shift Capture," in Test Conference (ITC), 2017 IEEE International, pp. 1–9, 2017, ISSN 2378-2250

[16] H. Jiang, F. Zhang, Y. Sun and J. Dworak, "One more time! Increasing fault detection with scan shift capture," 2018 IEEE 27th North Atlantic Test Workshop (NATW), Essex, VT, 2018, pp. 1-7. doi: 10.1109/NATW.2018.8388862

[17] N. Mukherjee et al., "Test Time and Area Optimized BrST Scheme for Automotive ICs," 2019 IEEE International Test Conference (ITC), 2019, pp. 1-10, doi: 10.1109/ITC44170.2019.9000133.

[18] http://opencores.org/