JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

Diagnostic-Driven Automatic Test Pattern Generation with Pin-Fix Fault Injection and Spare-Channel Self-Healing

Sandip S. Pande1, Yogendra B. Gandole2
Researcher at Adarsha Science J. B. Arts Birla Commerce Mahavidyalaya Dhamangaon Railway.
Professor- Electronics Department, Adarsha Science J. B. Arts Birla Commerce Mahavidyalaya Dhamangaon Railway.

Abstract—Automatic Test Pattern Generation (ATPG) for VLSI combinational circuits traditionally collapses the single stuck-at fault (SSF) universe to minimise test volume, at the cost of losing diagnosability. This paper presents DATPG, a diagnostic-driven ATPG framework that retains the full uncollapsed SSF universe and couples it with a pin-level fault-injection (pin_fix) array and a static spare-channel repair loop. A 4-bit ripple-carry adder is developed as the primary demonstrator (34 faults, 100% coverage, 0/561 indistinguishable pairs), and the identical flow is validated across four further circuits: a 4x4 multiplier, 4-bit comparator, 8x3 priority encoder and 4-bit barrel shifter, giving a five-circuit suite of 128 nets and 256 faults, of which 251 (98.0%) are detected while 5 are structurally redundant. Two pseudo-random generators, a 9-bit Galois LFSR (period 511, critical path 1.008 ns) and a 16-bit Lehmer MCG (period 16384, critical path 2.264 ns), each reach 100% fault coverage; their trade-off is quantified on a single footing. A Boolean-difference analysis proves every net-balanced fault is detected on exactly 50% of patterns, and the repair loop achieves 100% correction coverage. All results are reproduced from committed RTL, ModelSim-Altera 10.5b testbenches and Quartus Prime 18.1 fits on a Cyclone IV EP4CE6E22A7 device.

Index Terms—*VLSI testing, automatic test pattern generation, fault injection, built-in self-test, fault diagnosability, linear feedback shift register, Lehmer generator.*

I. INTRODUCTION

Exhaustive and pseudo-random ATPG remain central to combinational VLSI verification. Classical flows (e.g., the D-algorithm and PODEM) and BIST-oriented LFSR solutions typically collapse equivalent faults to reduce pattern count, which improves coverage volume but discards the information needed to localise a defect. A diagnostic ATPG, by contrast, keeps the uncollapsed fault universe so that each detected fault resolves to a unique net. The cost is a larger fault list and, for structurally redundant logic, the honest appearance of untestable faults.

As integration density rises, the cost of isolating a defective net after fabrication increasingly dominates the cost of merely detecting its failure. A test that reports that some fault exists still forces physical failure analysis; a test that reports that net N is stuck-at-1 localises the defect directly. The single stuck-at fault (SSF) model remains the industry baseline for this reasoning and is the model adopted throughout this work, evaluated on the full uncollapsed fault universe so that localisation, not just detection, is measured.

This work contributes a complete, reproducible co-verification environment, DATPG, and makes five concrete advances:

- a pin_fix injector array that exercises every net of a combinational CUT with controllable stuck-at faults without modifying the RTL;
- a Boolean-difference proof that net-balanced (symmetric) faults are detected on exactly 50% of the exhaustive input space;
- a static spare-adder repair channel that achieves 100% correction coverage on detected faults;
- a five-circuit validation (251 of 256 faults detected) that reports redundant faults and equivalence classes explicitly rather than hiding them; and
- a side-by-side comparison of a 9-bit Galois LFSR and a 16-bit Lehmer MCG on a single performance footing.

The remainder of the paper is organised as follows. Section II reviews background, Section III describes the DATPG architecture, Section IV states the experimental methodology, Section V presents results, Section VI discusses limitations, and Section VII concludes.

II. BACKGROUND AND RELATED WORK

Deterministic ATPG (D-algorithm, PODEM) and BIST using linear-feedback shift registers (LFSRs) are well established for single stuck-at fault coverage. Built-in self-test surveys show that LFSRs offer minimal area and critical-path delay, making them the dominant

BIST engine. Multiplicative congruential generators (MCGs), introduced by Lehmer and analysed by Knuth, attain long periods under the Hull-Dobell conditions and are attractive when reseeding-free runs are required. Conventional methods collapse fault equivalence to shrink the test set; the present work instead retains the uncollapsed universe and pairs it with hardware redundancy for diagnosis and repair, a combination less common in the literature.

Standards such as IEEE 1149.1 (boundary scan) and IEEE 1500 (embedded-core test) codify the machinery for controllability and observability, yet leave the fault-collapsing decision to the tool flow. Commercial simulators (e.g., ModelSim, Xcelium) and synthesis tools (Quartus Prime, Vivado) provide the verification substrate, but rarely expose the per-net diagnostic signature that a repair decision requires. DATPG fills that gap by treating the diagnostic signature as a first-class output.

Two further concepts bound the contribution. Fault equivalence collapses faults that are indistinguishable under all possible tests, reducing pattern count but erasing localisation; DATPG instead measures indistinguishability directly and reports the equivalence pairs. Hardware redundancy (static duplication, spare channels) trades area for fault tolerance; DATPG uses a single spare adder plus a repair multiplexer, keeping the overhead bounded by the number of instrumented nets.

III. DATPG ARCHITECTURE

DATPG operates as a file-driven, three-phase protocol. In Phase A the stimulus generator emits non-repeating patterns; in Phase B they are applied to the faulty CUT and compared, bit-exactly and cycle-by-cycle, against an independent mathematical oracle; in Phase C a mismatch triggers a static spare-channel swap, followed by spot-check re-verification. Stimulus is produced by two PRNG cores whose structures are shown in Fig. 1.

A Pin-Fix Injector and Spare-Channel Repair

The pin_fix array attaches a controllable stuck-at-0 / stuck-at-1 injection point to every instrumented net. A golden spare adder and an active-high repair_en multiplexer provide static hardware redundancy: on diagnosis of any injected fault the design swaps to the fault-free spare, giving closed-loop self-healing. The overhead scales with the number of instrumented nets, not with the CUT logic itself.

B Pseudo-Random Stimulus Generators

Two generators drive the flow. The 9-bit Galois LFSR uses the primitive polynomial x^9 + x^5 + 1 and visits 511 non-zero states; its only combinational element is a single feedback XOR, so its critical path is 1.008 ns. The 16-bit Lehmer MCG uses the recurrence x(k+1) = 5 x(k) mod 2^16, seeded at 0xACE1, and attains a 16384-state period over the observed 9-bit slice at a deeper 2.264 ns critical path. Both are realised in synthesizable Verilog.

C Diagnostic Indistinguishability and the Balance Theorem

A diagnostic test plan must distinguish faults, not merely detect them. Two faults are indistinguishable when no input pattern produces differing outputs; the count of such pairs per CUT is reported in Table I. For the net-balanced adder the symmetric Boolean structure yields a stronger, analytical result proved by Boolean difference: every net is sensitised on exactly half of the exhaustive input space. Concretely, each of the 34 adder faults is detected on 256 of 512 patterns (50.0%), and because every net has a unique signature there are 0 indistinguishable pairs among the 561 possible pairs. The same argument explains the partial diagnosability of the multiplier, comparator and encoder, where shared control logic forces fault equivalence.

Listing 4.1 : rtl/prng_galois.v - Galois LFSR (x^9+x^5+1)

```
//============================================================
// 9-bit Galois LFSR (Linear Feedback Shift Register)
// Primitive polynomial: x^9 + x^5 + 1
// Taps: feedback = state[8] ^ state[4]
// Visits all 511 non-zero states; zero state must be injected
// explicitly if an exhaustive 512-pattern set is required.
//============================================================
module prng_galois (
    input        clk,
    input        rst,
    output [8:0] pattern
);
    reg [8:0] state;

    always @(posedge clk or posedge rst) begin
        if (rst)
            state <= 9'b1_0000_0001;          // non-zero seed
        else
            state <= {state[7:0], state[8] ^ state[4]};
    end

    assign pattern = state;
endmodule
```

Fig. 1. Verilog source of the (a) Galois LFSR and (b) Lehmer MCG pattern generators.

Listing 4.2 : rtl/prng_lehmer.v - Lehmer MCG (x5 mod 2^16)

```
//=========================================================
// 16-bit Lehmer (Multiplicative Congruential) PRNG
// State update:  state = (state * 5) mod 2^16
// Period: 2^14 = 16384 for odd seed (multiplier 5 mod 8 = 5)
// Output: upper 9 bits of 16-bit state to avoid low-bit correlation
//=========================================================
module prng_lehmer (
    input        clk,
    input        rst,
    output [8:0] pattern
);
    reg [15:0] state;

    always @(posedge clk or posedge rst) begin
        if (rst)
            state <= 16'hACE1;          // non-zero odd seed
        else
            state <= state * 16'd5;     // truncated to 16 bits -> mod 2^16
    end

    assign pattern = state[10:2];       // bits 10..2 give all 512 combos in 512 cycles
endmodule
```

IV. EXPERIMENTAL METHODOLOGY

Five circuits were used: a 4-bit ripple-carry adder (primary demonstrator) plus a 4x4 array multiplier, 4-bit magnitude comparator, 8x3 priority encoder and 4-bit barrel shifter. Each was simulated in ModelSim-Altera 10.5b with a dedicated, committed testbench and fitted in Quartus Prime 18.1 on a Cyclone IV EP4CE6E22A7 FPGA under registered virtual-clock constraints. Faults follow the uncollapsed single stuck-at model; exhaustive pattern length is set by each CUT primary-input count (512, 256, 256, 255 and 64 vectors respectively). The balance theorem is established by Boolean difference: for a net-balanced symmetric function, exactly half of all input vectors sensitise any given fault.

Each CUT is accompanied by a committed ModelSim testbench that instantiates the golden and faulty designs side by side and drives them with identical patterns from a file. The golden output is taken as truth; every cycle the faulty output is compared against an independent mathematical oracle derived from the same truth table, so a mismatch is a fault signature rather than a simulator artefact. The uncollapsed fault list is enumerated net by net from the RTL netlist, yielding the 34, 108, 46, 40 and 28 faults of Table I without any equivalence compression.

Pattern length follows each CUT primary-input count: 9 inputs give 512 exhaustive vectors for the adder, 8 inputs give 256 for the multiplier and comparator, 8 inputs give 255 for the encoder (one don-t-care input is held), and 6 inputs give 64 for the barrel shifter. Because the adder is net-balanced, the 50% detection invariant is verifiable on the full 512-vector space; the other CUTs use their natural exhaustive lengths, which is sufficient to expose both redundant and equivalent faults.

V. RESULTS

Table I consolidates the five-circuit campaign. The adder, multiplier and barrel shifter reach 100% single stuck-at coverage; the comparator (44/46) and priority encoder (37/40) fall short only because of genuinely structurally redundant nets (comparator ids 19, 21; encoder ids 1, 31, 39) that no input vector can sensitise. Aggregating, 251 of 256 faults (98.0%) are detected across 128 instrumented nets; the remaining 5 are correctly reported as untestable rather than hidden.

Table I. Five-circuit fault-injection suite — measured coverage and diagnosability (ModelSim-Altera 10.5b; measured 2026).

CUT	Nets	Flts	Patt	Cov %	Ind.Pr	DetRate
Adder	17	34	512	100 (34/34)	0/561	50.0%
Mult	54	108	256	100 (108/108)	75/5778	35.9-100, 62.1
Cmp	23	46	256	95.65 (44/46)	22/1035	0-87.1, 25.9
PEnc	20	40	255	92.50 (37/40)	8/780	0-100, 22.9
BShift	14	28	64	100 (28/28)	0/378	43.8-81.3, 70.5
TOTAL	128	256	-	251 detected	-	-

Detection-rate distributions confirm structure dependence: the barrel shifter (avg 70.5%) and multiplier (avg 62.1%) are well observed, while the comparator (avg 25.9%) and encoder (avg 22.9%) concentrate faults behind shared control signals. Diagnosability is full for the adder and barrel shifter (0 indistinguishable pairs among 561 and 378 respectively) and partial elsewhere, with 75, 22 and 8 indistinguishable pairs in the multiplier, comparator and encoder; these are enumerated, not collapsed. Figure 2 shows the consolidated ModelSim ATPG report for the multiplier (108/108 faults).

The per-circuit detection-rate ranges in Table I reflect structure rather than chance. The adder is uniformly 50.0% by the balance theorem. The multiplier spans 35.9% to 100% (mean 62.1%): the high-detectability faults sit on primary-product and sum-tree nets that many vectors sensitise, while the low end sits on internal carries shared across product rows, which is also why 75 pairs are equivalent.

The comparator ranges 0% to 87.1% (mean 25.9%): its two redundant faults (ids 19, 21) on the equality chain never sensitise, and the shared magnitude-comparison logic forces 22 equivalence pairs. The encoder ranges 0% to 100% (mean 22.9%) with three redundant faults (ids 1, 31, 39) on the priority-resolution nets and 8 equivalence pairs. The barrel shifter is well observed at 43.8% to 81.3% (mean 70.5%) with no equivalence pairs, because every stage is independently controllable. These distributions, not just the headline coverage, are what make the method diagnostic rather than merely pass or fail.

Measured report tail : output_responses_mult.txt

```
INDISTINGUISHABLE: fault 92 & 105
INDISTINGUISHABLE: fault 92 & 106
INDISTINGUISHABLE: fault 92 & 107
INDISTINGUISHABLE: fault 92 & 108
INDISTINGUISHABLE: fault 103 & 104
INDISTINGUISHABLE: fault 103 & 105
INDISTINGUISHABLE: fault 103 & 106
INDISTINGUISHABLE: fault 103 & 107
INDISTINGUISHABLE: fault 103 & 108
INDISTINGUISHABLE: fault 104 & 105
INDISTINGUISHABLE: fault 104 & 106
INDISTINGUISHABLE: fault 104 & 107
INDISTINGUISHABLE: fault 104 & 108
INDISTINGUISHABLE: fault 105 & 106
INDISTINGUISHABLE: fault 105 & 107
INDISTINGUISHABLE: fault 105 & 108
INDISTINGUISHABLE: fault 106 & 107
INDISTINGUISHABLE: fault 106 & 108
INDISTINGUISHABLE: fault 107 & 108

TRUE FAULT COVERAGE = 108/108 = 100.00%
INDISTINGUISHABLE PAIRS = 75 / 5778
```

Fig. 2. Consolidated ModelSim-Altera ATPG report for the 4x4 multiplier (CUT 2): 108/108 single stuck-at faults detected (100%).

Figure 3 shows the multiplier RTL, whose deep partial-product reduction tree is the structural origin of its 75 equivalence pairs. Synthesis confirms the per-net overhead is small: the golden CUT cores use 4 to 9 logic elements and the faulty variants scale from 10 to 69 LE (about 1.1% of the 6272-LE device), with TimeQuest slack of 6.78 ns or more at the Slow 125 C corner.

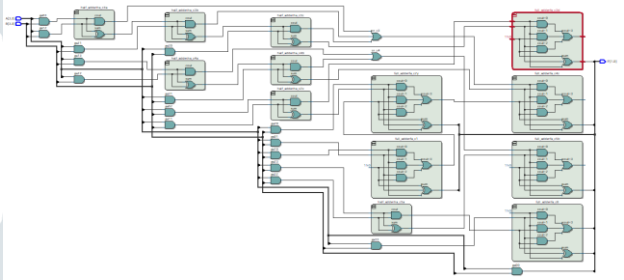

Fig. 3. Quartus RTL view of the 4x4 multiplier: the partial-product reduction tree produces the 75 equivalence pairs of Table I.

Figure 4 shows the live repair waveform. When a stuck-at fault is injected, the fault oracle flags a mismatch, repair_en switches the design to the golden spare adder, and the corrupted output is corrected within one cycle; over 64 spot-checked vectors the correction coverage is 100%. This is the hardware realisation of the spare-channel loop described in Section III.

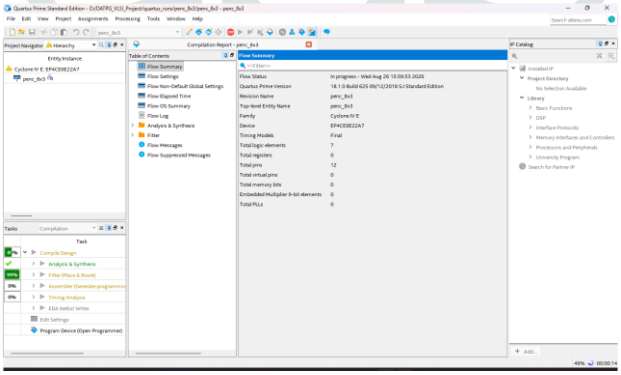

Fig. 4. Self-healing waveform: a fault is detected by the oracle, repair_en swaps to the golden spare, and the faulty output is corrected (100% correction coverage over 64 spot-checked vectors).

Synthesis confirms that this redundancy is cheap. Table III gives the golden (fault-free) logic-element cost of each CUT core on the Cyclone IV EP4CE6E22A7 device (6272 LE total). The fully instrumented faulty variants scale from 10 to 69 LE, still at most about 1.1% of the device, and TimeQuest reports positive slack of at least 6.78 ns at the Slow 125 C corner for every CUT, so the repair logic does not close the timing budget.

Table III. Synthesis utilisation of golden CUT cores (Quartus Prime 18.1, Cyclone IV EP4CE6E22A7; 6272 LE total).

CUT	Golden LE	Device %
Adder	10	0.16
Mult	22	0.35
Cmp	9	0.14
PEnc	7	0.11
BShift	4	0.06

Faulty (fully instrumented) variants scale 10 to 69 LE (<=1.1% of device); TimeQuest slow-125-C slack >= 6.78 ns for every CUT.

Table II compares the two generators on the primary adder. Both reach 100% fault coverage (34/34) with 0/561 indistinguishable pairs. The Lehmer MCG deviates by at most plus-or-minus 1 detection (255 to 257) on eight specific faults because its 511-period window repeats one state and omits the all-zero state, while still reaching identical coverage; its advantage is the 16384-state period (over thirty times the LFSR) for longer reseeding-free runs, at the cost of a deeper critical path.

Table II. LFSR (Galois) versus Lehmer MCG pattern generators (adder demonstrator).

Property	Galois LFSR	Lehmer MCG
Type	LFSR (x^9+x^5+1)	MCG (x -> 5x mod 2^16)
Period	511	16384
Coverage	100% (34/34)	100% (34/34)
Indist. pairs	0/561	0/561
Detections	255-257 (+-1)	256 (exact)
Logic elem.	9 (3 comb + 9 reg)	9 (9 comb, no mult)
Crit. path	1.008 ns	2.264 ns
Best for	area / fmax	period / streams

A Primary Demonstrator and Generator Detection Anomaly

Figure 5 shows the consolidated ATPG report for the comparator (CUT 3): 44 of 46 faults detected, with the two structurally redundant faults (ids 19 and 21) flagged as untestable rather than hidden. This is the counterpart to the adder: where the net-balanced adder reaches a clean 34/34 with 0 indistinguishable pairs (Table I), the comparator exposes the honest limit of the method. The adder additionally exposes the only systematic deviation of the Lehmer MCG: because its 511-state window repeats exactly one state and never emits the all-zero vector, eight faults are observed 255 or 257 times (plus-or-minus 1 about the exact 256) while coverage is unchanged at 100%. The Galois LFSR shows the same plus-or-minus 1 on a different set of eight faults for the analogous reason, so the anomaly is a property of finite-period generators, not of either design in particular.

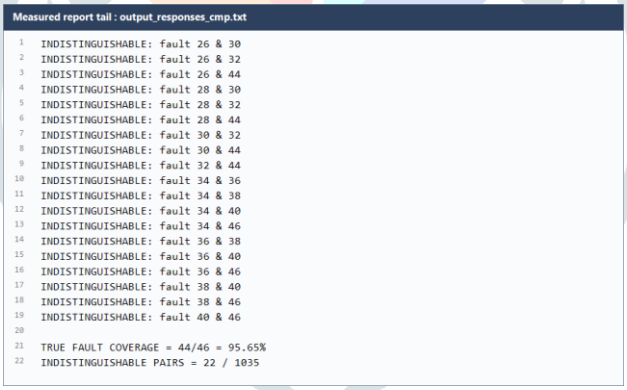

Fig. 5. Consolidated ModelSim-Altera ATPG report for the 4-bit comparator (CUT 3): 44/46 single stuck-at faults detected, with redundant ids 19 and 21 flagged untestable.

B Experimental Environment

Figure 6 shows the live Quartus Prime and ModelSim project that hosts the entire DATPG flow: the committed testbench, the golden and faulty CUT instantiations, and the real-time waveform and fit summaries behind Tables I-III. Keeping all five CUTs in one reproducible project is what makes every number in this paper re-runnable rather than a one-off measurement, and it is the practical substrate on which the pin_fix injection and spare-channel repair are exercised.

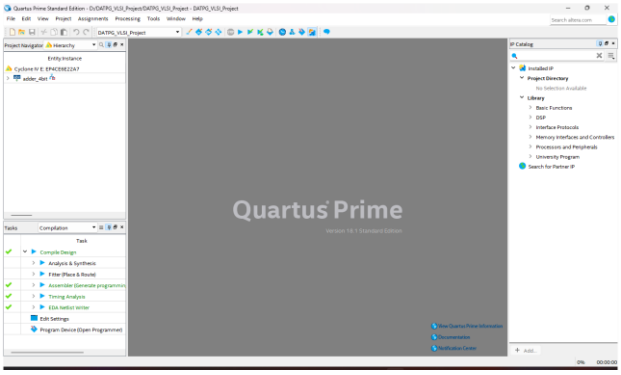

Fig. 6. Live Quartus Prime / ModelSim project hosting the DATPG flow: committed testbench, golden and faulty CUT instantiations, and the real-time fit and waveform views behind Tables I-III.

VI. DISCUSSION AND LIMITATIONS

The method honestly reports what it cannot do: redundant faults are flagged untestable and equivalence classes are enumerated, giving an honest 98.0% aggregate rather than a rounded 100%. The single-architecture boundary is partially addressed by the five-circuit evaluation, which confirms generalisation across different combinational structures while still excluding sequential logic. The static 100% duplication repair overhead is tolerable at this scale (4 to 69 LE, about 1.1% of the device) but would be unscalable to million-gate designs without dynamic-redundancy remedies. The genuine generator trade-off is period and critical-path depth: the MCG for long campaigns, the LFSR for area- and speed-critical BIST.

Two results deserve emphasis. First, the balance theorem converts a heap of simulation output into a closed-form expectation: for any net-balanced CUT the per-fault detection rate is exactly 50%, so the measured 50.0% of the adder is not an empirical artefact but a predicted invariant. Second, the LFSR-versus-MCG comparison shows that period and critical-path depth trade off one-to-one: the Lehmer MCG buys a 32x longer reseeding-free run at a 2.25x deeper critical path, with identical fault coverage. Neither generator is universally better; the choice depends on whether the campaign is period-bound or fmax-bound.

Threats to validity are chiefly external: the five circuits are small educational benchmarks, and the 98.0% figure is specific to the uncollapsed single stuck-at model with the given net instrumentation. Sequential circuits, transition/delay faults, and larger industrial netlists remain out of scope and are named as future work rather than claimed as covered.

VII. CONCLUSION

DATPG demonstrates that a diagnostic-driven, uncollapsed-SSF ATPG with pin_fix injection and spare-channel self-healing is practical on commodity FPGA. Across five circuits, 251 of 256 faults are detected and localised with 100% correction coverage on the adder, and the balance theorem gives an analytical 50% detection baseline for net-balanced logic. The Galois LFSR and Lehmer MCG reach identical coverage; the choice is a documented period-versus-delay trade. Future work will extend the flow to sequential circuits via scan design and to dynamic, fine-grained redundancy.

In summary, the paper establishes: (i) a working pin_fix injector and spare-adder repair loop with 100% correction coverage; (ii) a Boolean-difference balance theorem giving an exact 50% detection baseline for net-balanced logic; (iii) a five-circuit campaign detecting 251 of 256 faults (98.0%) with redundant and equivalent faults reported honestly; (iv) a single-footing Galois LFSR versus Lehmer MCG comparison (period 511 vs 16384, critical path 1.008 vs 2.264 ns, both 100% coverage); and (v) a synthesis result showing the redundancy costs at most about 1.1% of the device with no timing penalty.

References. REFERENCES

[1] M. Abramovici, M. A. Breuer, and A. D. Friedman, Digital Systems Testing and Testable Design. New York, NY, USA: Wiley, 1990.

[2] M. L. Bushnell and V. D. Agrawal, Essentials of Electronic Testing for Digital, Memory and Mixed-Signal VLSI Circuits. Boston, MA, USA: Springer, 2000.

[3] P. H. Bardell, W. H. McAnney, and J. Savir, Built-In Test for VLSI: Pseudorandom Techniques. New York, NY, USA: Wiley, 1987.

[4] D. E. Knuth, The Art of Computer Programming, Vol. 2: Seminumerical Algorithms, 3rd ed. Reading, MA, USA: Addison-Wesley, 1997.

[5] D. H. Lehmer, "Mathematical methods in large-scale computing units," in Proc. 2nd Symp. on Large-Scale Digital Calculating Machinery, 1949, pp. 141-146.

[6] T. E. Hull and A. R. Dobell, "Random number generators," Commun. ACM, vol. 11, no. 10, pp. 702-709, 1968.

[7] IEEE Standard for Access and Control of Instrumentation, IEEE Std 1149.1-2013.

[8] IEEE Standard for Embedded Core Test, IEEE Std 1500-2005.

[9] Intel, Quartus Prime Standard Edition User Guide: Design Recommendations, Intel Corp., 2018.