P-PET: Partial Pseudo-Exhaustive Test for High Defect Coverage

Abdullah Mumtaz, Michael E. Imhof, Hans-Joachim Wunderlich
Institute of Computer Architecture and Computer Engineering
University of Stuttgart, Pfaffenwaldring 47, D-70569 Stuttgart, Germany
email: {mumtazah, imhof, wu}@iti.uni-stuttgart.de

***Abstract*—Pattern generation for embedded testing often consists of a phase generating random patterns and a second phase where deterministic patterns are applied. This paper presents a method which optimizes the first phase significantly and increases the defect coverage, while reducing the number of deterministic patterns required in the second phase.**

The method is based on the concept of pseudo-exhaustive testing (PET), which was proposed as a method for fault model independent testing with high defect coverage. As its test length can grow exponentially with the circuit size, an application to larger circuits is usually impractical.

In this paper, partial pseudo-exhaustive testing (P-PET) is presented as a synthesis technique for multiple polynomial feedback shift registers. It scales with actual technology and is comparable with the usual pseudo-random (PR) pattern testing regarding test costs and test application time. The advantages with respect to the defect coverage, N-detectability for stuck-at faults and the reduction of deterministic test lengths are shown using state-of-the art industrial circuits.

***Keywords*—BIST, Pseudo-Exhaustive Testing, Defect Coverage, N-Detect**

I. Introduction

Nearly three decades ago, pseudo-exhaustive testing has been proposed for increasing defect coverage without the limitations of specialized fault models [1–14]. In pseudo-exhaustive testing, each output function of a combinational circuit is tested exhaustively. As in general a single output depends on a subset of all primary inputs only, testing each output function exhaustively needs much less patterns than testing the circuit exhaustively.

For a primary output o of a combinational circuit C with the inputs I and outputs O, a *cone* is defined as the minimal sub-circuit containing all structural predecessors of o. The set k_o denotes all inputs connected to the output o, and its cardinality $|k_o|$ is called the *cone size*.

A pseudo-exhaustive test set T for C is a set of test patterns that includes an exhaustive test for each circuit cone. If the cones are tested in succession, the test application time is

$$|T| = \sum_{o \in O} 2^{|k_o|} \tag{1}$$

As some cones can be tested in parallel, the test time is limited by

$$2^w \leq |T| \leq |O| \times 2^w \tag{2}$$

where w represents the largest cone size in the circuit. The inequality above shows that the test set T increases exponentially with w. Testing the complete circuit pseudo-exhaustively is only practical if w does not exceed a certain limit.

The reasons why pseudo-exhaustive testing has been proposed were the limitations of fault models for reflecting real defects and for obtaining high defect coverage. With the advent of nano-electronic circuits and their problems of robustness, numerous defect mechanisms and unpredictable behavior, these reasons hold more than ever [15–18].

The reason why pseudo-exhaustive testing has not emerged as the dominating test strategy was its limitations in scaling for today's circuits. The increasing size of the maximum cones and the growing number of primary inputs and outputs made PET unfeasible, and different techniques had to be applied.

The standard approach is mixed mode testing where first pseudo-random patterns are applied, and afterwards deterministic test patterns are used for undetected faults. Pseudo-random testing can reach rather high defect coverage [19, 20] unless the circuit contains random pattern resistant areas.

For deterministic pattern generation, the stuck-at fault model is widely used for its simplicity, but it does not model the behavior of production defects completely [21]. Even achieving 100% stuck-at fault coverage does not guarantee the detection of all defects in a chip [22], thereby limiting the achievable defect coverage. The defect coverage can be enhanced by using the N-detect approach [20], where each single stuck-at fault is detected at least N times or the maximum number of times that fault can be detected. However, the size of the required deterministic test set grows significantly with the value N [23].

In this paper, we substitute the first phase of mixed-mode test, the pseudo-random pattern generation phase, by a novel scheme named partial pseudo-exhaustive test (P-PET). In P-PET, instead of all the circuit cones, only cones up to a given size MAX_{size} are tested pseudo-exhaustively. For the cones larger than MAX_{size}, the generated test set is not guaranteed to enumerate all of their input assignments, and behaves like pseudo-random patterns.

This scheme is based on the observation that modern circuits are well optimized for speed, and contain short paths. The resulting reduction in cone sizes means that a significant portion of the circuit can be tested pseudo-exhaustively. This implies that compared to PR testing

- testing this portion of the circuit pseudo-exhaustively results in a very high non-target fault coverage for the

978-1-4577-0152-8/11/$26.00

complete circuit.

- the pseudo-exhaustively tested part of the circuit offers the highest possible N-detectability, which significantly improves the overall defect coverage.
- a higher stuck-at fault coverage is achieved in the first phase of the mixed mode test scheme.
- the insertion of test points may further improve the fault coverage and defect coverage.
- the use of P-PET leads to a significantly lower number of deterministic pattern required in the second phase to reach a certain fault coverage.

The P-PET scheme consists of a synthesis algorithm for computing multiple feedback polynomials of a limited degree, a mapping for transforming the multiple scan chains problem to the single scan chain problem, and a multi-polynomial feedback shift register. The hardware of this register is directly taken from [24], and needs only negligible overhead compared to standard approaches. The synthesis algorithm for the multiple feedback polynomials is based on the theory published in [4], where a single polynomial of unbounded degree is computed for a single scan chain. Instead of this, we use multiple polynomials of bounded degrees, each of which covers a set of cones of a size $\leq MAX_{size}$. Dealing with all the cones $\leq MAX_{size}$ by a minimum number of polynomials is reduced to a set covering problem.

The rest of this paper is organized as follows: After describing of state of the art in the next section, the P-PET principle is presented in section III. Section IV explains the mapping of multiple scan chains to a single one, section V describes the used check for exhaustive enumeration, and the set covering heuristic is presented in section VI. Experiments performed on industrial circuits are discussed in section VII. We show and quantify the benefits in terms of increased fault coverage, high defect coverage, increased N-detectability, and the reduction of the deterministic patterns additionally required.

II. Previous work and state of the Art

A. Mixed-mode hardware schemes

Figure 1 shows the basic embedded test architecture using multiple scan chains: Self Test Using MISR and Parallel Shift register sequence generator (STUMPS).

The pattern generation is performed by a linear feedback shift register (LFSR) connected to the scan chains. The LFSR

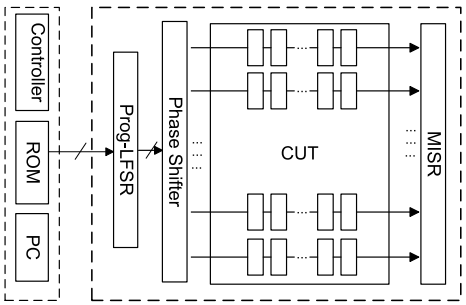


Fig. 1. Basic STUMPS Architecture.

implements a fixed feedback primitive polynomial of degree g and enumerates all input vectors of length g. In general, the pattern length $2^g - 1$ for this is not feasible, and only a subsequence is applied during pseudo-random testing.
Due to linear dependencies, some faults may not be detected by a single LFSR-sequence. By equipping the LFSR with programmability [24], several primitive polynomials with different linear dependencies can be used while the hardware overhead consists of some AND gates (fig. 2) and the mask $(h_{g-1}, h_{g-2}, ...h_0)$.

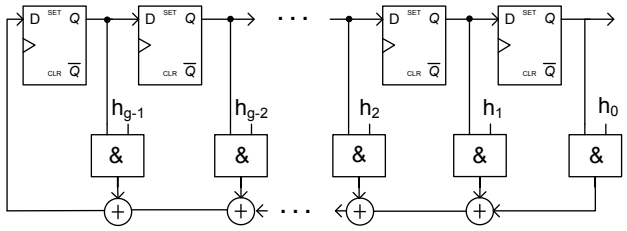


Fig. 2. Programmable LFSR [24].

The presented work adapts the programmable LFSR for the generation of pseudo-exhaustive test patterns, where the required primitive polynomials are calculated from the circuit structure. These polynomials are stored in the ROM (fig. 1) to update the programmable LFSR during pattern generation. Hence, in comparison to standard approaches (e.g. PR testing), the hardware overhead of the proposed P-PET is negligible and consist of a few AND gates and a small ROM. The approach is applicable for both embedded testing and for built-in self testing.

B. Pseudo-exhaustive pattern generation

Finding a pseudo-exhaustive test set with minimal length is known to be NP-complete [25]. However, several heuristics and techniques have been proposed to generate test sets with almost optimal sizes. Some of them include:

- *Syndrome-driver counters* [5].
- *Constant-weight code* based schemes [1].
- Hardware approaches like linear networks (XOR tree) based on linear codes [6] and linear sums [8].
- *Condensed LFSRs*, either based on linear codes [2] or cyclic codes [3].
- *LFSR and shift register* based schemes [4].

These schemes do not scale with today's circuits as they may produce test sets with extremely high pattern counts.

III. The P-PET Principle

An LFSR of length r and an exhaustive LFSR sequence of length $2^r - 1$ are uniquely defined by a characteristic polynomial or feedback polynomial p of degree r. Let $I := \{1, ..., n\}$ denote the primary inputs of the combinational circuit under test. For each output o, we identify its cone $k_o = \{i_1, ...i_s\} \subseteq I$ with the set of its primary inputs. If the feedback polynomial is primitive, the LFSR cycles through $2^r - 1$ different states and generates $2^r - 1$ different patterns.

We say, polynomial p tests cone k_o, $p \prec k_o$, if the $2^r - 1$ different patterns cover all $2^s - 1$ different assignments of k_o except the all zero vector which will be applied separately. Let $K_c := \{k_o | o \in O \wedge |k_o| \leq MAX_{size}\}$ be the set of all cones with MAX_{size} inputs at most. Our goal is to find a minimum set P of polynomials of degree $\leq MAX_{size}$, so that each cone is tested by at least one polynomial:

$$\forall\, k \in K_c\, \exists\, p \in P \quad p \prec k \tag{3}$$

The search for P is organized in several steps:

1) Map the P-PET problem for multiple scan chains to the P-PET problem for a single scan chain.
2) Implement an efficient procedure for checking $p \prec k$.
3) Look for an efficient procedure to solve the NP-complete set covering problem defined by (3).

IV. P-PET for multiple scan chains

Assume the scan elements of the circuit under test are organized into h scan chains, let the size of the largest scan chain be t. These scan chains can be treated as one big scan chain if the channel separation t_i between the different scan chains is at least t (fig. 3).

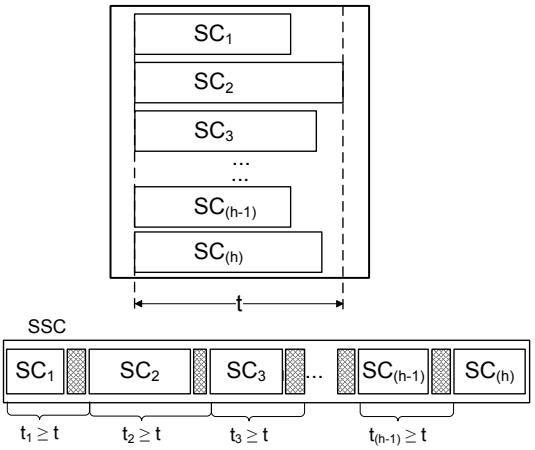


Fig. 3. Mapping of multiple scan chains

A carefully designed phase shifter [26] can be used to achieve this goal. In figure 3, h scan chains are mapped into a single scan chain SSC where two successive scan chains SC_i and SC_{i+1} are separated by t_i. However, this cannot be directly applied to multi-polynomial LFSRs as t_i also depends on the applied polynomial p. In this case, $t_i < t$ is possible, and a relation as seen in figure 4 may occur.

The figure contains two overlapping scanchains and presents two possible scenarios. First, if the inputs of a cone k_1 are mapped onto different positions in the unified scanchain, then $p \prec k_1$ is possible and has to be checked. In the second scenario, if some inputs of a cone (k_2 in figure 4) are mapped to the same positions in a virtual scan chain, these overlapping inputs will never receive different values. Hence, we can conclude without any further check that $p \prec k_2$ is not possible.

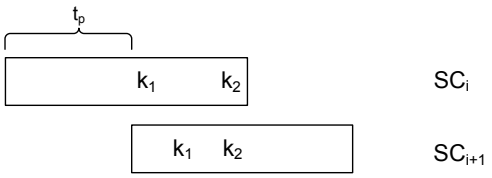


Fig. 4. Scan chain separation with and without conflict.

To distinguish the two cases, the position of each scan element $e \in SC_{i+1}$ in the virtual single scan chain is computed by

$$\left(\sum_{j=1}^{i} t_j + id(e)\right) \mod 2^r - 1 \tag{4}$$

where $id(e)$ is the index of e in SC_{i+1}.

From now on, we assume just a virtual single scan chain.

V. Checking for exhaustive enumeration

The naive check $p \prec k$ could be implemented by generating the $2^r - 1$ different patterns of p and validating that the 2^s different assignments of k are generated. However, this is extremely time consuming, and as this has to be done for a large set of polynomials it is not feasible. However, a much more efficient method is provided by Barzilai's theorem:

Theorem (Barzliai et al. [4])
Let $(a_\tau)_{\tau \geq 0}$ be a shift register sequence generated by a primitive feedback polynomial p of degree r. The set T:=$(a_0, ..., a_{r-1})$, $(a_1, ..., a_r)$, ...,$(a_{2^r-2}, a_0, ..., a_{r-2})$ is an exhaustive enumeration of the assignment of $(i_1, ..., i_s)$, if the remainder classes (X^{i_1} mod p), ..., (X^{i_s} mod p) over GF(2) are linearly independent.

Figure 5 shows an example where an LFSR with feedback polynomial $p = x^3 + x + 1$ is connected to a scan chain. The polynomial is primitive and repeats the sequence after $2^3 - 1$ cycles as highlighted in the figure.

To check if a cone $k_1 = \{0, 3, 4\}$, marked in the figure, receives every non zero sequence from p, the residue classes are calculated.

$$(x^0)\ mod\ (x^3 + x + 1) = (1)$$
$$(x^3)\ mod\ (x^3 + x + 1) = (x + 1)$$
$$(x^4)\ mod\ (x^3 + x + 1) = (x^2 + x)$$

As the three remainders are linearly independent, we have $p \prec k_1$. This can be verified by using figure 5 and initializing the LFSR to any random value. All unique $2^3 - 1$ nonzero sequences are applied at the desired positions after 11 cycles.

On the contrary, cone $k_2 = \{0, 4, 5\}$ fails the linear independency check. Hence, the polynomial p does not cover k_2, which can easily be verified by using figure 5.

Barzilai's theorem reduces the check of $p \prec k$ to s simple polynomial divisions and a check for linear dependency.

VI. Set Covering Heuristic

Let P_k be the set of primitive polynomials of degree k, let $f_p := \{k \in K_c / p \prec k\}$ be the set of all cones tested by a polynomial p. We want to find a set $L \subset P_{MAX_{size}}$ such that $K_c = \bigcup_{p \in L} f_p$ and $|L|$ is minimum.
The search procedure has two phases:

A) Reduction of K_c by removing redundant cones.
B) Iterative construction of polynomials until the new K_c is covered completely.

A. Reduction of the cone set

The bit sequence C, generated by an LFSR with polynomial $P(x) = \sum_{i=0}^{n} a_i x^i$ can be calculated by the following recurrence relation:

$$y_{m+n} = a_0 y_m + a_1 y_{m+1} + ... + a_{n-1} y_{m+n-1}, m \geq 0 \quad (5)$$

For each subsequence $c = c_0, c_1, ..., c_{n-1}$ in C, there is a $c' = c_{n-1}, c_0, ..., c_{n-2}$ where c' is a cyclic shift of c. This implies that in a scan chain, the position of a cone is irrelevant if the relative distances between the cone's inputs remain intact. Every displacement of this cone receives the same sequence. Therefore, we assume without loss of generality that the cones are shifted to the beginning of the scan chain, while keeping the relative distance between their inputs, and for each cone $k = \{i_1, ..., i_s\}$ we have $i_1 = 0$.

If we have two cones k_a, k_b with $k_a \subseteq k_b$, we can remove k_a, as any enumeration of k_b would also enumerate k_a. From now we assume that all redundant cones are removed from K_c.

Example: Suppose K_c contains two cones $k_2 = \{3, 7, 9\}$ and $k_3 = \{12, 16, 18, 25\}$. By shifting the cones at the beginning of the scan chain, the cones become: $k_2 = \{0, 4, 6\}$ and $k_3 = \{0, 4, 6, 13\}$. As $k_2 \subset k_3$ cone k_2 is identified as redundant and can be removed to get K_{oc} with only one cone k_3.

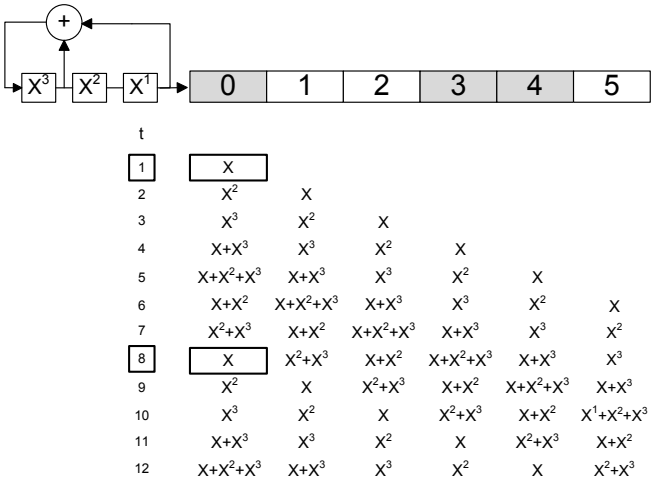


Fig. 5. LFSR sequence.

B. Iterative polynomial construction

K_c may still contain a large number of cones k, and checking pairwise $p \prec k$ can be rather time consuming for all primitive polynomials of a certain degree. Hence, we check $p \prec k$ only for cones of MAX_{size} and get a set of possible polynomials, each of them tests the largest number of big cones. From this set of possible polynomials, we select a polynomial which tests the largest subset of all smaller cones. The complete algorithm is found in figure 6.

1) Let $PP := \phi$
2) $K_c := \{k \in K \mid |k| \leq MAX_{size}\}$
3) $K_{oc} := \{k_b \in K_c | \nexists\, k_a \in K \quad k_b \subsetneqq k_a\}$
4) $MAX := MAX_{size}$
5) $K_d := \{k \in K_{oc} \mid |k| = MAX\}, idx = MAX$
6) If ($K_d := \phi$) $MAX = MAX - 1$, goto step 5.
7) $P_{cnd} := \{\exists p \in P_{idx} \mid$ p covers max cones of $K_d\}$
8) If ($P_{cnd} := \phi$), $idx = idx + 1$, goto step 7.
9) $K_s := K_{oc} - K_d$
10) Find $p_s \in P_{cnd}$ that covers max cones from K_s , add p_s to PP.
11) $K_{oc} := K_{oc} - \{k \in K_c \mid (p_s \prec k)\}$
12) $K_d := K_d - \{k \in K_d \mid (p_s \prec k)\}$, $P_{cnd} := P_{cnd} - p_s$
13) If ($K_d \neq \phi$) AND ($P_{cnd} \neq \phi$) goto step 9
14) If ($K_{oc} \neq \phi$) goto step 4.
15) Return PP.

Fig. 6. Algorithm: Calculate Covering Primitive Polynomials

The variable MAX is initialized to MAX_{size} and represents the largest cone size in K_{oc}. The set K_d is a subset of K_{oc} and contains cones of exactly the size MAX. If K_d is empty, MAX is decreased by one, and step 5 is repeated. A set of primitive polynomials of size idx (P_{cnd}) is found that covers a maximum number of cones from K_d. This set contains the solution candidates of the current iteration. As $|\mathrm{K}_d| \ll |\mathrm{K}_c|$, only a few cones are checked against P_{idx} to get P_{cnd}, where $|\mathrm{P}_{cnd}| \ll |\mathrm{P}_{idx}|$. This step immensely reduces the runtime of the algorithm and makes it scalable with the circuit size.

The set K_s contains the cones from K_{oc} which are not in K_d. The polynomial $p_s \in P_{cnd}$ that covers a maximum number of cones from K_s is the solution of the current iteration. The selected polynomial (p_s) is then added to the final solution set PP. The cones which are covered by p_s are removed from K_{oc} and K_d. If K_d is not empty and there are still candidate polynomials in P_{cnd}, we go to step 9. The algorithm continues unless K_{oc} is empty.

The final result is a set PP containing selected primitive polynomials that covers all the cones up to the given size MAX_{size}.
Every polynomial in PP is used as an LFSR feedback function and all possible unique patterns per polynomial are applied. The all zero pattern is applied separately. The number of patterns to be applied are calculated by using the following equation:

$$\sum_{i \in PP} 2^{PD_i} - |PP| + 1 + t \tag{6}$$

where PD_i is the degree of the polynomial and the 1 represents all zero pattern.

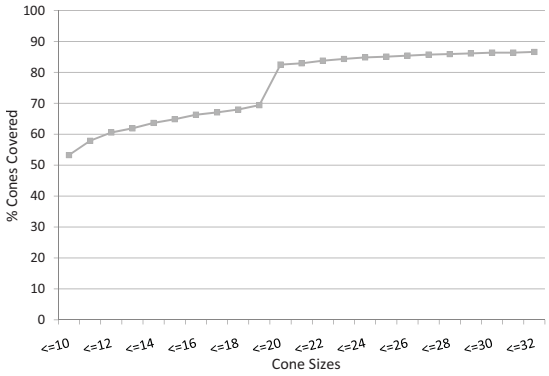
VII. Experimental Results

The experiments show the advantages of P-PET with and without circuit modification by test point insertion. The experiments are performed in two major parts. First, the advantages of P-PET are proven experimentally, while the second part evaluates the additional achievable gain if circuit modifications are possible. All experiments were conducted on full scan circuits kindly provided by NXP.

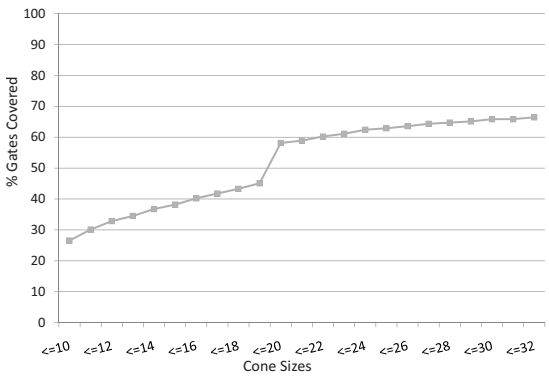
A. Partial Pseudo-exhaustive Testing

1) Analysis: P The cone distribution of a circuit determines the portion being P-PET testable. Figure 7 shows this distribution for p239k. In both graphs, the x-axis presents cones up to the size 32. For example, "$<= 16$" represents all cones up to the size 16. The largest cone contained by this circuit is of size 442. The y-axis in figure 7(a) represents the percentage of the cones covered. Figure 7(b) plots the percentage of gates covered by the cones up to a given size.

The results show that the majority of the cones is relatively small. More than half of the cones in the circuit p239k are of size 10 or smaller. This percentage of covered cones increases to almost 85% if cones up to a size of 22 are allowed.



(a) Cones Sizes vs Cones Covered



(b) Cones Sizes vs Gates Covered

Fig. 7. p239k: Achievable coverage by P-PET.

The portion of the circuit covered by these cones in terms of gates is presented in figure 7(b). It can be seen that the cones of limited sizes cover a significant portion of the circuit. Considering cones up to the size of 22 covers more than 60% of the circuit gates.

The maximum covered cone size (MAX_{size}) is user defined, circuit dependent and test time dependent. Analyzing the cone and gate coverages of the examined circuits showed an optimal trade-off between coverage and test time at 24, which will be used in all following experiments.

Table I presents the statistics of the considered industrial circuits. The first three columns show name, number of inputs and output (primary+pseudo-primary) of the circuits.

The name of the circuit roughly reflects the number of logic gates. The fourth column represents the percentages of the cones up to a size of 24, while the percentages of the gates that are covered by these cones are given in the last column.

Circuit	#(PI+PPI)	#(PO+PPO)	Cones (%)	Gates(%)
p35k	2912	2229	74.07	38.54
p45k	3739	2550	57.28	55.26
p89k	4632	4557	64.18	30.54
p100k	5902	5829	82.75	49.76
p141k	11290	10502	45.05	34.54
p239k	18692	18495	83.91	62.41
p259k	18713	18495	83.25	65.84
p279k	18074	17827	58.98	52.16
p378k	15732	17420	68.65	82.54
p418k	30430	29809	58.42	48.04
p483k	33264	32610	85.48	60.08
p533k	33373	32610	83.68	66.66

TABLE I
Cone and Gate Coverage ($MAX_{size} = 24$)

For the majority of the circuits, more than 60% of the cones have a maximum size of 24. For some circuits, this percentage is even beyond 80%. The last column shows that these relatively small cones correspond to almost half of the circuit.

2) Stuck-At Fault Coverage and N-Detectability: By using the proposed synthesis scheme, we calculated the required primitive polynomials for different circuits. The calculation is performed by using a single core of a 64-bit machine running at 2.4Ghz with 4GB RAM. Table II reports these results. The second column shows the number and degrees of the required polynomials. The third column presents the number of patterns generated from these polynomials and are calculated by using the equation 6. The time (in minutes) needed to calculate the required polynomials is shown in the last column. The numbers indicates the efficiency of the proposed heuristic, which can easily be adapted to take advantage of multi-core processors architectures to further reduce the calculation time.

For example circuit, p45k requires 1 primitive polynomial of degree 24. The circuit p35k does not have any cone between the size 17 and 24, the largest primitive polynomial required is of degree 16.

Circuit	Req-Poly	#Patterns	Time(Min)
p35k	$1 \times 2^{16} + 2 \times 2^{11}$	72544	8
p45k	1×2^{24}	16780955	4
p89k	$1 \times 2^{24} + 1 \times 2^{23}$	25170455	7
p100k	2×2^{24}	3356033	10
p141k	$2 \times 2^{24} + 2 \times 2^{23}$	50342935	123
p239k	$3 \times 2^{24} + 1 \times 2^{23}$	58738945	56
p259k	3×2^{24}	50350359	178
p279k	6×2^{24}	100681365	32
p378k	8×2^{24}	134233453	1418
p418k	5×2^{24}	83916506	115
p483k	$5 \times 2^{24} + 1 \times 2^{23}$	92307947	325
p533k	8×2^{24}	134251094	872

TABLE II
REQUIRED PRIMITIVE POLYNOMIALS AND PATTERN COUNT.

Firstly, experiments are performed to compare P-PET with PR testing while considering the single stuck-at fault model. For both the cases, we apply the same number of test patterns as reported in table II. For PR we used a single primitive polynomial of degree 128. A higher degree was chosen to make the comparison realistic, as usually a primitive polynomial of a degree higher than 24 is used for PR testing. The results are presented in table III where the column "Faults" shows the total number of stuck-at faults. The number of undetected faults for P-PET and PR cases are reported under the columns "PPET" and "PR" respectively. The difference between the two approaches is calculated by using the equation

$$\frac{PR - PPET}{PR} \times 100 \tag{7}$$

and shows the percentage of extra faults which are detected by P-PET as compared to PR case. The column "Dif" reports these results. For all the circuits, P-PET shows better results than PR in detecting hard pseudo-random resistive faults. This gives P-PET a significant benefit in the second half of mixed-mode testing where much less deterministic patterns are required as compared to PR case. Circuit p378k is an exception as it does not have any pseudo-random resistive fault.

Design	Faults	Undetected Faults		
		PR	PPET	Dif(%)
p35k	67988	27487	25476	7.32
p45k	71848	288	189	34.38
p89k	155794	14153	12745	9.95
p100k	166960	822	654	20.44
p141k	287552	9816	6735	31.39
p239k	455992	7180	5551	22.69
p259k	607536	10476	6024	42.50
p279k	493744	24690	19225	22.13
p378k	816274	0	0	0
p418k	688808	44536	40184	9.78
p483k	903348	22646	17669	21.97
p533k	1148846	23859	19376	18.79

TABLE III
SINGLE STUCK AT FAULT COVERAGE.

In the second set of experiments, N-detect is used as a defect coverage metric to evaluate the effectiveness of both the approaches. We keep the same experimental setup as in the previous set of experiments except that instead of a single detect, we target to detect a fault $N = 15$ times. The results are reported in table IV.

Design	Undetected (#Detect. < 15)			Deterministic Patterns		
	PR	PPET	Dif(%)	PR	PPET	Dif(%)
p35k	32626	30447	6.67	17005	16973	0.19
p45k	493	306	37.93	486	404	16.87
p89k	41113	34316	16.53	2791	2357	15.54
p100k	1246	922	26.00	688	592	13.95
p141k	35036	24145	31.08	2322	1985	14.51
p239k	9133	6181	32.33	1327	1037	21.85
p259k	35128	22968	34.61	1462	1294	11.61
p279k	53955	41245	23.55	7785	4475	57.48
p378k	23467	11628	50.44	638	397	37.77
p418k	62438	52661	15.65	2611	2172	16.81
p483k	43350	29811	31.23	1485	1018	31.44
p533k	48464	33317	31.25	1459	1193	18.23

TABLE IV
STUCK-AT FAULT COVERAGE TARGETING 15-DETECT.

In the first half of the table, the faults which are either undetected or detected less than 15 times are reported for both PR and P-PET cases. The percentage of extra faults which are detected by P-PET as compared to PR are calculated by using equation 7 and are presented in the column "Dif".

The numbers show the effectiveness of the proposed P-PET patterns during N-detection. For every circuit, as compared with PR testing, it detects a significant number of extra faults which fulfills the criteria of 15-detect.

This is clearly visible in the case of circuit p378k. As reported in table III, this circuit does not have any pseudo-random resistive faults and in both test cases, all the faults are detected. However, the superiority of P-PET test patterns is evident when the two test sets are evaluated for achievable defect coverages. The N-detect coverage of PR testing is much lower than with P-PET patterns, and the proposed approach detects more than 50% extra faults. These results indicate the fact that the P-PET testing not only gives advantages in detecting pseudo-random resistive faults, but is also more effective in N-detect and achieves a high defect coverage.

Deterministic patterns were calculated which target the undetected faults in table IV by using a commercial tool. They too are reported for both the cases in table IV. The columns "PR" and "PPET" show the required number of deterministic patterns. The column "Dif" represents the difference (in percent) between the two approaches. The high N-detectability of P-PET implies that it needs significantly less deterministic patterns as compared to PR testing, and requires 0.19% to 57% less storage.

As explained before, both approaches are comparable in terms of applying the deterministic patterns. Hence, with both schemes, the deterministic patterns can be applied efficiently by using compression schemes (e.g [27, 28]). However, due to lower pattern counts, significant savings in pattern storage are achieved for the P-PET case.

3) Non-Target Faults: The quality of test sets in terms of defect coverage is further evaluated by their effect on surrogate

(non-target) faults.

The experimental setup used previously for single stuck-at faults (table III) is retained except for the fault model. We consider four-way byzantine bridging faults as surrogate and randomly inject 40,000 faults per circuit. Their fault coverage is taken as an indicator of the un-modeled defect coverage achieved. The results are reported in table V where the columns 2 and 3 present the number of undetected bridging faults for P-PET and PR testing. In the last column, the percentage of the faults detected additionally by P-PET are reported.

Circuit	Un-Detected Brig faults		
	PR	PPET	Dif(%)
p35k	17743	15196	14.35
p45k	966	271	71.95
p89k	4025	1613	59.93
p100k	360	222	38.33
p141k	934	380	59.31
p239k	381	197	48.29
p259k	416	225	45.91
p279k	381	174	54.33
p378k	601	141	76.54
p418k	925	333	64.00
p483k	634	312	50.79
p533k	1099	667	39.31

TABLE V
P-PET: NON-TARGETED BRIDGING FAULTS COVERAGE.

The P-PET test detects all bridging faults inside an exhaustively tested cone, however, bridges between two cones may not be detected. For these inter-cones bridges, it behaves like pseudo-random patterns. Compared to the application of PR patterns, this increased circuit and gate coverage achieved by P-PET significantly improves the non-target fault coverage and thereby the defect coverage. For almost all the circuits (except p35k), P-PET detects more than 38% extra faults and reaches 76% for p378k. These results support the previous findings of N-detectability (table IV). As the proposed approach is more effective in detecting hard faults multiple times, it detects significantly more non-target faults.

B. Pseudo-Exhaustive Testing With Circuit Modification

When the circuit is tested pseudo-randomly, often not all of the faults are detected. One standard approach is the use of test points to improve the controllability and observability in a circuit [29, 30]. It is a part of the standard flow, which is used by the industry and supported by major industrial tools.

Test points insertion for pseudo-exhaustive (PE) testing has been studied thoroughly [7, 10, 31]. By using test points, all of the larger cones are reduced to at most predefined maximum size, and the complete circuit is made pseudo-exhaustively testable.

We conducted experiments to investigate the impact of test point insertion on the defect coverage for both P-PET and PR testing.

For these experiments, reported in table VI, we first insert test points for PET by setting the maximum allowed cone size to 24. Thereby it is guaranteed, that the modified circuit does not contain cones larger than size 24. The same amount of test points are inserted for PR case and are reported in column "#tp". Using the proposed algorithm, the required multiple primitive polynomials are found for the modified PE testable circuit. The number and degrees of the required polynomials are shown in column "Req-poly", while the fourth column presents the corresponding number of patterns. The bridging fault lists of the previous experiments (table V) are again used as surrogate faults.

In a PET modified circuit, every cone is tested pseudo-exhaustively. The pseudo-random test points insertion increases the controllability and observability of the circuit but the basic property of PR testing holds and detection of all the non-target faults still remains fortuitous. These facts are reflected in the results presented in table VI. For example circuit p279k, the table V reports the difference between P-PET and PR testing as 54%. When the test points are inserted for pseudo-exhaustive and pseudo-random testing, the difference increases to 76% (table VI).

Circuit	#tp	Req-poly	#Patterns	#Undet Brig faults		
				PR	PPET	Dif(%)
p35k	756	5×2^{24}	83888988	573	121	78,88
p45k	308	3×2^{24}	50335385	566	152	73.14
p89k	2167	7×2^{24}	117445138	251	49	80.48
p100k	1646	5×2^{24}	83891978	236	49	79.24
p141k	2473	6×2^{24}	100674581	215	74	65.58
p239k	3670	7×2^{24}	117459198	194	96	50.52
p259k	4007	8×2^{24}	134236434	220	101	54.09
p279k	4662	7×2^{24}	117458580	125	29	76.80
p378k	1820	8×2^{24}	134233453	534	85	84.08
p418k	4581	9×2^{24}	151028200	378	112	70.37
p483k	6887	9×2^{24}	151025366	402	60	85.07
p533k	7463	11×2^{24}	184582739	414	89	78.50

TABLE VI
REQUIRED POLYNOMIALS AND NON-TARGETED BRIDGING FAULTS COVERAGE WITH TEST POINTS INSERTION.

This is true for all the circuits and shows that the insertion of test points for pseudo-exhaustive testing has greater impact on non-target faults and defect coverage than doing it for pseudo-random case.

VIII. CONCLUSION

We introduced a new approach called partial pseudo-exhaustive test (P-PET), where cones up to a given size are tested pseudo-exhaustively. A synthesis technique for multiple polynomial feedback shift registers is presented which scales with actual technology and is comparable with the usual pseudo-random pattern testing regarding test costs and test application time.

With state-of-the art industrial circuits, we showed and quantified the advantages of the proposed scheme. The defect coverage was evaluated by using non-target fault and N-detectability metrics and the results show the substancial advantages achieved by P-PET. We also showed that the use

of P-PET results in a major reduction in the number of deterministic pattern required to reach a certain fault coverage.

The defect coverage can be further improved if circuit modifications are possible. We showed that the insertion of pseudo-exhaustive test points yields in significant benefits if compared to circuit modifications for the pseudo-random case.

IX. Acknowledgement

The authors would like to thank NXP for providing the industrial circuits.

References

[1] E. McCluskey, "Verification Testing - A Pseudoexhaustive Test Technique," *Proc. of IEEE Transactions on Computers*, pp. 541–546, 1984.

[2] L. Wang and E. McCluskey, "Condensed Linear Feedback Shift Register (LFSR) Testing A Pseudoexhaustive Test Technique," *Proc. of IEEE transactions on computers*, vol. 35, no. 4, pp. 367–370, 1986.

[3] ——, "Linear feedback shift register design using cyclic codes," *Proc. of IEEE Transactions on Computers*, vol. 37, no. 10, pp. 1302–1306, 1988.

[4] Z. Barzilai, D. Coppersmith, and A. Rosenberg, "Exhaustive generation of bit patterns with applications to VLSI self-testing," *Proc. of IEEE Transactions on Computers*, vol. 100, no. 32, pp. 190–194, 1983.

[5] Z. Barzilai, J. Savir, G. Markowsky, and M. Smith, "The weighted syndrome sums approach to VLSI testing," *Proc. of IEEE Transactions on Computers*, pp. 996–1000, 1981.

[6] N. Vasanthavada and P. Marinos, "An operationally efficient scheme for exhaustive test-pattern generation using linear codes," in *Proc. of IEEE International Test Conference*, 1985, pp. 476–482.

[7] I. Shperling and E. McCluskey, "Circuit segmentation for pseudo-exhaustive testing via simulated annealing," in *Proc. of IEEE International Test Conference*, 1987, pp. 58–65.

[8] S. Akers, "On the use of linear sums in exhaustive testing," in *Proc. of International Symposium on Fault-Tolerant Computing*, 1985, pp. 148–153.

[9] S. Hellebrand, H. Wunderlich, and O. Haberl, "Generating Pseudo-Exhaustive Vectors for External Testing," in *Proc. of IEEE International Test Conference*, 1990, pp. 670–679.

[10] S. Hellebrand and H. Wunderlich, "Tools and devices supporting the pseudo-exhaustive test," in *Proc. of Conference on European Design Automation*, 1990, pp. 13–17.

[11] D. Tang and L. Woo, "Exhaustive test pattern generation with constant weight vectors," *Proc. of IEEE Transactions on Computers*, pp. 1145–1150, 1983.

[12] J. Udell Jr, "Test set generation for pseudo-exhaustive BIST," in *Proc. of International Conference on Computer Aided Design*, 1986, pp. 52–55.

[13] H. Wunderlich and S. Hellebrand, "Generating pattern sequences for the pseudo-exhaustive test of MOS-circuits," *Proc. of International Symposium on Fault-Tolerant Computing*, pp. 36–41, 1988.

[14] ——, "The pseudo-exhaustive test of sequential circuits," in *Proc. of IEEE International Test Conference*, 1989, pp. 19–27.

[15] R. Aitken, "Nanometer technology effects on fault models for IC testing," *Proc. of IEEE Computer society press*, vol. 32, no. 11, pp. 46–51, 2002.

[16] J. McPherson, "Reliability challenges for 45nm and beyond," in *Proc. of ACM/IEEE Design Automation Conference*, 2006, pp. 176–181.

[17] F. Hopsch, B. Becker, S. Hellebrand, I. Polian, B. Straube, W. Vermeiren, and H.-J. Wunderlich, "Variation-Aware Fault Modeling," in *Proc. of IEEE Asian Test Symposium*, 2010, pp. 87–93.

[18] B. Becker, S. Hellebrand, I. Polian, B. Straube, W. Vermeiren, and H. Wunderlich, "Massive statistical process variations: A grand challenge for testing nanoelectronic circuits," in *Proc. of International Conference on Dependable Systems and Networks Workshops*, 2010, pp. 95–100.

[19] C. Tseng, S. Mitra, E. McCluskey, and S. Davidson, "An evaluation of pseudo random testing for detecting real defects," in *Proc. of IEEE VLSI Test Symposium*, 2001, pp. 404–409.

[20] S. Ma, P. Franco, and E. McCluskey, "An experimental chip to evaluate test techniques: Experiment results," in *Proc. of IEEE International Test Conference*, 1995, pp. 663–672.

[21] E. McCluskey and C. Tseng, "Stuck-fault tests vs. actual defects," in *Proc. of IEEE International Test Conference*, 2000, pp. 336–343.

[22] L. Wang, M. Mercer, T. Williams, and S. Kao, "On the decline of testing efficiency as fault coverage approaches 100%," in *Proc. of IEEE VLSI Test Symposium*, 1995, pp. 74–83.

[23] I. Pomeranz and S. Reddy, "Stuck-at tuple-detection: A fault model based on stuck-at faults for improved defect coverage," in *Proc. of IEEE VLSI Test Symposium*, 2002, pp. 289–294.

[24] S. Hellebrand, S. Tarnick, J. Rajski, and B. Courtois, "Generation of vector patterns through reseeding of multiple-polynomial linear feedback shift registers," in *Proc. of IEEE International Test Conference*, 1992, pp. 120–129.

[25] G. Seroussi and N. Bshouty, "Vector sets for exhaustive testing of logic circuits," *Proc. of IEEE Transactions on Information Theory*, vol. 34, no. 3, pp. 513–522, 1988.

[26] J. Rajski, N. Tamarapalli, and J. Tyszer, "Automated synthesis of phase shifters for built-in self-test applications," *Proc. of IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 19, no. 10, pp. 1175–1188, 2002.

[27] A. Hakmi, S. Holst, H. Wunderlich, J. Schlöffel, F. Hapke, and A. Glowatz, "Restrict encoding for mixed-mode BIST," in *Proc. of IEEE VLSI Test Symposium*, 2009, pp. 179–184.

[28] L. Li, K. Chakrabarty, and N. Touba, "Test data compression using dictionaries with selective entries and fixed-length indices," *Proc. of ACM Transactions on Design Automation of Electronic Systems*, vol. 8, no. 4, pp. 470–490, 2003.

[29] N. Tamarapalli and J. Rajski, "Constructive Multi Phase Test Point Insertion for Scan-based BIST," in *Proc. of IEEE International Test Conference.*, 1996, pp. 649–658.

[30] Y. Savaria, M. Youssef, B. Kaminska, and M. Koudil, "Automatic test point insertion for pseudo-random testing," in *Proc. of IEEE International Symposium on Circuits and Systems.*, 1991, pp. 1960–1963.

[31] R. Srinivasan, S. Gupta, and M. Breuer, "An efficient partitioning strategy for pseudo-exhaustive testing," in *Proc. of International Design Automation Conference*, 1993, pp. 242–248.