Special Session: Delay Fault Testing - Present and Future

Jubayer Mahmod, Spencer Millican, Ujjwal Guin, and Vishwani Agrawal
Department of Electrical and Computer Engineering
Auburn University, AL 36849
{jubayer, millican, ujjwal.guin, agrawvd@auburn.edu}

***Abstract*—This article presents a brief survey of digital delay fault testing, which lists 100+ references on fault models, simulators, ATPG, DFT, and tools. Continuing studies are needed in this maturing field for new technologies, signal integrity, process variations, faster than critical path operation, asynchronous circuits, counterfeit ICs, and hardware Trojans. This information is compiled to provide direction to students, practicing engineers, and researchers alike.**

***Index Terms*—Survey, delay fault testing, static timing analysis, small delay defect, hardware Trojan, recycled IC, timing-aware ATPG.**

I. Introduction

In modern digital integrated circuits (ICs), narrow timing margins result in delay defects that are difficult to detect. Such defects are becoming common due to shrinking IC feature sizes, increasing process variations, and faster operating frequencies [1]–[4]. Although not activated at low speeds, delay defects can cause timing failures at the rated speed. Delay tests detect timing failure causing defects to ensure that a circuit meets the desired timing specifications.

Unlike stuck-at logical faults, a pair of vectors is needed to test a delay fault; the first, an initialization vector v_1, activates the fault, and the second, called the launch vector v_2, creates a transition at the fault location and propagates the fault effect to an observation point. An illustrative example of a delay test is shown in Figure 1. In this example, the circuit-under-test (CUT) functionally operates with a rated clock of period T_c. During test, two separate clocks with period T_s drive input and output latches. The two test clocks are skewed by the rated clock period, T_c. Vectors v_1 and v_2 are applied at times t_0 and t_1, respectively. Even though the circuit is operating at slow speed during test ($T_s > T_c$), the response is captured at the rated clock frequency, T_c, and compared against a fault-free circuit to detect delay-causing faults in the CUT.

Delay fault testing is an active area of research due to inherent difficulties in testing them. Challenges are inflicted by process variations, the prospect of improvising new delay simulation technologies, and applications to hardware security. This survey provides an introduction to delay fault testing with recent progress, challenges, and emerging prospects in the field.

Next, Section II, is an encompassing survey of delay fault testing and its applications. Section III lists future research directions and Section IV concludes this article.

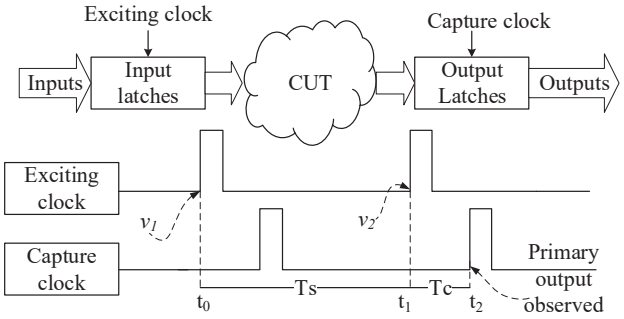

Figure 1: A typical delay test configuration for a scan circuit.

II. A Survey of Delay Testing

The well-known methods of testing a digital circuit with tests for stuck-at faults [1] imply logical (or functional) correctness. The question whether the circuit can perform fast enough to satisfy its timing specifications is answered by delay fault testing. Timing specifications are based on requirements of the system where the circuit is to be used. The timing is considered through design and manufacture in three stages.

First, during logic design, exact delays of gates and interconnects are not available and must be approximated from the knowledge of technology (nominal gate delays, wire capacitance, etc.) and circuit topology (fanouts, etc.). The estimated quantities, generally referred to as wire-load delays [5], are used to determine the delays of input to output paths, and to re-synthesize the logic to minimize the delay of longest paths known as critical paths. In the second stage, after the physical design and layout, actual delays are computed from sizes and electrical behavior of transistors and interconnects. A more accurate analysis now verifies the timing performance of the design. In the third stage, delay testing is applied to each fabricated circuit using tests that aim to detect certain modeled delay faults.

A. Terms and definitions

Topics in delay testing include *delay characterization*, *timing analysis and critical path*, *delay fault models*, *fault simulation*, *design for testability* (DFT), and *automatic test pattern generation* (ATPG). Delay fault models represent the logical and/or timing effects of a delay defect in a circuit under test (CUT). The modeled faults are targeted by ATPG algorithms to generate tests. Timing-aware ATPG (TA-ATPG)

978-1-7281-1170-4/19/$31.00 ©2019 IEEE

algorithms are designed to produce test vectors using circuit's timing information. To improve a delay fault tests quality (i.e., delay fault coverage, test generation, or computation time), one may need to modify a CUT with DFT hardware.

B. Delay characterization

Delay characterization occurs during the design phases, namely, logic and physical design. Gates and interconnects contribute delays in a digital circuit. *Switching or inertial delay* is the interval between input change and output change for a gate. It depends on input capacitance, device (transistor) characteristics and output capacitance of the gate. To be exact it also depends on input rise or fall times and states of other inputs (second-order effects). For simplicity, fixed rise and fall delays (a single fixed delay or a min-max delay range) are determined for each gate output. In addition, a signals experiences *propagation delay* as it traverses through interconnects. This is the time a transition takes to travel between gates. It depends on transmission line effects (distributed *R, L, C* parameters, length, and loading) of routing paths. These are often modeled as lumped delays for gate inputs.

The accuracy of delay characterization depends upon the available detail in various design phases. For example, before physical design, only the circuit *netlist* is available. Lumped delays, sometimes referred to as wire-load delays [5],are assigned to each gate output based on fanout the gate drives. After physical design, more accurate delays can be calculated for specific transistor sizes and interconnect geometries. Approximate delays of interconnects are often determined from the Elmore delay formula [6]. Advanced formulas [7], [8] take the signal slope into account to compute more accurate delays. Most accurate values are obtained by detailed circuit analysis [9].

C. Static timing analysis (STA)

PERT (Program Evaluation and Review Technique) is used in industry for project management. The milestones of the project are represented as vertices on a graph. A directed edge between vertices represents a task, edge weight being the time to execute the task. A "start" vertex has zero indegree and a "finish" vertex has zero outdegree. The longest path from *start* to *finish*, called the *critical path*, is determined using a graph search algorithm. The sum of weights of all edges on the critical path provides the project duration. PERT has been applied to logic design [10] and timing analysis [11] of digital circuits. The netlist of a combinational circuit is a graph where a vertex is gate and a directed edge between a vertex-pair is an interconnect with propagation delay as the edge weight. All primary inputs (PI) are collapsed into a single "start" vertex and all primary outputs are collapsed into a "finish" vertex. Static timing analysis (STA) is a graph search algorithm that finds paths between the *start* and *finish* vertices and determines path delays.

Static timing analysis (STA) [10]–[13] is a useful tool for timing verification and optimization of digital circuits. Its complexity is significantly lower than alternative tools using timing [1] or circuit [9] simulation. STA does not use any input stimuli like a simulator, but exhaustively analyzes the entire circuit topology (netlist graph) using modeled gate and interconnect delays to find critical paths [14], [15]. Graph-based linear time analysis makes STAs highly efficient. STA is a complete timing verification method that can guarantee the operation of a CUT at the rated speed. The most important aspect of STA is that it can analyze the entire design at once and can check all possible timing related violations. Its one downside is that it cannot identify *false paths* [16], [17], which often leads to overdesign. False paths cannot propagate a transition and hence their delays do not affect the circuit timing. In fact, a circuit can be optimized in area and delay by re-synthesis to remove false paths [16], [17].

D. Delay fault models

ATPG tools must handle large designs while using minimal computational resources and therefore require efficient fault models to represent delay defects. Here, several delay fault models which are used in industry and research are explored.

Path delay fault (PDF) model: Under the PDF model, a circuit is faulty if the delay of any circuit path exceeds the specified clock/input vector period [18]. A path starts at a primary input (i.e., circuit input) or flip-flop output and ends at a primary output (i.e., circuit output) or a flip-flop input. A PDF is capable of modeling gross delay defects (GDDs) as well as distributed delays along a path. PDF testability has been correlated to single stuck-at faults [19], but an issue with PDFs is that their number is exponentially proportional to the number of gates in a circuit [20].

Transition delay fault (TDF) model: TDFs model the slowing of a signal transition as if the transition will never reach an observation point within the specified clock period [21]. The two kinds of TDF are *slow-to-rise* and *slow-to-fall*, which are equivalent to a stuck-at-0 and stuck-at-1 fault, respectively, after the fault's initial condition is excited. The number of TDFs is twice the number of lines in the circuit and therefore this model is more widely adopted than the PDF model. ATPGs developed for stuck-at faults can be adopted for TDFs [22].

Line delay fault (LDF) model: This model places rising or falling delay defect on a gate inputs and outputs [23]. The number of LDFs in a circuit is twice the number of lines (slow-to-rise and slow-to-fall on each line). To test an LDF, a transition on the line must take the longest sensitizable path through the line to an observation point. Sensitizing the longest path through the line is advantageous because it detects small delay defects (SDDs) on the target line [24]. However, LDFs can fail to detect some SDDs since only one propagation path is considered for each line [25]. In addition to robustly testable paths, there could exist non-robust tests for some longer paths through the target line (see Section II-G).

Gate delay fault (GDF) model: GDFs manifest at gates (as opposed to lines) [26], [27]. It models a lumped delay

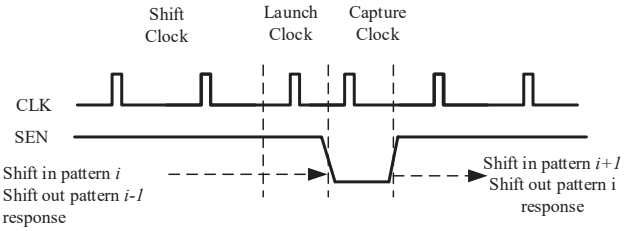

Figure 2: Launch-off-shift (LOS) delay test waveform [39].

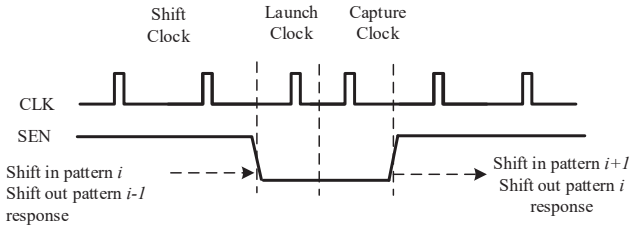

Figure 3: Launch-off-capture (LOC) delay test waveform [39].

defect at the gate output. There is a magnitude (or size) of the delay (e.g., picoseconds) defect that is associated with the GDF. This size should be large enough to be detectable when all other gates are within the nominal range. The number of GDFs is linearly proportional to the number of gates in the circuit where each gate can have one or more faults of different magnitudes. A known limitation of this model is that it can fail to represent distributed delay defects [27]. Although some algorithms have been reported [28], GDFs have not found applications in digital testing. Perhaps one reason is that their detection requires the circuit output to be observed at precise time instant, which would add complexity to the test program.

Segment delay fault (SDF) model: SDFs combine properties of PDFs and TDFs using the topological location of defects. SDFs have been observed to be an accurate model for hardware defects [29]. SDFs assume a defect will affect several gates and paths in a local region called a *segment*. The segment length (L) ranges from 1 to L_{max}, where L_{max} is the number of gates in the longest path of CUT. For $L = 1$, a SDF becomes a TDF, while $L = L_{max}$ makes SDF a PDF.

Small delay defect (SDD) model: SDDs introduce small amounts of extra delays to the design [30]. These delays are smaller than circuit's rated clock period. Although this delay is very small, the accumulation of several delay defects on a path can cause failures [31]. Besides, SDDs have been observed to be correlated with early life failures of circuits [32], [33], and hence should not be ignored either in the test phase [34]–[36] or in the diagnosis phase [37], [38].

E. Test application schemes and DFT

Test pattern generation and test coverage are influenced by available scan-chains and how these scan-chains are controlled [1]. Creating tests for sequential circuits without scan increases the complexity of ATPG significantly since the application of vectors and the observation of faults will require multiple clock cycles [2]. Arbitrary vector pairs cannot be applied in a non-scan or even in standard scan design. A vector pair $< v_1, v_2 >$ becomes $< s_1 + i_1, s_2 + i_2 >$ for non-scan circuits, where i_1 and i_2 are the inputs of a CUT at state s_1 and s_2, and one must calculate $f(s_1, i_1)$ to produce the next desired state s_2. Depending on how transitions are launched and captured, testing can be categorized into three methods [39] (see Figures 2 and 3):

- *Launch-off-shift* (LOS) or *skewed load*: Faults are excited on the last cycle of scan-in, restricting v_2 to be a 1-bit shift of v_1. A time-critical scan-enable (SEN) signal captures the transition at the capture clock edge [40].
- *Launch-off-capture* (LOC) or *broadside*: The circuit is initialized with vector v_1 at the end of the scan-in. The vector v_2 is the functional response of v_1. This makes the vector pairs restricted since v_2 must be a function of v_1. Consequently, test coverage is lower compared to LOS. However, LOC does not require the SEN input to be applied at the rated circuit speed [41].
- *Enhanced scan*: Arbitrary vector pairs are applied using hold latches between scan flip-flops [1], [2], [42].

Each scheme has its advantages and disadvantages. ATPG is easier and fault coverage higher for enhanced scan [43] since the vector pairs, $< v_1, v_2 >$, do not need to be correlated. However, hold latches require higher DFT area overhead and an additional hold signal is required to hold transition latches. Hence, circuit performance is negatively affected when using enhanced scan.

Scan-based delay testing can have yield loss issues. A chip may fail delay test because functionally unsensitizeable faults are activated during test, although they can never be active during the normal operation of the circuit [17], [44]–[46].

Much research effort has been spent to improve the quality of delay tests using other DFT circuit modifications. Since many designs suffer from low PDF coverage, many delay fault improvement schemes have been developed for testing PDF. Chakrabarti *et al.* [47] proposed a synthesizing method for symmetric Boolean functions to achieve complete *robust* PDF testability. As explained in Section II-G, a robust PDF test detects the fault on a target path regardless of delays on other paths, while a non-robust PDF test may fail to detect the fault if other paths are faulty [1]. Le *et al.* [48] modied random-access scan architecture to reduce test time for PDF. Pomeranz and Reddy [49] used multiplexers and Siebert *et al.* [50] developed a single gate based DFT method to improve the PDF coverage. Although these methods increase PDF coverage by making all paths testable, additional control input pins make these approaches unattractive. A Reduced-Ordered Binary Decision Diagram (ROBDD) based scheme has been proposed to eliminate input pin requirements and improve PDF testability [4]. Multiple independent scan-enable (SEN) signals have also been used for clustered groups of flip-flops to improve TDF coverage [51].

F. Delay fault simulation

Fault simulation is essential for test pattern generation and test quality analysis. Delay faults can be simulated using *logic-based*, *probabilistic*, or *static analysis* methods. Logic-based simulation performs Boolean operations without timing information. Logic simulators are comparatively fast [21] and use the GDF or TDF model and model signal transitions without considering delay magnitudes [52]. Graph-based methods that rely on logic simulation and path counting and be used to simulate PDFs in feasible time [53]. Probabilistic simulation estimates fault detection using statistical delay distributions [30], [54]. Static analysis uses corners cases (i.e., best and worst-case timing analysis [52], [55]) to calculate delay. Ideally, all timing events will not be modeled to decrease simulation run-time. More run-time is required to model *circuit to circuit* delay variation, i.e., when timing behavior varies from circuit to circuit [36]. Partial fault simulation (i.e., fault sampling) is a viable practice, but fault coverage will no longer be a deterministic value but instead becomes a statistical estimate [56].

Recent delay fault simulation studies focus on fine-grained evaluation of SDDs. A GPU-accelerated time-domain simulation has been shown to be significantly faster compared to the traditional gate-level simulation [57]. Gate-level parallelism, fault-parallelism, and pattern-parallelism can be simultaneously exploited to exhaustively simulate SDDs [58]. Using logic-level and switch-level abstractions in *multi-level simulation* to expedite timing simulation on GPUs, a 89% run-time reduction over full switch-level simulation has been reported [59].

G. Delay fault test generation

Goals of ATPG include decreasing test data volume, decreasing test generation (i.e., computation) time, and increasing fault coverage. Numerous test generation methods have been proposed to overcome these challenges, to which this section discusses existing test generation methods and their nuances.

A PDF is illustrated in Figure 4 [1]. This four-gate circuit has 5 input to output paths of which three, $P1, P2$ and $P3$, are shown. Nominal gate delays ranging from 1 to 3 time units are noted on gates. Thus, $P3$ with delay of 6 units, is a critical path (shown with bold line). For testing PDF $\downarrow P3$, i.e., a falling transition propagating through $P3$, a vector $V1 = < 010 >$ initializes all signals to steady state so that an appropriate transition on path $P3$ can be created by vector $V2 = < 100 >$. This vector-pair can detect a falling path delay fault on $P3$ if the output is observed after a specific time period (clock period), which should be slightly longer than the critical path delay of 6 units. Note that faster paths $P1$ and $P2$ put the *off-path* signal for $P3$ in a non-controlling value so that the transition from primary input B is observable at the primary output and its timing indicates the delay of the path under test.

A careful look at Figure 4 shows that the time positions, 2, 4 and 6 units, of three transitions at the output are actually the

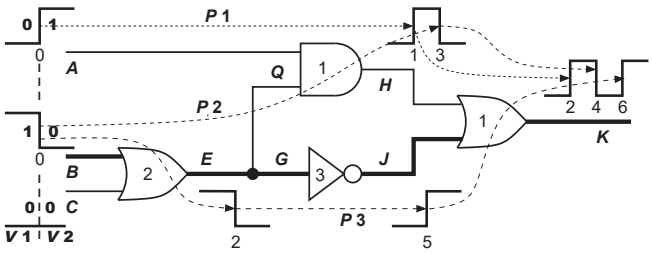

Figure 4: An example of path delay test [1].

delays of paths $P1$, $P2$ and $P3$, respectively. Each transition at the output is actually an input transition traveling through an activated path (see dotted line arrows), delayed in time by the path delay. In general, some transitions may be suppressed due to interference from off-path signals. If the delay of the target path $\downarrow P3$ exceeds 6 units due to a fault, then the last transition will be delayed beyond the observation instant determined by the clock period, and a 0 will be captured instead of the expected value 1, therefore, detecting the fault. If path $P1$ or $P2$ were to have a delay fault as well then their transitions could interfere with the output signal and the fault on $P3$ may go undetected.

PDF tests can be classified into two categories: *robust* tests, and *non-robust* tests. A *robust* PDF test detects the fault on a target path regardless of delays on other paths, while a *non-robust* PDF test guarantees fault detection only if all other paths are fault-free. Given multiple faults may be present in a circuit, the quality of non-robust tests is inferior to that of robust tests [60]. Note that a non-robust test is effective if all off-path signals are in their non-controlling states when the on-path transition arrives - this is called *static sensitization*. Although testing a fault robustly is desirable, some faults may only have non-robust tests. Majhi *et al.* [25] proposed 9-value logic for generating tests for robust and non-robust PDFs, simultaneously and showed test generation time is reduced by incorporating multiple backtrace procedures for signal value justification. Cheng and Chen [60] propose the use of circuit timing information fir generating higher-quality non-robust tests. Another ATPG algorithm [61] achieves high PDF coverage by capturing SDDs and a high TDF coverage to detect gross delay defects by selectively discounting non-robust paths to minimize overlap among already tested robust paths. That algorithm also clusters paths to improve the test vector quality and shows that clustering can reduce test data size by 40% compared to tests that do not cluster paths.

Fuchs *et al.* proposed deterministic test generation approaches for PDFs: DYNAMITE [62] and RESIST [63]. DYNAMITE employs 10-value logic and generates both robust and non-robust tests, but it handles only small circuits due to the use of a sorting algorithm to separate common paths from a specific PI to a given PO [64], which is impractical for large circuits. RESIST executes a sensitization step for common sub-paths, and is thus significantly faster than DYNAMITE.

ATPG for PDFs can be used to detect TDFs with significant

efficiency. ATPG for TDFs using PDF test generation has been applied to SDDs [65]. Reportedly, PDF tests can be used to detect all TDFs on a single path using a *three-pass method* [65]. A 12.5% reduction in test pattern volume, 35% reduction in ATPG runtime, and 5% improvement in fault coverage compared to a commercial TA-ATPG tool is reported.

ATPG algorithms for TDFs provide low fault coverage for SDDs since they do not consider the circuit timing to propagate SDDs through short paths [39], [66], [67]. This shortcoming of the TDF model, particularly where a faulty signal transitions through multiple clock cycles, has been observed [68]. A modified TDF model assumes *unspecified values*, *"X"*, in the circuit when the fault occurs [68]. This method allows delay faults of any magnitude to be modeled as a single fault. Another algorithm uses stuck-at fault ATPG to construct high-quality delay tests [22]. The reported experiments demonstrate a 20% reduction in test set size, test application time, and test data volume compared to TDF test ATPG tools without compromising fault coverage.

Clock delay faults (delay faults in a clock network) can be detected if there is clock skew at the clock port of a flip-flop. Metra *et al.* [69] argue that clock trees need to be separately tested since *flush tests* of scan chains and at-speed tests [70] are not reliable tests for detecting SDDs. Yang *et al.* [71] test SDDs in clock networks without requiring any change in the clock network. Test clocks are manipulated during the flush-test of scan chains showing that delays as small as $52.8\ ps$ were detected.

N-detect ATPG is effective for SDD detection without requiring circuit timing information [72], [73]. *N-detect* ATPG generates patterns to detect a fault *N*-times through different paths. The CPU run-time is smaller for *N-detect* ATPG compared to TA-ATPG, but *N-detect* ATPG generates a larger number of patterns. A comparative study shows that TA-ATPG and *N-detect* algorithms detect more SDDs than 1-detect (timing-unaware) ATPG, but the pattern count and CPU run-time increase linearly with *N* [39].

Since more than 50% of delay faults can be detected by timing-unaware TDF tests, a combined TDF ATPG and TA-ATPG reduces the number of test patterns compared to a TA-ATPG-only approach [74]. Test generation methods called *TDF+Top-off-SDD* and *Top-off-SDD+Top-off-TDF* that run a standard TDF testing method have been proposed [74]. *Top-off-SDD+Top-off-TDF* generates a smaller pattern set than *TDF+Top-off-SDD* but produces lower delay fault coverage.

Faster-than-at-speed tests (FAST) are applied to a circuit under test (CUT) at a frequency higher than the rated frequency [75]–[80] and have been found to effectively detect delay defects. FAST can detect very small hidden delay faults (small gate delay faults that can only be propagated along short paths) [81]. They require on-chip clock generation to be faster than the rated frequency of the device, but this can cause *yield loss* (classifying good chips as bad) due to increased IR drop [78]. Besides, unknown logic (*X-values*) are created when circuit paths do not complete their computation

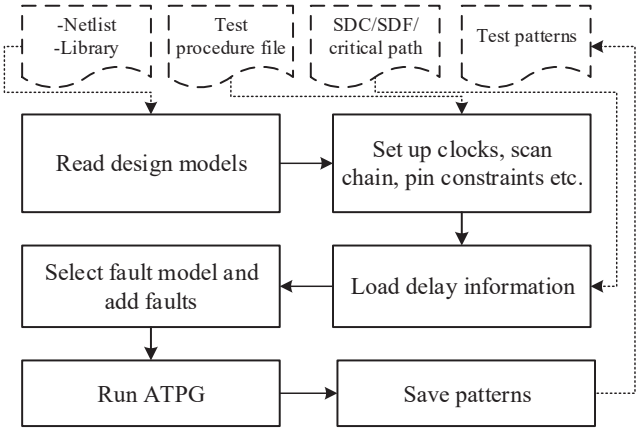

Figure 5: Simplified flow for transition and path delay ATPG.

at the time of sampling [82]. To reduce the number of *X-values*, *X-tolerant* test response compaction methods have been used [83]–[85]. Singh *et al.* [86] used multiplexers to select *X-value* free outputs every shift cycle of a FAST test, which although effective, requires a large pattern size to reach a given fault coverage. Hellebrand *et al.* [32] added *X-canceling* MISRs [84], [87] to an embedded memory for storing intermediate signatures, but this exacerbates memory size optimization problems.

Recent advances in Boolean satisfiability techniques have created efficient SAT-based ATPG methods [88], [89], which have been applied to TA-ATPG. A pseudo-Boolean optimization (PBO) [90] TA-ATPG approach was proposed [91] for generating robust tests for TDFs. A robust PDF ATPG using PBO that can generate tests for most paths without explicitly considering path lengths has been proposed [91]. An As-Robust-As-Possible (ARAP) [92] ATPG employs PBO to improve fault coverage by including satisfiability conditions required for robust test patterns. A Boolean difference-based ATPG generates a sequence of instructions for functional tests using a gate-level netlist and preprocessing steps [93]–[95].

H. Industrial tools and practices

Industry has adopted numerous algorithms to improve delay test quality, reduce the size of test sets, and decrease fault simulation time. For example, *TetraMAX ATPG* from Synopsys [96] uses timing information from *PrimeTime*, a static timing analyzer [97], to achieve timing resolution needed for SDD detection. *TetraMAX ATPG* does not use FAST algorithms to avoid unnecessary yield loss. Mentor Graphics provides *Tessent FastScan* for at-speed delay fault test that includes multiple detection, timing-awareness, and critical path testing [98]. *Modus DFT* from Cadence Design System applies timing information from SDF or SDC files to generated test patterns for delay fault simulation [99]. Most tools use different algorithms to generate test patterns for delay fault. However, the test pattern generation can be simplified and presented as a flow chart shown in Figure 5. Note that an appropriate delay information file must be used during ATPG

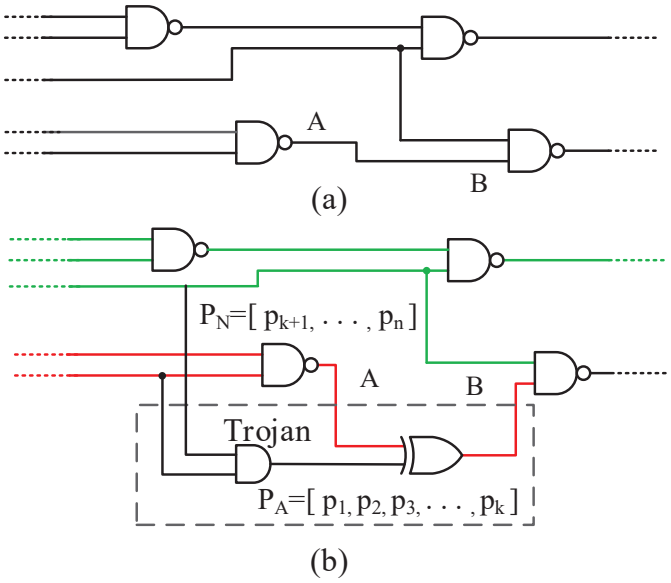

Figure 6: An illustration of Hardware Trojan (a) Part of an original circuit (b) Paths highlighted in red (set P_A) indicate the changes in original paths due to Trojan insertion and paths highlighted in green indicate the unaffected paths (set P_N).

mode selection. For example, path delay fault generation in *TetraMAX ATPG* requires critical path information generated typically by *PrimeTime.*

III. New applications and future research

In this section, we will discuss newer applications and possible future research direction of the delay fault test.

A. Hardware Trojan and recycled IC detection

Delay tests have potential applications in Trojan detection. A hardware Trojan (HT) is a malicious modification of an original circuit so that an adversary may exercise illegitimate control over a system [100]. Small circuit modifications can noticeably alter delay of the paths affected by hardware Trojan. A design's known delay can be compared against a device's measured delay to detect if there are modifications by hardware Trojans in the circuit. Figure 6 illustrates the effect of a Trojan on the path delay of a circuit. Clearly, any logic gate insertion will change some paths ($P_A = [p_1, p_2, p_3, ..., p_K]$ shown in red lines in Figure 6(b)) in the circuit that will ultimately affect the delay characteristics. To detect these changes, applied test vectors must test affected paths (P_A). Plusquellic *et al.* [101] devised measurements to detect small changes in delays introduced by capacitive loading from HT, and also proposed an on-chip embedded test structure to detect them. Li *et al.* [102] proposed measuring register-to-register delay using an extra shadow register in parallel with destination registers at the end of combinational paths. The shadow register is clocked with a negatively skewed clock with respect to the system clock, and the output of the destination register is compared to the output of the shadow register. However, this method requires extra on-chip hardware and process variation can introduce errors during detection process. Cha *et al.* [103] proposed a shortest path-based HT detection method to minimize the effects of process variations and argued that an HT has an influence on the shortest path delay and therefore will suppress the delay distribution introduced by process variations. In [104], a test set with high fault coverage is applied to a Trojan-free chip to get the path delay information. CUT's are tested with the same test set and the observed path delay information is compared with the reference fingerprint to detect HTs.

When a chip is used in the field, path delays in the circuit increase due to aging [105]. As a result, the delay distribution of paths shifts from its original new-chip value and one can use this information to determine whether or not the chip has been used before. Using this concept, signature path delays have been used to identify recycled ICs [106].

Recycled ICs are often reclaimed from the used electronic systems and sold as new in the open market [107], [108]. These chips exhibit poor performance and reduced remaining useful lifetime (RUL) [109]. Various methods have been proposed over the years focusing on the change in delays due to aging [110]–[113]. Moudgil *et al.* [110] exploited the relationship in terms of delays among two or more paths within a chip. First, delays of selected paths are measured for a new chip and for a trusted chip subjected to accelerated aging. Then, a mathematical method, e.g.,, using shortest perpendicular distance between the delays of an untrusted chip under test and the trusted aged chip, is predicts the age. Zheng *et al.* [111] presented a characterization method, which relies on the extraction of scan path delay signatures of a chip. In contrast to the traditional path delay based techniques, Guin *et al.* [112], [113] proposed a ring oscillator (RO) based self-referencing approach, where the frequencies of two ROs are compared for detecting recycled ICs.

Let us understand a path delay-based approach to identify recycled ICs. Figure 7 illustrates the method [106]. A large number of fast aging paths in the circuit are selected using workload analysis. Note that these paths will display larger delays when they age. This approach consists of two phases - fingerprint generation and recycled chip identification. During the fingerprint generation phase, delays of selected paths are measured for a large number of newly manufactured chips at the foundry. Clock sweeping technique, which is commonly employed by industry for speed binning, is used to measure the path delays. To model accurate process variation, a large number (m) of chips is required and for n paths, each chip yields n delays. In a n-dimensional space, path delays of each chip is represented by a point, thus creating a cluster of m points. This statistical data is subjected to principal component analysis (PCA). Three principal components then provide a m-point fingerprint in the form of a convex hull. In the identification phase, n path delays are measured for the chip under test. If this data falls outside the fingerprint area then the chip is classified as recycled.

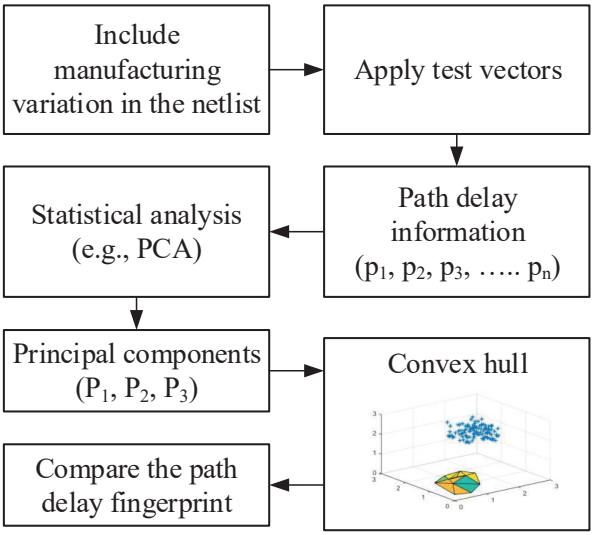

Figure 7: Path delay fingerprinting for detecting recycled ICs.

Even though delay-based tests have the potential to address the aforementioned issues, their applicability in detecting non-authentic chips still remains questionable. The primary reason is the manufacturing process variation (PV), which makes the device parameters (e.g., delay) vary. The delay variation in different chips can be as much as 20% in current manufacturing processes. It, therefore, seems quite possible that the effect of a hardware Trojan remains well-hidden under PV, unless the future research shows otherwise. For recycled chip identification, we might have a better chance When the chip has been used for a long time because the delay increase from aging would exceed delay variation from PV. However, we believe that the measurement of steady state quiescent current (I_{DDQ}) rather than delay can be a better approach for detecting recycled ICs. Recent work shows that the influence of process variation can be effectively eliminated when ΔI_{DDQ} is used [114].

B. Signal integrity and process variation

Efficient simulation and ATPG that consider crosstalk-induced delays are essential for reliable delay fault testing. Crosstalk [115] and power supply noise [116] create pattern-dependent delays that make achieving high fault coverage challenging in sub-micron technologies [117]. To improve simulation accuracy for such faults, an enhanced fault model would be useful; a fuzzy model has been proposed [118]. Study of correlation between delay fault coverage and process variation is another potential research problem.

C. Parallel computing

Parallel algorithms for reducing computation time of test generation and simulation have been attractive research topics [52], [119]. Timing simulation is expensive, especially under process variations. Circuit-level simulation with multiple delay distributions can be done using GPUs by leveraging their highly parallel nature. Simulation of multiple copies of identical circuits concurrently on GPUs significantly reduces simulation time [120].

D. Research scope in emerging nano-devices

Emerging technologies need new techniques for testing faults. Logic cells based on ambipolar devices (e.g., carbon nanotube FETs and silicon nano-wire FETs) show complex electrical characteristics. The static behavior of these ambipolar devices is compatible with general CMOS/FinFET behavior, but dynamic switching behavior differs because of in-field controllability of polarity of these devices. CMOS and FinFET models do not cover all defects that may be present in ambipolar devices [121], thus standard gate level models cannot be used to generate test patterns with high fault coverage. Dalpasso *et al.* [122] argue that gate-level models fail to customize test patterns for specific types of delay faults, thus allowing some faults to escape. They developed a switch-level model to generate patterns considered equivalent in gate-level models. Multiple FinFETs can be used in parallel to increase the drive strength of gates and one or more broken fins in the parallel structure may not cause the circuit to fail, but such defects can lead to SDDs. Another emerging technology, "gate all around" (GAA), for 7nm transistors can suffer from similar SDDs [65].

E. Faster than critical path synchronous circuits

Synchronous circuits can be made to work faster than critical paths would allow [123]. Traditionally, system clock is designed with timing margins for process, voltage and temperature (PVT) variations. As critical paths are not active for every functional stimulus, circuit can be run faster for the inactive states of critical paths and therefore timing margin can be tightened for those functional inputs [124]. These types of circuits can be self-adaptive as they may increase frequency until error is detected [125]. A delay monitor inserts extra clock cycles only when longer paths are activated. Possibly, such a circuit will continue to work at a slower speed in the presence of delay faults. Its testing needs investigation.

F. Delay faults in asynchronous and self-timed circuits

Both asynchronous and self-timed circuits may not have a well defined clock signal. Hence, all traditional test methods cannot be directly applied. Although, such circuits are difficult to design and manufacture, they have advantages in terms of economy of hardware, timing and power [126]. Asynchronous circuits are designed to tolerate varying delays, but a delay fault can still corrupt the system [127], [128].

Asynchronous circuits are often not compatible with traditional design and test tools. The requirements for delay test depends on the type of the circuit [129], and high-speed asynchronous designs do need some form of delay testing [130]. An LOC-based delay test method can work with single-rail, bundled-data, self-timed asynchronous structures [131], [132].

IV. Conclusion

For a reader interested in working in the area of delay test, this article provides a large set of references, as the field of delay testing is wide and reaching maturity. Section III, outlines several research topics in need of exploration. Authors' believe that continued progress in delay fault testing of VLSI circuits and systems will yield significant contributions to industry and will produce techniques beneficial to other areas in test and circuit design.

References

[1] M. L. Bushnell and V. D. Agrawal, *Essentials of Electronic Testing for Digital, Memory and Mixed-Signal VLSI Circuits*. Springer Science & Business Media, 2004, vol. 17.

[2] A. Krstic and K.-T. Cheng, *Delay fault testing for VLSI circuits*. Springer Science & Business Media, 1998, vol. 14.

[3] G. Sai, B. Halak, and M. Zwolinski, "Multi-path aging sensor for cost-efficient delay fault prediction," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 65, no. 4, pp. 491–495, 2018.

[4] T. Shah, V. Singh, and A. Matrosova, "ROBDD based path delay fault testable combinational circuit synthesis," in *Proc. IEEE East-West Design & Test Symposium (EWDTS)*, 2016, pp. 1–4.

[5] M. J. S. Smith, *Application-Specific Integrated Circuits*. Reading, Massachusetts: Addison-Wesley, 1997.

[6] W. Elmore, "The transient response of damped linear networks with particular regard to wideband amplifiers," *J. Appl. Phys.*, vol. 19, no. 1, pp. 55–63, Jan. 1948.

[7] J. Rubinstein, P. Penfield, and M. A. Horowitz, "Signal delay in RC tree networks," *IEEE Trans. on Computer-Aided Design*, vol. CAD-2, no. 3, pp. 202–211, Jul. 1983.

[8] E. G. Friedman and J. H. Mulligan, "Ramp input response of RC tree networks," *Analog Integrated Circuits and Signal Processing*, vol. 14, pp. 53–58, Jan. 1997.

[9] G. W. Roberts and A. S. Sedra, *SPICE, Second Edition*. New York: Oxford University Press, 1996.

[10] T. I. Kirkpatrick and N. R. Clark, "PERT as an aid to logic design," *IBM J. Res. Dev.*, vol. 10, no. 2, pp. 135–141, Mar. 1966.

[11] H. Chang and S. S. Sapatnekar, "Statistical timing analysis considering spatial correlations using a single PERT-like traversal," in *Proc. International Conf. on Computer-Aided Design*, 2003, pp. 621–625.

[12] V. D. Agrawal, "Synchronous path analysis in MOS circuit simulator," in *Proc. 19th Design Automation Conf.*, 1982, pp. 629–635.

[13] R. B. Hitchcock, "Timing verification and the timing analysis program," in *Proc. 19th Design Automation Conf.*, 1982, pp. 594–604.

[14] S. Kundu, "An incremental algorithm for identification of longest (shortest) paths," *Integration, the VLSI Journal*, vol. 17, no. 1, pp. 25–35, Aug. 1994.

[15] N. Maheshwari and S. S. Sapatnekar, *Timing Analysis and Optimization of Sequential Circuits*. Springer, 1999.

[16] G. DeMicheli, *Synthesis and Optimization of Digital Circuits*. New York: McGraw-Hill, 1994.

[17] M. A. Gharaybeh, V. D. Agrawal, M. L. Bushnell, and C. G. Parodi, "False-path removal using delay fault simulation," *J. Electronic Testing: Theory and Applic.*, vol. 16, pp. 463–476, Oct. 2000.

[18] G. L. Smith, "Model for delay faults based upon paths," in *Proc. International Test Conference*, 1985, pp. 342–351.

[19] M. A. Gharaybeh, M. L. Bushnell, and V. D. Agrawal, "Classification and test generation for path-delay faults using single struck-at fault tests," *J. Electronic Testing: Theory and Applic.*, vol. 11, no. 1, pp. 55–67, 1997.

[20] T. Liu, Y. Zhou, Y. Liu, and S. Cai, "Harzard-based ATPG for improving delay test quality," *J. Electronic Testing: Theory and Applic.*, vol. 31, no. 1, pp. 27–34, 2015.

[21] J. A. Waicukauski, E. Lindbloom, B. K. Rosen, and V. S. Iyengar, "Transition fault simulation," *IEEE Design & Test of Computers*, vol. 4, no. 2, pp. 32–38, 1987.

[22] X. Liu, M. S. Hsiao, S. Chakravarty, and P. J. Thadikaran, "Efficient transition fault ATPG algorithms based on stuck-at test vectors," *J. Electronic Testing: Theory and Applic.*, vol. 19, no. 4, pp. 437–445, 2003.

[23] A. K. Majhi, J. Jacob, L. M. Patnaik, and V. D. Agrawal, "On test coverage of path delay faults," in *Proc. 9th International Conference on VLSI Design*, 1996, pp. 418–421.

[24] A. K. Majhi and V. D. Agrawal, "Delay fault models and coverage," in *Proc. 11th International Conference on VLSI Design*, 1998, pp. 364–369.

[25] A. K. Majhi, J. Jacob, L. M. Patnaik, and V. D. Agrawal, "An efficient automatic test generation system for path delay faults in combinational circuits," in *Proc. 8th International Conference on VLSI Design*, Jan. 1995, pp. 161–165.

[26] A. K. Pramanick and S. M. Reddy, "On the detection of delay faults," in *Proc. International Test Conference*, 1988, pp. 845–856.

[27] V. S. Iyengar, B. K. Rosen, and J. A. Waicukauski, "On computing the sizes of detected delay faults," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 9, no. 3, pp. 299–312, March 1990.

[28] F. Pöhl and W. Anheier, "Quality determination for gate delay fault tests considering three-state elements," *J. Electronic Testing: Theory and Applic.*, vol. 14, no. 1-2, pp. 49–55, 1999.

[29] K. Heragu, J. H. Patel, and V. D. Agrawal, "Segment delay faults: a new fault model," in *Proc. 14th IEEE VLSI Test Symposium*, 1996, pp. 32–39.

[30] E. S. Park, M. Mercer, and T. Williams, "Statistical delay fault coverage and defect level for delay faults," in *Proc. International Test Conference*, 1988, pp. 492–499.

[31] R. Mattiuzzo, D. Appello, and C. Allsup, "Small-delay-defect testing," *EDN (Electrical Design News)*, vol. 54, no. 13, p. 28, 2009.

[32] S. Hellebrand, T. Indlekofer, M. Kampmann, M. A. Kochte, C. Liu, and H.-J. Wunderlich, "FAST-BIST: Faster-than-at-Speed BIST targeting hidden delay defects," in *Proc. International Test Conference*, 2014, pp. 1–8.

[33] Y. M. Kim, Y. Kameda, H. Kim, M. Mizuno, and S. Mitra, "Low-cost gate-oxide early-life failure detection in robust systems," in *Proc. IEEE Symp. VLSI Circuits (VLSIC)*, 2010, pp. 125–126.

[34] M. Sauer, A. Czutro, I. Polian, and B. Becker, "Small-delay-fault ATPG with waveform accuracy," in *Proc. International Conference on Computer-Aided Design*, 2012, pp. 30–36.

[35] S. Eggersglüß and R. Drechsler, *High Quality Test Pattern Generation and Boolean Satisfiability*. Springer Science & Business Media, 2012.

[36] I. Polian, B. Becker, S. Hellebrand, H.-J. Wunderlich, and P. Maxwell, "Towards variation-aware test methods," in *Proc. 16th IEEE European Test Symposium (ETS)*, 2011, pp. 219–225.

[37] X. Qian and A. D. Singh, "Distinguishing resistive small delay defects from random parameter variations," in *Proc. 19th IEEE Asian Test Symposium (ATS)*, 2010, pp. 325–330.

[38] S. Davidson, "Book Review: Test and Diagnosis for Small-Delay Defects, by M. Tehranipoor, *et al.*, Springer, 2011," *IEEE Design & Test of Computers*, vol. 29, no. 1, pp. 65–67, Feb. 2012.

[39] M. Tehranipoor, K. Peng, and K. Chakrabarty, "Delay test and small-delay defects," in *Test and Diagnosis for Small-Delay Defects*. Springer, 2011, pp. 21–36.

[40] J. Savir, "Skewed-load transition test: Part I, calculus," in *Proc. International Test Conference*, 1992, p. 705.

[41] J. Savir and S. Patil, "Broad-side delay test," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 13, no. 8, pp. 1057–1064, 1994.

[42] B. I. Dervisoglu and G. E. Stong, "Design for testability using scanpath techniques for path-delay test and measurement," in *Proc. International Test Conference*, 1991, p. 365.

[43] F. Zokaee, H. Sabaghian-Bidgoli, V. Janfaza, P. Behnam, and Z. Navabi, "A novel SAT-based ATPG approach for transition delay faults," in *Proc. IEEE International High Level Design Validation and Test Workshop (HLDVT)*, 2017, pp. 17–22.

[44] W.-C. Lai, A. Krstic, and K.-T. Cheng, "On testing the path delay faults of a microprocessor using its instruction set," in *Proc. 18th IEEE VLSI Test Symposium*, 2000, pp. 15–20.

[45] ——, "Test program synthesis for path delay faults in microprocessor cores," in *Proc. International Test Conference*, 2000, pp. 1080–1089.

[46] J. Rearick, "Too much delay fault coverage is a bad thing," in *Proc. International Test Conference*, 2001, pp. 624–633.

[47] S. Chakrabarti, S. Das, D. K. Das, and B. B. Bhattacharya, "Synthesis of symmetric functions for path-delay fault testability," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 19, no. 9, pp. 1076–1081, 2000.

[48] K. T. Le, D. H. Baik, and K. K. Saluja, "Test time reduction to test for path-delay faults using enhanced random-access scan," in *Proc. 20th International Conference on VLSI Design*, 2007, pp. 769–774.

[49] I. Pomeranz and S. M. Reddy, "Design-for-testability for improved path delay fault coverage of critical paths," in *Proc. 21st International Conference on VLSI Design*, 2008, pp. 175–180.

[50] M. Siebert and E. Gramatová, "Delay fault coverage increasing in digital circuits," in *Proc. IEEE EuroMicro Conference on Digital System Design (DSD)*, 2013, pp. 475–478.

[51] N. Devtaprasanna, A. Gunda, P. Krishnamurthy, S. M. Reddy, and I. Pomeranz, "A novel method of improving transition delay fault coverage using multiple scan enable signals," in *Proc. IEEE International Conference on Computer Design (ICCD)*, 2005, pp. 471–474.

[52] E. Schneider, M. A. Kochte, S. Holst, X. Wen, and H.-J. Wunderlich, "GPU-accelerated simulation of small delay faults," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 36, no. 5, pp. 829–841, 2017.

[53] M. A. Gharaybeh, M. L. Bushnell, and V. D. Agrawal, "The path-status graph with application to delay fault simulation," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 17, no. 4, pp. 324–332, 1998.

[54] 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," in *Proc. International Test Conference*, 2005, pp. 9–pp.

[55] S. Bose, H. Grimes, and V. D. Agrawal, "Delay fault simulation with bounded gate delay mode," in *Proc. International Test Conf.*, 2007, pp. 1–10.

[56] A. Czutro, M. E. Imhof, J. Jiang, A. Mumtaz, M. Sauer, B. Becker, I. Polian, and H.-J. Wunderlich, "Variation-aware fault grading," in *Proc. 21st IEEE Asian Test Symposium (ATS)*, 2012, pp. 344–349.

[57] S. Holst, M. E. Imhof, and H.-J. Wunderlich, "High-throughput logic timing simulation on GPGPUs," *ACM Transactions on Design Automation of Electronic Systems*, vol. 20, no. 3, pp. 37:1–37:22, 2015.

[58] E. Schneider, S. Holst, M. A. Kochte, X. Wen, and H.-J. Wunderlich, "GPU-accelerated small delay fault simulation," in *Proc. Design, Automation & Test in Europe Conference & Exhibition*, 2015, pp. 1174–1179.

[59] E. Schneider, M. A. Kochte, and H.-J. Wunderlich, "Multi-level timing simulation on GPUs," in *Proc. 23rd Asia and South Pacific Design Automation Conference (ASP-DAC)*, 2018, pp. 470–475.

[60] K.-T. Cheng and H.-C. Chen, "Generation of high quality non-robust tests for path delay faults," in *Proc. 31st Design Automation Conference*, 1994, pp. 365–369.

[61] P. Gupta and M. S. Hsiao, "High quality ATPG for delay defects," in *Proc. International Test Conf.*, 2003, pp. 584–591.

[62] K. Fuchs, F. Fink, and M. H. Schulz, "DYNAMITE: An efficient automatic test pattern generation system for path delay faults," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 10, no. 10, pp. 1323–1335, 1991.

[63] K. Fuchs, M. Pabst, and T. Rossel, "RESIST: A recursive test pattern generation algorithm for path delay faults considering various test classes," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 13, no. 12, pp. 1550–1562, 1994.

[64] M. H. Schulz, K. Fuchs, and F. Fink, "Advanced automatic test pattern generation techniques for path delay faults," in *19th IEEE International Symposium on Fault-Tolerant Computing, Digest of Papers*, 1989, pp. 44–51.

[65] A. Srivastava, A. D. Singh, V. Singh, and K. K. Saluja, "Exploiting path delay test generation to develop better TDF tests for small delay defects," in *Proc. International Test Conference*, 2017, pp. 1–10.

[66] N. Ahmed, M. Tehranipoor, and V. Jayaram, "Timing-based delay test for screening small delay defects," in *Proceedings of the 43rd annual Design Automation Conference*, 2006, pp. 320–325.

[67] R. Putman and R. Gawde, "Enhanced timing-based transition delay testing for small delay defects," in *Proc. 24th IEEE VLSI Test Symposium*, 2006.

[68] I. Pomeranz and S. M. Reddy, "Unspecified transition faults: a transition fault model for at-speed fault simulation and test generation," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 27, no. 1, pp. 137–146, 2008.

[69] C. Metra, M. Omana, T. M. Mak, and S. Tam, "New design for testability approach for clock fault testing," *IEEE Transactions on Computers*, vol. 61, no. 4, pp. 448–457, 2012.

[70] Y. Higami, H. Takahashi, S.-y. Kobayashi, and K. K. Saluja, "Fault simulation and test generation for clock delay faults," in *Proc. 16th Asia & South Pacific Design Automation Conf. (ASP-DAC)*, 2011, pp. 799–805.

[71] S.-F. Yang, Z.-Y. Wen, S.-Y. Huang, K.-H. Tsai, and W.-T. Cheng, "Circuit and methodology for testing small delay faults in the clock network," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2018.

[72] Y. Huang, "On N-detect pattern set optimization," in *Proc. 7th International Symp. on Quality Electronic Design (ISQED)*, 2006, pp. 6–pp.

[73] M. E. Amyeen, S. Venkataraman, A. Ojha, and S. Lee, "Evaluation of the quality of N-detect scan ATPG patterns on a processor," in *Proc. International Test Conference*, 2004, pp. 669–678.

[74] S. K. Goel, N. Devta-Prasanna, and R. P. Turakhia, "Effective and efficient test pattern generation for small delay defect," in *Proc. 27th IEEE VLSI Test Symposium*, 2009, pp. 111–116.

[75] W.-W. Mao and M. D. Ciletti, "A variable observation time method for testing delay faults," in *Proc. 27th ACM/IEEE Design Automation Conference*, 1990, pp. 728–731. [Online]. Available: http://doi.acm.org/10.1145/123186.123454

[76] H. Yan and A. D. Singh, "Experiments in detecting delay faults using multiple higher frequency clocks and results from neighboring die," in *Proc. International Test Conference*, vol. 1, 2003, pp. 105–111.

[77] ——, "Evaluating the effectiveness of detecting delay defects in the slack interval: a simulation study," in *Proc. International Test Conference*, 2004, pp. 242–251.

[78] N. Ahmed, M. Tehranipoor, and V. Jayaram, "A novel framework for faster-than-at-speed delay test considering IR-drop effects," in *Proc. IEEE/ACM International Conference on Computer Aided Design*, Nov. 2006, pp. 198–203.

[79] N. Ahmed and M. Tehranipoor, "A novel faster-than-at-speed transition-delay test method considering IR-drop effects," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 28, no. 10, pp. 1573–1582, Oct. 2009.

[80] T. Y. Li, S. Y. Huang, H. J. Hsu, C. W. Tzeng, C. T. Huang, J. J. Liou, H. P. Ma, P. C. Huang, J. C. Bor, C. W. Wu, C. C. Tien, and M. Wang, "AF-Test: Adaptive-frequency scan test methodology for small-delay defects," in *Proc. IEEE 25th International Symposium on Defect and Fault Tolerance in VLSI Systems*, Oct. 2010, pp. 340–348.

[81] M. Kampmann, M. A. Kochte, E. Schneider, T. Indlekofer, S. Hellebrand, and H.-J. Wunderlich, "Optimized selection of frequencies for faster-than-at-speed test," in *Proc. IEEE 24th Asian Test Symposium (ATS)*, 2015, pp. 109–114.

[82] M. Kampmann and S. Hellebrand, "Design for small delay test - a simulation study," *Microelectronics Reliability*, vol. 80, pp. 124–133, 2018.

[83] S. Mitra, S. S. Lumetta, and M. Mitzenmacher, "X-tolerant signature analysis," in *Proc. International Test Conference*, 2004, pp. 432–441.

[84] N. A. Touba, "X-canceling MISR an X-tolerant methodology for compacting output responses with unknowns using a MISR," in *Proc. International Test Conference*, 2007, pp. 1–10.

[85] D. Czysz, G. Mrugalski, N. Mukherjee, J. Rajski, and J. Tyszer, "On compaction utilizing inter and intra-correlation of unknown states," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 29, no. 1, pp. 117–126, 2010.

[86] A. Singh, C. Han, and X. Qian, "An output compression scheme for handling X-states from over-clocked delay tests," in *Proc. 28th IEEE VLSI Test Symposium*, 2010, pp. 57–62.

[87] R. Garg, R. Putman, and N. A. Touba, "Increasing output compaction in presence of unknowns using an X-canceling MISR with deterministic observation," in *Proc. 26th IEEE VLSI Test Symp.*, 2008, pp. 35–42.

[88] T. Larrabee, "Test pattern generation using Boolean satisfiability," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 11, no. 1, pp. 4–15, 1992.

[89] S. Alouneh, S. Abed, M. H. Al Shayeji, and R. Mesleh, "A comprehensive study and analysis on SAT-solvers: advances, usages and achievements," *Artificial Intelligence Review*, pp. 1–27, 2018.

[90] Y. Crama and P. L. Hammer, "Pseudo-Boolean optimization," *Handbook of Applied Optimization*, pp. 445–450, 2002.

[91] S. Eggersglüß, M. Yilmaz, and K. Chakrabarty, "Robust timing-aware test generation using pseudo-Boolean optimization," in *Proc. IEEE 21st Asian Test Symposium (ATS)*, 2012, pp. 290–295.

[92] S. Eggersglüß and R. Drechsler, "As-robust-as-possible test generation in the presence of small delay defects using pseudo-Boolean optimization," in *Proc. Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2011, pp. 1–6.

[93] S. Gurumurthy, R. Vemu, J. A. Abraham, and D. G. Saab, "Automatic generation of instructions to robustly test delay defects in processors," in *Proc. 12th IEEE European Test Symposium*, 2007, pp. 173–178.

[94] F. A. Hussin, N. H. Hamid, N. B. Z. Ali *et al.*, "Automatic generation of test instructions for path delay faults based-on stuck-at fault in processor cores using assignment decision diagram," in *Proc. 5th International Conference on Intelligent and Advanced Systems (ICIAS)*, 2014, pp. 1–5.

[95] M. Yilmaz, K. Chakrabarty, and M. Tehranipoor, "Test-pattern selection for screening small-delay defects in very-deep submicrometer integrated circuits," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 29, no. 5, pp. 760–773, 2010.

[96] *TetraMAX ATPG:Automatic Test Pattern Generation*, Synopsys, Inc, Oct. 2018.

[97] H. Bhatnagar, *Advanced ASIC Chip Synthesis, Second Edition*. Springer, 2002.

[98] *Tessent FastScan: Advanced Automatic Test Pattern Generation*, Mentor Graphics Corporation, Oct. 2018.

[99] *Modus DFT Software Solution*, Cadence Design Systems, Inc, Oct. 2018.

[100] Z. Zhou, U. Guin, and V. D. Agrawal, "Modeling and test generation for combinational hardware Trojans," in *Proc. IEEE 36th VLSI Test Symposium (VTS)*, April 2018, pp. 1–6.

[101] J. Plusquellic and F. Saqib, "Detecting hardware Trojans using delay analysis," in *The Hardware Trojan War*. Springer, 2018, pp. 219–267.

[102] J. Li and J. Lach, "At-speed delay characterization for IC authentication and Trojan horse detection," in *Proc. IEEE International Workshop on Hardware-Oriented Security and Trust (HOST)*, 2008, pp. 8–14.

[103] B. Cha and S. K. Gupta, "Trojan detection via delay measurements: A new approach to select paths and vectors to maximize effectiveness and minimize cost," in *Proceedings of the Conference on Design, Automation and Test in Europe*, 2013, pp. 1265–1270.

[104] Y. Jin and Y. Makris, "Hardware Trojan detection using path delay fingerprint," in *Proc. IEEE International Workshop on Hardware-Oriented Security and Trust (HOST)*, 2008, pp. 51–57.

[105] D. K. Schroder and J. A. Babcock, "Negative bias temperature instability: Road to cross in deep submicron silicon semiconductor manufacturing," *Jour. applied Physics*, vol. 94, no. 1, pp. 1–18, 2003.

[106] X. Zhang, K. Xiao, and M. Tehranipoor, "Path-delay fingerprinting for identification of recovered ICs," in *Proc. IEEE International Symp. on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT)*, 2012, pp. 13–18.

[107] U. Guin, D. DiMase, and M. Tehranipoor, "Counterfeit integrated circuits: Detection, avoidance, and the challenges ahead," *J. Electronic Testing: Theory and Applic.*, vol. 30, no. 1, pp. 9–23, 2014.

[108] M. M. Tehranipoor, U. Guin, and D. Forte, *Counterfeit Integrated Circuits: Detection and Avoidance*. Springer, 2015.

[109] U. Guin, K. Huang, D. DiMase, J. Carulli, M. Tehranipoor, and Y. Makris, "Counterfeit integrated circuits: A rising threat in the global semiconductor supply chain," *Proceedings IEEE*, vol. 102, no. 8, pp. 1207–1228, Aug. 2014.

[110] R. Moudgil, D. Ganta, L. Nazhandali, M. Hsiao, C. Wang, and S. Hall, "A novel statistical and circuit-based technique for counterfeit detection in existing ICs," in *Proc. 23rd ACM International Conference on Great Lakes Symposium on VLSI (GLSVLSI)*, 2013, pp. 1–6.

[111] Y. Zheng, X. Wang, and S. Bhunia, "SACCI: Scan-based characterization through clock phase sweep for counterfeit chip detection," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 23, no. 5, pp. 831–841, 2015.

[112] U. Guin, X. Zhang, D. Forte, and M. Tehranipoor, "Low-cost on-chip structures for combating die and IC recycling," in *Proc. ACM/IEEE 51st Design Automation Conference*, 2014, pp. 1–6.

[113] U. Guin, D. Forte, and M. Tehranipoor, "Design of accurate low-cost on-chip structures for protecting integrated circuits against recycling," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 24, no. 4, pp. 1233–1246, 2016.

[114] P. Chowdhury, U. Guin, A. D. Singh, and V. D. Agrawal, "Two-pattern ΔI_{DDQ} test for recycled IC detection," in *Proc. 32nd International Conference on VLSI Design*, 2019, pp. 82–87.

[115] A. Krstic, J.-J. Liou, Y.-M. Jiang, and K.-T. Cheng, "Delay testing considering crosstalk-induced effects," in *Proc. International Test Conference*, 2001, pp. 558–567.

[116] A. Krstic, Y.-M. Jiang, and K.-T. Cheng, "Delay testing considering power supply noise effects," in *Proc. International Test Conference*, 1999, pp. 181–190.

[117] D. M. H. Walker and M. S. Hsiao, "Delay Testing," in *System-on-Chip Test Architectures, Chapter 6*, L.-T. Wang, C. E. Stroud, and N. A. Touba, Eds. Burlington, Massachusetts: Morgan Kaufmann, 2008, pp. 263–306.

[118] M. Favalli, "A fuzzy model for path delay fault detection," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 13, no. 8, pp. 943–956, 2005.

[119] P. Banerjee, *Parallel Algorithms for VLSI Computer-Aided Design*. Englewood Cliffs, New Jersey: PTR Prentice Hall, 1994.

[120] H. Li, D. Xu, and K.-T. Cheng, "GPU-accelerated fault simulation and its new applications," in *Proc. IEEE International Symp. on VLSI Design, Automation and Test (VLSI-DAT)*, 2011, pp. 1–4.

[121] H. G. Mohammadi, P.-E. Gaillardon, and G. DeMicheli, "Fault modeling in controllable polarity silicon nanowire circuits," in *Proc. Design, Automation & Test in Europe Conference & Exhibition*, 2015, pp. 453–458.

[122] M. Dalpasso, D. Bertozzi, and M. Favalli, "A Boolean model for delay fault testing of emerging digital technologies based on ambipolar devices," in *Proc. Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2018, pp. 297–300.

[123] R. T. Uppu, R. K. Uppu, A. D. Singh, and A. Chatterjee, "A high throughput multiplier design exploiting input based statistical distribution in completion delays," in *Proc. 26th International Conference on VLSI Design*, Jan. 2013, pp. 109–114.

[124] R. K. Uppu, R. T. Uppu, A. D. Singh, and I. Polian, "Better-than-worst-case timing design with latch buffers on short paths," in *Proc. 27th International Conference on VLSI Design*, Jan. 2014, pp. 133–138.

[125] I. Polian, J. Jiang, and A. Singh, "Detection conditions for errors in self-adaptive better-than-worst-case designs," in *2014 19th IEEE European Test Symposium (ETS)*, May 2014, pp. 1–6.

[126] I. E. Sutherland, "The tyranny of the clock," *Comm. ACM*, vol. 55, no. 10, pp. 35–36, Oct. 2012.

[127] H. Hulgaard, S. M. Burns, and G. Borriello, "Testing asynchronous circuits: A survey," *Integration, the VLSI Journal*, vol. 19, no. 3, pp. 111–131, Nov. 1995.

[128] S. Zeidler and M. Krstić, "A survey about testing asynchronous circuits," in *Proc. IEEE European Conference on Circuit Theory and Design (ECCTD)*, 2015, pp. 1–4.

[129] F. Shi and Y. Makris, "Testing delay faults in asynchronous handshake circuits," in *Proc. IEEE/ACM international conference on Computer-aided design*, 2006, pp. 193–197.

[130] Y. Monnet, M. Renaudin, and R. Leveugle, "Designing resistant circuits against malicious faults injection using asynchronous logic," *IEEE Transactions on Computers*, vol. 55, no. 9, pp. 1104–1115, Sep. 2006.

[131] O. A. Hasib, D. Crépeau, T. Awad, A. Dulipovici, Y. Savaria, and C. Thibeault, "Exploiting built-in delay lines for applying launch-on-capture at-speed testing on self-timed circuits," in *Proc. 36th IEEE VLSI Test Symposium (VTS)*, 2018, pp. 1–6.

[132] M. Laurence, "Introduction to octasic asynchronous processor technology," in *Proc. 18th IEEE International Symp. on Asynchronous Circuits and Systems*, 2012, pp. 113–117.