Bounds on Pseudo-Exhaustive Test Lengths

Rajagopalan Srinivasan, Sandeep K. Gupta,
and Melvin A. Breuer

CENG-96-09

Department of Electrical Engineering - Systems
University of Southern
Los Angeles, California 90089-2562
(213)740-4469

April 1996

Bounds on Pseudo-Exhaustive Test Lengths*

Rajagopalan Srinivasan†
Corporate CAD
SUN Microsystems
Menlo Park, CA 94025.

Sandeep K. Gupta and Melvin A. Breuer
Department of Electrical Engineering - Systems
University of Southern California
Los Angeles, CA 90089-2562.

Abstract

Pseudo-exhaustive testing involves applying all possible input patterns to the individual output cones of a combinational circuit. Based on our new algebraic results, we have derived both generic (cone-independent) and circuit-specific (cone-dependent) bounds on minimal length of test required so that each cone in a circuit is exhaustively tested. For any circuit with five or fewer outputs, and where each output has k or fewer inputs, we show that the circuit can always be pseudo-exhaustively tested with just 2^k patterns. We derive a tight upper bound on pseudo-exhaustive test length for a given circuit by utilizing the knowledge of the structure of the circuit output cones. Since our circuit-specific bound is sensitive to the ordering of the circuit inputs, we show how the bound can be improved by permuting these inputs.

Index Terms: *Bound, linear feedback shift register, pseudo-exhaustive testing, test length*

*This work was supported by the Advanced Research Projects Agency and monitored by the Federal Bureau of Investigation under Contract No. JFBI90092.

†This work was done while the author was at the University of Southern California.

1 Introduction

Exhaustive testing of a combinational circuit involves exercising the circuit with all possible input patterns. Exhaustive testing provides *comprehensive fault coverage* by ensuring detection of *all detectable combinational faults* in the circuit, where a **combinational fault** is a fault that does not manifest in any sequential behavior and is testable with a single input pattern. The test time associated with exhaustive testing increases exponentially with the number of inputs to the circuit. For circuits with a large number of inputs, exhaustive testing is very time consuming and *may not be practical.*

Pseudo-exhaustive testing of a combinational circuit involves exercising the individual output cones of the circuit with all possible input patterns [1]. An **output cone** consists of all logic that feeds an output. Pseudo-exhaustive testing provides *full coverage of stuck-at faults* that are considered likely in practice. The testing ensures detection of *all detectable combinational faults within the individual output cones* and *all detectable multiple stuck-at faults in the circuit.* The test time associated with pseudo-exhaustive testing is *typically much lower than* that for exhaustive testing.

Consider a combinational circuit with n inputs and m outputs. An output cone is said to *depend* on an input if there exists at least one path from that input to the output. The number of inputs on which an output cone depends is referred to as the **size** of the output cone. Let k be the maximum value among the sizes of the m output cones of the circuit. The value k is referred to as the **maximum cone size** of the circuit. The circuit can be characterized as an (n, m, k) circuit. Pseudo-exhaustive testing involves applying exhaustive tests to the m output cones. Generation of an optimal (minimum) pseudo-exhaustive test set for an (n, m, k) circuit is a hard problem. The pseudo-exhaustive test length is bounded below and above by 2^k and 2^n, respectively. *Estimation of realistic tight upper bound on the pseudo-exhaustive test length* is a very useful measure during the evaluation of test strategies for a circuit.

We have derived *provable upper bounds on pseudo-exhaustive test lengths.* The bounds can be classified into two categories, viz. **generic bounds** and **circuit-specific bounds.** Generic bounds are *independent of circuit output cone structure* and are derived using only the parameters n, m and k of the circuit. Circuit-specific bounds utilize the *structural information about the circuit output cones.* It is evident that circuit-specific bounds are tighter than generic bounds as they utilize more knowledge about circuit structure.

Autonomous linear feedback shift registers (LFSRs) [2] are widely used to generate pseudo-exhaustive test sets. LFSRs are characterized by their feedback connections represented as **polynomials**. For a non-zero initial state, the **period** of an LFSR is the number of states generated prior to repeating the initial state. An LFSR with n stages is said to be of **maximal length** if it has a period of $2^n - 1$ states. Maximal length LFSRs have **primitive** feedback polynomials, and are utilized by most pseudo-exhaustive test pattern generators. Maximal length LFSRs can be modified to generate the all-zero state. Pseudo-exhaustive test pattern generators (TPGs) that generate minimal length tests and/or utilize minimal hardware can be designed by utilizing knowledge about the circuit output cone dependencies. Examples of circuit-specific TPGs include LFSR/XORs [3, 4], LFSR/SRs [5, 6] and other structures proposed in [7, 8, 9, 10]. An LFSR/XOR structure is composed of a maximal length LFSR and an XOR network. An LFSR/SR structure is composed of a maximal length LFSR and a shift register (SR). These circuit-specific TPG structures are shown in Figure 1.

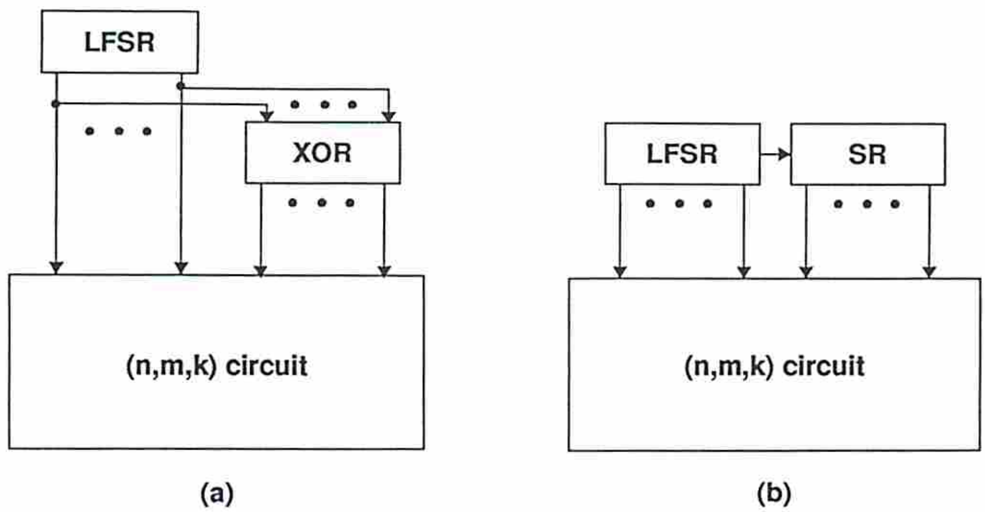

Figure 1: TPG structures (a) LFSR/XOR (b) LFSR/SR

A **test signal** is the unique sequence of binary values applied at a circuit input. A maximal length LFSR based on a primitive polynomial of degree k generates k linearly independent test signals from its k stages. An output cone of size k can be exhaustively tested with k linearly independent test signals. However, it may not be possible to concurrently test all the m output cones of an (n, m, k) circuit with k test signals because of conflicting input requirements among the cones. Thus the number of independent test signals required for pseudo-exhaustive testing of an (n, m, k) circuit is bounded below and above by k and n, respectively.

We have derived *new algebraic results on vector spaces* that are used in our bound computation. An upper bound on test length is computed as *the number of independent test signals that are sufficient* for pseudo-exhaustive testing of a given circuit. We have shown that *any circuit with five or less outputs, with each output being driven by k or less inputs, can always be pseudo-exhaustively tested with just 2^k patterns.* Previously this conclusion was known to be true *for all circuits having two or less outputs.* Additionally, we have derived *tight upper bounds on pseudo-exhaustive test lengths generated by circuit-specific TPG structures* such as LFSR/XORs and LFSR/SRs. These bounds are sensitive to the ordering of the circuit inputs. We have also developed an efficient method to permute the circuit inputs to obtain *the best improvement on the cone-dependent bounds.* The quality of these bounds are demonstrated by comparing them to existing bounds [3, 5].

The paper is organized as follows. Section 2 deals with algebraic results on vector spaces. The generic (cone-independent) bounds are derived in Section 3. Section 4 deals with the circuit-specific (cone-dependent) bounds and their improvements by allowing for the permutation of inputs. The conclusions are presented in the last section. The main paper contains only the *sketches for the proofs* of the theorems and lemmas. The detailed proofs are given in the appendix.

2 Algebraic Results

We shall present the definitions that are used in the bound computations. We define *vector space under modulo-2 addition operation (denoted as +) over the Galois field GF(2).* The modulo-2 addition operation satisfies the group properties such as commutativity, associativity and existence of additive inverses. The Galois field GF(2) forms a field with respect to modulo-2 addition and modulo-2 multiplication operations and satisfies all the standard axioms defined for a vector space.

Definition 1 *(Vector Space)*

- *A non-empty set S is a* **(vector) space** *over GF(2) if S is closed under modulo-2 addition.*
- *The* **(linear) span** *of a non-empty set B, denoted as $L(B)$, is the set of all linear combinations (modulo-2 addition) of elements in B.*
- *Any subset B of a vector space S is a* **basis** *of S if B consists of linearly independent elements and $L(B) = S$.*
- *The* **dimension** *of a vector space S spanned by a basis B equals $|B|$.*

Example 1 Consider the set $S = \{0, 1, x, 1+x, x^2, 1+x^2, x+x^2, 1+x+x^2\}$. The set S is closed under modulo-2 addition and hence forms a vector space. The set $B_1 = \{1, x, x^2\}$

consists of linearly independent elements and $L(B_1) = S$. Thus B_1 forms a basis of S and the dimension of S equals $|B_1| = 3$. The set $B_2 = \{1, 1+x, 1+x^2\}$ forms another basis of S. □

Definition 2 *(Operations between Vector Spaces)*

- *The* **direct sum** *of two spaces S_1 and S_2, denoted as $S_1 \oplus S_2$, equals the set $\{s_1 + s_2 \mid s_1 \in S_1,\ s_2 \in S_2\}$.*
- *The* **set union** *of two spaces S_1 and S_2, denoted as $S_1 \cup S_2$, equals the set $\{s \mid s \in S_1 \text{ or } s \in S_2\}$.*
- *The* **set intersection** *of two spaces S_1 and S_2, denoted as $S_1 \cap S_2$, equals the set $\{s \mid s \in S_1 \text{ and } s \in S_2\}$.*

Example 2 For the vector space $S = \{0, 1, x, 1+x, x^2, 1+x^2, x+x^2, 1+x+x^2\}$, consider two distinct subspaces $S_1 = \{0, 1, x, 1+x\}$ and $S_2 = \{0, 1, x^2, 1+x^2\}$ contained in S. The direct sum operation between S_1 and S_2, $S_1 \oplus S_2 = \{0, 1, x, 1+x, x^2, 1+x^2, x+x^2, 1+x+x^2\} = S$. The set union operation between S_1 and S_2 is $S_1 \cup S_2 = \{0, 1, x, 1+x, x^2, 1+x^2\} \neq S$ and is not a vector space. The set intersection operation between S_1 and S_2, $S_1 \cap S_2 = \{0, 1\}$, forms a subspace. □

Conventional algebraic theory deals with direct sum operation between vector subspaces. In contrast, our bound computations are based on set union and intersection operations between subspaces. We have derived some algebraic results regarding set union and intersection operations between subspaces and these results differ from the classical results in linear algebra. The following results characterize some properties of subspaces contained in a vector space that are used in the subsequent proofs. For convenience, only the sketches of proofs follow the lemmas and theorems and the detailed proofs are given in the appendix.

Lemma 1 *Consider a k-dimensional space S and any two distinct subspaces S_1 and S_2 of dimensions k_1 and k_2 contained in S. The set $S_1 \cap S_2$ is a subspace contained in S and consists of at least $\lceil 2^{k_1+k_2-k} \rceil$ elements.*

Proof Sketch: It can be proven that the set $S_1 \cap S_2$ is a subspace using the closure property. By choosing the intersection of properly chosen bases for S_1 and S_2 as the basis for $S_1 \cap S_2$, it can be shown that $S_1 \cap S_2$ consists of at least $\lceil 2^{k_1+k_2-k} \rceil$ elements. □

Corollary 1 *Consider a k-dimensional space S and any two distinct $(k-1)$-dimensional subspaces S_1 and S_2 contained in S. The set $S_1 \cap S_2$ is a $(k-2)$-dimensional subspace contained in S.*

Lemma 2 *A k-dimensional space is composed of at least $(2^i + 1)$ distinct subspaces of dimensions less than or equal to $(k-i)$, where $1 \leq i \leq (k-1)$.*

Proof Sketch: By considering the minimum overlaps between the subspaces, the result can be derived based on counting arguments. □

Example 3 Consider the three-dimensional vector space $S = \{0, 1, x, 1+x, x^2, 1+x^2, x+x^2, 1+x+x^2\}$ and all of its distinct two-dimensional subspaces $S_1 = \{0, 1, x, 1+x\}$, $S_2 = \{0, 1, x^2, 1+x^2\}$, $S_3 = \{0, x, x^2, x+x^2\}$, $S_4 = \{0, 1, x+x^2, 1+x+x^2\}$, $S_5 = \{0, x, 1+x^2, 1+x+x^2\}$, $S_6 = \{0, x^2, 1+x, 1+x+x^2\}$ and $S_7 = \{0, 1+x, 1+x^2, x+x^2\}$. It can be easily verified that $S_i \cup S_j \neq S \; \forall i, j$ and S is composed of at least three distinct two-dimensional subspaces (e.g. $S_1 \cup S_2 \cup S_4 = S$). □

Lemma 3 *Consider a k-dimensional space S and any three distinct $(k-1)$-dimensional subspaces S_1, S_2 and S_3 contained in S. Let $S^* = S_1 \cap S_2$. The subspace S_3 satisfies the relation $S_1 \cup S_2 \cup S_3 = S$ if and only if $S_1 \cap S_2 \cap S_3 = S^*$.*

Proof Sketch: It can be shown that the sets S^*, $S_1 - S^*$ (say T_1), $S_2 - S^*$ (say T_2) and $S - S_1 - S_2$ (say T_3) form equal sized disjoint partitions (cosets) of S. The smallest subspace that contains the elements of T_3 is of dimension $(k-1)$ and must also contain all the elements of S^*. Thus it can be shown that the subspace S_3 satisfies the relation $S_1 \cup S_2 \cup S_3 = S$ if and only if $S_1 \cap S_2 \cap S_3 = S^*$. The subspaces and the cosets are shown in Figure 2. □

Example 4 Consider the three-dimensional space S and all of its distinct two-dimensional subspaces S_1 through S_7 as given in Example 3. Let $S^* = S_1 \cap S_2 = \{0, 1\}$. It can be easily verified that S_4 is the only two-dimensional subspace that contains S^* and hence satisfies the relation $S_1 \cup S_2 \cup S_4 = S$. □

Lemma 1 gives a condition on the minimum overlap between any two subspaces contained in a k-dimensional space. Lemma 2 specifies the minimum number of distinct subspaces of smaller dimensions contained in a k-dimensional space. Lemma 3 states that the elements of a k-dimensional space S are not entirely covered by the elements of any two distinct $(k-1)$-dimensional subspaces contained in S. A unique third subspace of dimension no less

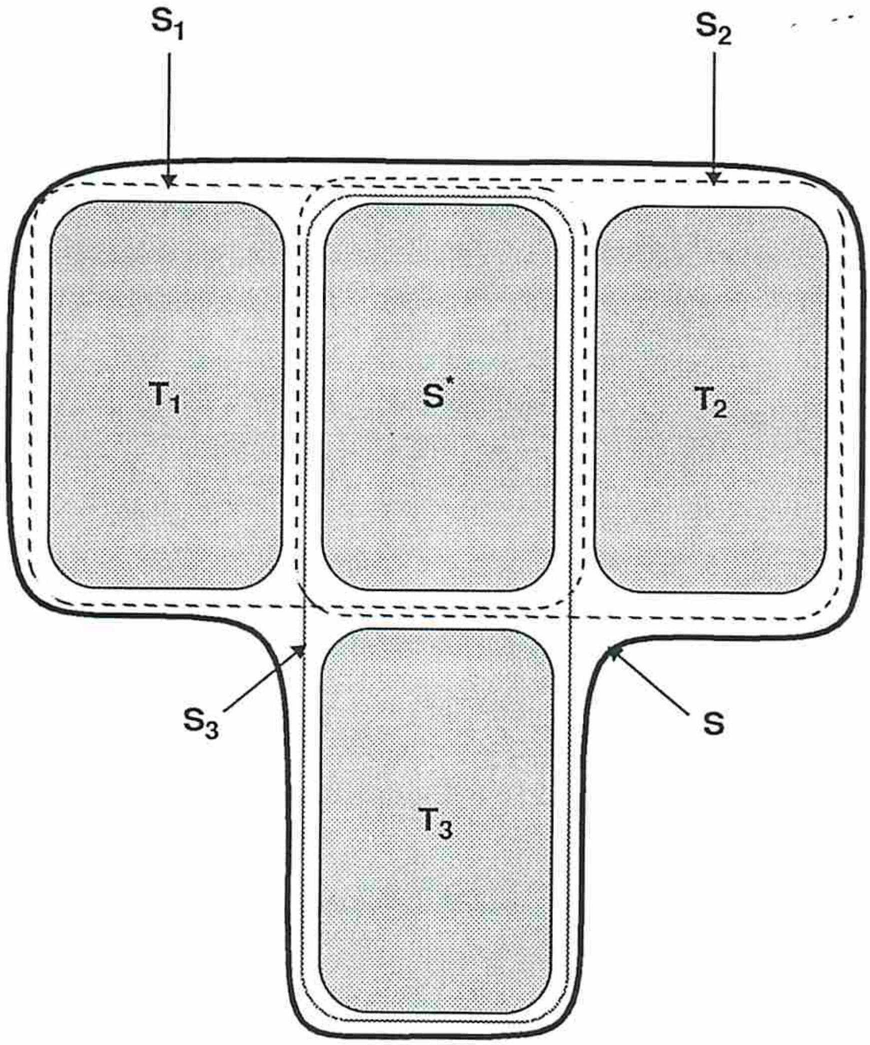

Figure 2: A vector space and its subspaces and cosets

than $(k-1)$ is required to cover all the elements of S. In fact Lemma 3 provides an outline for constructing the minimum number of distinct subspaces specified by Lemma 2.

The algebraic results described above form the basic building blocks in proving our results on both generic and circuit-specific bounds. Lemmas 1 and 2 are used in deriving both generic and circuit-specific bounds while Lemma 3 is used in the derivation of generic bounds.

3 Generic Bounds

For an (n, m, k) circuit, the computation of an upper bound on the pseudo-exhaustive test length involves determining the smallest number of independent test signals (say $k^* \geq k$) that are sufficient for pseudo-exhaustive testing of the circuit. We shall derive a few important cone-independent bounds on test lengths.

A set of 2^{k^*} distinct test signals can be obtained as linear combinations of k^* independent test signals. The distinct test signals are considered as distinct *residues*. The k^* independent test signals can be considered as a basis of a k^*-dimensional space and the residues can be considered as elements of this space. The k^* independent test signals can be generated using a k^* degree LFSR and linear combinations of these test signals can be obtained by an XOR network. Hence, if for a given (n, m, k) value a bound of k^* test signals is derived, then a TPG consisting of k^* stage LFSR and some XOR gates [11] can generate pseudo-exhaustive test set for any (n, m, k) circuit.

3.1 Basic Results

Consider an (n, m, k) circuit along with the following notation. The n inputs are denoted as θ_i, $i = 1, 2, \ldots, n$, and the m outputs as O_j, $j = 1, 2, \ldots, m$, respectively. The inputs are partitioned into m sets $I_1, I_2, \ldots, I_m$ such that I_i denotes the set of inputs that drive exactly i outputs of the circuit. We first summarize some previously known results that, along with the above algebraic results, form the foundation for our bound computation.

Definition 3 *A residue r is said to be a* **proper residue** *with respect to a set of residues R if r is linearly independent with respect to the residues in R. Residue r is said to be a* **prohibited residue** *with respect to R if r is a linear combination of a subset of residues in R.*

Theorem 1 *[5] An output cone will be exhaustively tested if and only if the inputs driving the output cone are assigned proper residues.*

For an (n, m, k) circuit we need to assign proper residues to the circuit inputs from a k^*-dimensional space (where $k^* \geq k$) such that the residues assigned to the inputs driving any output cone are linearly independent. The bound computation involves guaranteeing the availability of proper residues (elements) for all circuit inputs from the k^*-dimensional space.

Definition 4 *Output O_i is said to* **dominate** *output O_j if each input that drives O_j also drives O_i. Output O_i is said to be a* **dominating** *output if it is not dominated by any other output.*

Lemma 4 *It is sufficient to consider only the dominating outputs of the circuit for determining pseudo-exhaustive test lengths.*

Proof: Let an output O_i dominate another output O_j in a circuit. Proper residue assignment to the set of inputs driving O_i ensures exhaustive testing for both output cones O_i and O_j. Hence there is no need to consider residue assignments separately for O_j. □

Definition 5 *A circuit is said to be* **reduced** *if none of its outputs is dominated by any other output.*

Any given circuit can be reduced by ignoring all of its dominated outputs. In practice, these cone-independent bounds can be applied to circuits whose cone information is available. The reduction of an (n, m, k) circuit gives an (n, m', k) reduced circuit, where $m' \leq m$. The application of the following cone-independent bounds to the reduced (n, m', k) circuit can provide tighter bounds on test length. Henceforth, we shall consider only reduced circuits.

Example 5 Consider the $(6, 6, 3)$ circuit shown in Figure 3. The inputs are denoted by θ_1 through θ_6 and outputs denoted by O_1 through O_6 respectively. The inputs can be partitioned as follows: $I_6 = I_5 = \{\}$, $I_4 = \{\theta_5\}$, $I_3 = \{\theta_1, \theta_2, \theta_3, \theta_4\}$, $I_2 = \{\theta_6\}$ and $I_1 = \{\}$. The circuit is a reduced circuit as none of its outputs is dominated by any other output. □

While determining a sufficient number of test signals for pseudo-exhaustive testing of a circuit, we need to guarantee the availability of proper residues (generated by these test signals) only to a subset of inputs. The remaining inputs are guaranteed of proper residues as stated by Lemma 5. Thus only a subset of inputs need to be considered for bound computation. In all the following results, it is assumed that all the inputs in I_i are assigned residues prior to any input in I_j $(j < i)$ is considered.

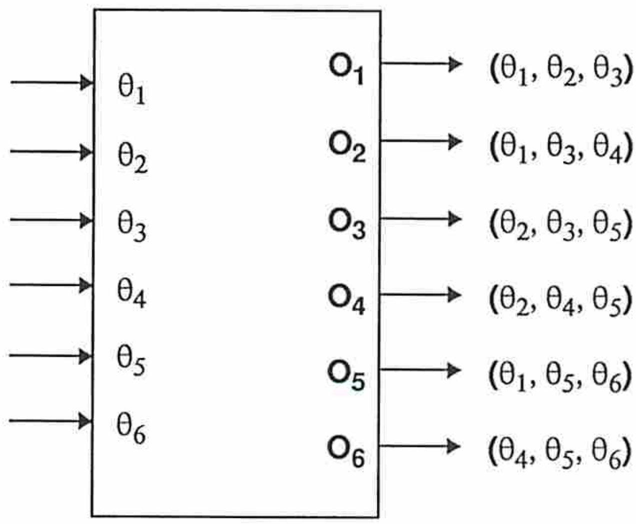

Figure 3: An (6,6,3) example circuit

Lemma 5 *For an (n, m, k) circuit, let k^* ($\geq k$) independent test signals be sufficient to assign proper residues for all inputs in I_i for all $i > 2^{k^*-k+1}$. Then these test signals are also sufficient to assign proper residues for all inputs in I_j for all $j \leq 2^{k^*-k+1}$.*

Proof Sketch: For any input that belongs to I_j for all $j \leq 2^{k^*-k+1}$, it can be shown using Lemma 2 that the total number of prohibited residues is less than 2^{k^*} and hence k^* test signals are sufficient. □

Corollary 2 *For an (n, m, k) circuit, let k independent test signals be sufficient to assign proper residues for all inputs in I_i for all $i > 2$. Then these test signals are also sufficient to assign proper residues for all inputs in I_2 and I_1.*

Lemma 5 and its corollary helps in reducing the number of inputs that need to be considered while determining the upper bound on pseudo-exhaustive test length.

Definition 6 *[1] An (n, m, k) circuit is said to be a* **maximal test concurrent** *(MTC) circuit, if it can be pseudo-exhaustively tested with k independent test signals.*

Any (n, m, k) circuit needs at least k test signals due to its maximum cone size. If $m < 3$, then the circuit inputs can only be partitioned into I_2 and I_1. Thus Corollary 2 directly leads to the following theorem (inferred from [1]).

Theorem 2 *[1] Any (n, m, k) circuit with $m < 3$ is a MTC circuit.*

We consider assigning linear combinations of test signals to inputs that is not considered in [1]. Our method can be interpreted as a generalization of [1] and helps in reducing the total number of independent test signals required for pseudo-exhaustive testing of a circuit. The above results will now be used to derive several new results including a stronger version of Theorem 2.

3.2 Results on MTC Circuits

We shall present our results on MTC circuits in this section. Lemma 5 can be strengthened by considering MTC circuits with less than six outputs. The stronger result is given by the following lemma and is useful for proving one of our main results that deals with circuits with less than six outputs.

Lemma 6 *For an (n, m, k) circuit with $m < 6$, let k independent test signals be sufficient to assign proper residues for all inputs in I_5 and I_4. Then these test signals are also sufficient to assign proper residues for all inputs in I_3.*

Proof Sketch: Let $\theta \in I_3$ drive outputs O_1, O_2 and O_3. Let k_1, k_2 and k_3 be the number of inputs driving O_1, O_2 and O_3, respectively, that are already assigned proper residues. It can be easily shown through counting arguments using Lemma 2 that the total number of prohibited residues for θ is less than 2^k provided the values of k_1, k_2 and k_3 are not simultaneously equal to $(k-1)$. There can be at most only one input in I_3 with $k_1 = k_2 = k_3 = (k-1)$. If $k_1 = k_2 = k_3 = (k-1)$ for θ, then the counting arguments result in the total number of prohibited residues for θ as 2^k. For that case, it can be shown using Lemma 3 that there exists another residue assignment for inputs in $I_3 - \{\theta\}$ such that θ can also be assigned a proper residue. □

By justifying the elimination of the assumption made in Lemma 6, a much stronger result can be obtained as given by Theorem 3. The theorem states that any circuit with five or less outputs and with maximum cone size of k inputs can always be pseudo-exhaustively tested with just 2^k patterns.

Theorem 3 *Any (n, m, k) circuit with $m < 6$ is a MTC circuit.*

Proof Sketch: For any five output circuit, it can be easily shown that all inputs in I_5 can be easily assigned proper residues from a k-dimensional space S spanned by basis B. It only needs to be shown that all inputs in I_4 can also be assigned proper residues from S. The inputs in I_4 are partitioned into five subsets $I_{4,1}, I_{4,2}, \ldots, I_{4,5}$ such that $I_{4,i} = \{$inputs that do not drive O_i $\}$ $(i = 1, 2, \ldots, 5)$. If each of the partition $I_{4,i}$ is not empty, select one input (say θ'_i) from each $I_{4,i}$ and form the set $I = \{\theta'_1, \theta'_2, \theta'_3, \theta'_4, \theta'_5\}$. Select four elements from B, say $\{x^j, x^{j+1}, x^{j+2}, x^{j+3}\}$, and assign the five residues $\{x^j, x^{j+1}, x^{j+2}, x^{j+3}, x^j + x^{j+1} + x^{j+2} + x^{j+3}\}$ to the five elements in I. This process is repeated until all the inputs from one of the partitions is selected. The remaining inputs in I_4 can be easily assigned proper residues from S. All inputs that belong to I_3, I_2 and I_1 can be assigned proper residues as per Lemma 6 and Corollary 2. □

Theorem 3 states that any five output circuit is a MTC circuit. The result is independent of the number of inputs and the maximum cone sizes of the circuits. Our result is a significant improvement over the well known result that any two output circuit is a MTC circuit (Theorem 2).

Example 6 illustrates a six output non-MTC circuit. In the example, note that even though all inputs drive exactly three outputs, the circuit is not a MTC circuit since it contains six outputs. The example illustrates the strictness of both Lemma 6 and Theorem 3.

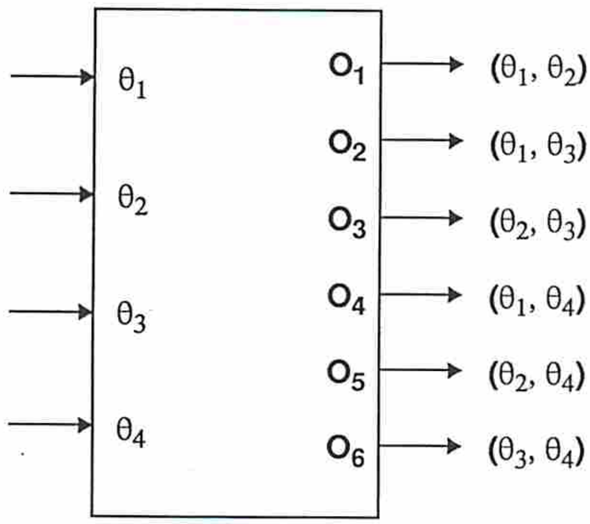

Figure 4: An (4,6,2) non-MTC circuit

Example 6 Consider the $(4,6,2)$ circuit driven by inputs $\{\theta_1, \theta_2, \theta_3, \theta_4\}$ as shown in Figure 4. Though each of the outputs depend exactly on two inputs, the circuit is not a MTC circuit and needs three independent test signals (say $1, x, x^2$). Inputs θ_1 through θ_4 can be assigned residues $1, x, 1+x$ and x^2 respectively. □

3.3 Results on (n,m,k) Circuits

The following section, containing Theorems 4 and 5 and Conjecture 1, summarizes our generic bounds on (n,m,k) circuits. It should be noted that circuits with more than five outputs can be MTC circuits.

Theorem 4 *For any (n,m,k) circuit, let $k^*(\geq k)$ be the smallest number satisfying the following inequality*

$$m \leq 2^{k^*-k+1}. \tag{1}$$

Then k^* independent test signals are sufficient for pseudo-exhaustive testing of the circuit.

Proof: Since the circuit has only at most 2^{k^*-k+1} outputs, any input can drive only at most 2^{k^*-k+1} outputs. From Lemma 5, we know that all inputs that drive at most 2^{k^*-k+1} outputs can be assigned proper residues by k^* independent test signals. □

Theorem 5 *For any (n,m,k) circuit, our bound on the number of independent test signals for pseudo-exhaustive testing given by Theorem 4 is tighter than the bound derived in [3].*

Proof: It has been shown in [3] that k^* independent test signals are sufficient if k^* satisfies the inequality

$$m \leq 2^{k^*-k}. \tag{2}$$

It is evident that our bound is tighter than the bound derived in [3] as we can accommodate twice the number of outputs for the same number of test signals. □

Conjecture 1 *For an (n,m,k) circuit, let k^* $(\geq k)$ be the smallest number satisfying the following inequality*

$$m \leq 2^{k^*-k+2}+1. \tag{3}$$

Then k^* independent test signals are sufficient for pseudo-exhaustive testing of the circuit.

Table 1: Bounds on test lengths for (n, m, k) circuit

Number of	Number of outputs		
Test Signals	Akers [3]	Theorem 4	Conjecture 1
k	1	2	$5^{\dagger}$
$k+1$	2	4	9
$k+2$	4	8	17
$k+3$	8	16	33
...	...	...	...
k^*	2^{k^*-k}	2^{k^*-k+1}	$2^{k^*-k+2}+1$

(† — proven by **Theorem 3**)

Conjecture 1 is true for MTC circuits since any (n, m, k) circuit with $m \leq 5$ is an MTC circuit as per Theorem 3.

Table 1 shows three upper bounds on the number of outputs for an (n, m, k) circuit that can be pseudo-exhaustively tested with the number of test signals given in the first column. For example, any (n, m, k) circuit with at most four outputs can be pseudo-exhaustively tested with $(k+2)$ test signals according to the bound derived in [3]. Theorem 4 states that $(k+1)$ test signals are sufficient for pseudo-exhaustive testing of any (n, m, k) circuit with $m \leq 4$. Conjecture 1 states that k test signals are sufficient for the same circuit. For a given number of test signals (say k^*), we guarantee exhaustive testing of twice the number of output cones (Theorem 4) and possibly four times the number of output cones (Conjecture 1) compared to the number of output cones guaranteed by the bound in [3].

Table 2 presents the cone-independent bounds on pseudo-exhaustive test lengths for the partitioned versions of ISCAS combinational benchmark circuits [12] and unpartitioned versions of a few ISCAS sequential benchmark circuits [13]. The combinational benchmark circuits are partitioned using our partitioning procedure [14] such that the output cones are driven by 20 or less inputs. Columns 2 and 3 present the original (n, m, k) and reduced (n, m', k) characteristics of these circuits. The last three columns present the generic bounds on pseudo-exhaustive test lengths (in terms of the number of independent test signals) based on Akers' results [3] and our results given by Theorem 4 and Conjecture 1. From the table it is evident that our bounds are tighter.

Table 2: Generic Bounds for ISCAS Benchmark Circuits

Benchmark Circuit	Original (n, m, k)	Reduced (n, m', k)	Cone Independent Bound		
			Akers [3]	Theorem 4	Conjecture 1
c432	(56,27,20)	(56,20,20)	25	24	23
c499	(49,40,14)	(49,40,14)	20	19	18
c880	(70,36,17)	(70,29,17)	22	21	20
c1355	(49,40,14)	(49,40,14)	20	19	18
c1908	(47,39,20)	(47,26,20)	25	24	23
c2670	(262,169,20)	(262,117,20)	27	26	25
c3540	(108,80,20)	(108,57,20)	26	25	24
c5315	(215,160,20)	(215,91,20)	27	26	25
c6288	(99,98,20)	(99,39,20)	26	25	24
c7552	(286,187,20)	(286,69,20)	27	26	25
s27	(7,2,6)	(7,2,6)	7	6	6
s208	(19,10,18)	(19,3,18)	19	19	$18^{\dagger}$
s298	(17,19,8)	(17,10,8)	12	11	10
s344	(24,21,13)	(24,9,13)	17	16	14
s349	(24,21,13)	(24,9,13)	17	16	14
s382	(24,15,14)	(24,10,14)	18	17	16
s386	(13,6,12)	(13,2,12)	13	12	12
s420	(35,18,34)	(35,5,34)	35	35	$34^{\dagger}$
s444	(24,15,14)	(24,10,14)	18	17	16
s510	(25,5,20)	(25,2,20)	21	20	20
s526	(24,24,14)	(24,10,14)	18	17	16
s641	(50,35,28)	(50,23,28)	33	32	31
s713	(50,35,28)	(50,23,28)	33	32	31
s820	(23,15,21)	(23,3,21)	23	22	$21^{\dagger}$
s832	(23,15,21)	(23,3,21)	23	22	$21^{\dagger}$
s838	(67,34,66)	(67,9,66)	67	67	67
s953	(22,20,18)	(22,6,18)	21	20	19

($\dagger$ — proven by **Theorem 3**)

4 Circuit-specific Bounds

For an overwhelming majority of circuits, we can utilize the information about cone dependencies to derive a bound on the number of independent test signals required for pseudo-exhaustive testing of the circuit. Circuit-specific bounds are tighter than the generic bounds derived earlier. We shall derive bounds for both LFSR/XOR and LFSR/SR structures and show that our bounds are better than those derived in [3] and [5].

Let us consider the (n, m, k) circuit along with the notation that input θ_i is assigned a unique index π_i, where $1 \leq \pi_i \leq n$. A permutation of inputs is specified completely by the n-tuple $(\pi_1, \pi_2, \ldots, \pi_n)$. The default permutation is given by $\pi_i = i$ for θ_i, $i = 1, 2, \ldots, n$. We shall assume the default permutation of inputs unless stated otherwise.

The input dependencies for an output is represented by an ordered set of inputs. The inputs are arranged in the ordered set in increasing order of their indices. Consider output O_j being driven by k inputs $\theta_{i_1}, \theta_{i_2}, \ldots, \theta_{i_k}$. Let $1 \leq i_1 < i_2 < \ldots i_k \leq n$. Under the default permutation of inputs, the input dependencies for O_j is represented by the ordered set $\{\theta_{i_1}, \theta_{i_2}, \ldots, \theta_{i_k}\}$. Let $p_{i,j}$ denote the position of θ_i in the ordered dependency set for O_j. If θ_i drives O_j, then $p_{i,j}$ takes appropriate value between 1 and k, otherwise $p_{i,j} = 0$. Let $p_i^* = max \ \{p_{i,1}, p_{i,2}, \ldots, p_{i,m}\}$ denote the maximum position in which θ_i occurs among the input dependencies for all m outputs. Let $f_{i,j}$ be a boolean variable such that $f_{i,j} = 1$ if $p_{i,j} > 0$ and $f_{i,j} = 0$ if $p_{i,j} = 0$. Let $f_i^* = \sum_{j=1}^{m} f_{i,j}$ denote the number of occurrences (frequency) of θ_i among all m outputs. The notation is illustrated in the following example.

Example 7 Consider the $(6, 6, 3)$ circuit shown in Figure 3. Let us assume the default permutation where θ_i is assigned index $\pi_i = i$. The input dependencies for the six outputs are

$$\begin{array}{lll} O_1 :: \ \theta_1 \ \theta_2 \ \theta_3 & O_2 :: \ \theta_1 \ \theta_3 \ \theta_4 & O_3 :: \ \theta_2 \ \theta_3 \ \theta_5 \\ O_4 :: \ \theta_2 \ \theta_4 \ \theta_5 & O_5 :: \ \theta_1 \ \theta_5 \ \theta_6 & O_6 :: \ \theta_4 \ \theta_5 \ \theta_6 \end{array}$$

Input θ_2 appears in second position for O_1 and first positions for O_3 and O_4. Thus for θ_2 we have $p_{2,j}$ values $(j = 1, 2, \ldots, 6)$ of 2, 0, 1, 1, 0 and 0, respectively, and $f_{2,j}$ values $(j = 1, 2, \ldots, 6)$ of 1, 0, 1, 1, 0 and 0, respectively. Hence $p_2^* = 2$ and $f_2^* = 3$. □

4.1 LFSR/XORs

We shall derive tight upper bounds for the test sets generated by LFSR/XOR structures for a given (n, m, k) circuit. The circuit cones are described in terms of the parameters defined above. Since these bounds are derived based on the ordering of the circuit inputs, we shall determine the best permutation of inputs to achieve the best improvement of these bounds.

4.1.1 Bounds based on Default Permutation

Theorem 6 *For an (n, m, k) circuit, let $p_{i,j}$, p_i^* and $f_{i,j}$ be the circuit parameters (defined earlier) characterizing the cone dependencies. Let k^* be the smallest number satisfying the following inequality for all inputs θ_i, $1 \leq i \leq n$.*

$$\lceil 2^{2p_i^*-2-k^*} \rceil + \sum_{j=1}^{m} f_{i,j}\{2^{p_{i,j}-1} - \lceil 2^{p_i^*+p_{i,j}-2-k^*} \rceil\} < 2^{k^*} \tag{4}$$

Then k^* independent test signals are sufficient for pseudo-exhaustive testing of the circuit.

Proof Sketch: For input θ_i, the output in which θ_i appears at position p_i^* is considered as the reference output. For this output, θ_i appears along with $(p_i^* - 1)$ inputs that have been already assigned proper residues. These $(p_i^* - 1)$ residues span a $(p_i^* - 1)$-dimensional subspace (say S_{j^*}) and all the elements in this subspace are prohibited residues for θ_i. For any output O_j with $p_{i,j} > 0$, θ_i appears along with $(p_{i,j} - 1)$ inputs that have been already assigned proper residues. These residues span a $(p_{i,j} - 1)$-dimensional space (say S_j) and all the elements in this space are prohibited residues for θ_i. From Lemma 1, we know that subspaces S_{j^*} and S_j have at least $\lceil 2^{p_i^*+p_{i,j}-2-k^*} \rceil$ common elements. Thus the LHS expression of Equation 4 gives an upper bound on the total number of prohibited residues for θ_i. The first term in the LHS expression is due to the error in the summation for the reference output. As long as this expression is less than 2^{k^*}, a proper residue from S is guaranteed for θ_i. □

Theorem 7 *The cone dependent bound on the number of independent test signals given by Theorem 6 is tighter than the cone independent bound given by Theorem 4.*

Proof: It is enough to show that the cone independent bound can be derived by assuming the worst case in the derivation of cone dependent bound. For an input θ_i with $p_{i,j} = k$ for all m outputs, we have $p_i^* = k$ and Equation 4 simplifies to

$$\begin{aligned} \lceil 2^{2k-2-k^*} \rceil + m \times (2^{k-1} - \lceil 2^{2k-2-k^*} \rceil) &< 2^{k^*} \\ \Rightarrow \quad (m-1)(2^{k-1} - \lceil 2^{2k-2-k^*} \rceil) &< 2^{k^*} - 2^{k-1} \\ \Rightarrow \quad m &\leq 2^{k^*-k+1} \end{aligned}$$

Thus the cone dependent bound is tighter than the cone independent bound. □

4.1.2 Improvement on Bounds by Permutation

Given an (n, m, k) circuit, the bound on the number of independent test signals given by Theorem 6 can be improved by allowing the permutation of inputs. We shall describe a permutation algorithm that assigns unique indices to circuit inputs resulting in low (high) $p_{i,j}$ values for inputs driving many (few) outputs. The algorithm modifies the circuit parameters (that characterizes the cone dependencies) and allows Equation 4 to be satisfied for a smaller value of k^*.

Algorithm XORBound

Input: Output cone dependencies of (n, m, k) circuit.
Output: Upper bound on the number of independent test signals k^* ($\geq k$).

1. Determine all dominating outputs and consider only the reduced circuit.
2. $k^* \leftarrow k$. /* k^* is the number of independent test signals */
3. Determine f_i^* for input θ_i $\forall i = 1, 2, \ldots, n$. /* determine the input frequencies */
4. $\pi_i \leftarrow 0$ $\forall i = 1, 2, \ldots, n$; $n^* \leftarrow n$.
   /* initialize the indices of inputs and n^* is the current highest index */
5. For each unassigned θ_i do
   (a) If $f_i^* \leq 2^{k^*-k+1}$ then { $\pi_i \leftarrow n^*$; $n^* \leftarrow n^* - 1$ }
6. While n^* is decremented do
   (a) For each unassigned θ_i do
      i. $\pi_i \leftarrow n^*$.
      ii. Check the satisfiability of Equation 4 for θ_i.
      iii. If the equation is satisfied then $n^* \leftarrow n^* - 1$; else $\pi_i \leftarrow 0$.
7. If $n^* > 0$ then
   (a) If $k^* = n$, go to Step 8.
   (b) $k^* \leftarrow k^* + 1$; Go to Step 4.
8. Output the number of test signals (k^*).

The algorithm *XORBound* determines a minimal number of independent test signals that are sufficient for pseudo-exhaustive testing of a given circuit. Lemma 4 enables us to consider only dominating outputs for determining the bound on test length. Lemma 5 states that a set of k^* test signals guarantees proper residues for each input that drives at most 2^{k^*-k+1} outputs and hence all these inputs are assigned highest possible indices. From the remaining set of unassigned inputs, an input (say θ_i) is assigned the current highest index (n^*) provided it satisfies Equation 4. The $p_{i,j}$ values for θ_i are determined based on the fact that the remaining unassigned inputs can have indices only less than n^*. The unassigned inputs are repeatedly considered for assignment until there is no decrease in the value of n^*. Any further existence of unassigned inputs mandates an increment to the number of test signals and an iteration of the entire algorithm.

The complexity of the algorithm can be computed as follows. Every iteration of the *while* loop results in assigning proper indices to one or more inputs. The number of iterations of the *while* loop is bounded above by $n(n+1)/2$ since every iteration can result in assigning a proper index to only one input. The satisfiability check for input θ_i involves determining $p_{i,j}$ values for all m outputs. Thus the complexity of the *while* loop is given by $O(mn^2)$. The number of iterations of the entire algorithm is bounded above by $(n-k)$. Thus the complexity of the algorithm is given by $O(mn^3)$, where n and m are the number of inputs and outputs to the circuit respectively.

In general, considering all permutations of inputs and using Theorem 6 for determining the tightest possible bound has exponential complexity. The following theorem states that our permutation algorithm of polynomial complexity is sufficient to find the tightest possible bound using Theorem 6.

Theorem 8 *Algorithm XORBound is of polynomial complexity and determines the tightest possible bound on the number of test signals that can be achieved using Theorem 6.*

Proof Sketch: It will suffice to show that algorithm *XORBound* results in a minimum subset of inputs that are not assigned indices after the completion of the *while* loop. This can be proven by contradiction on the minimality of the set of unassigned inputs. □

Example 8 Consider the $(6,6,3)$ circuit described in Example 5. Akers' bound using Equation 2 requires *six* signals. Our bound using Equation 4 without allowing a permutation of inputs requires *four* signals. Applying the algorithm *XORBound* reduces our bound to *three* test signals. The circuit can be tested with three independent test signals. Residues $\{1, x, x^2, 1+x, 1+x^2, x\}$ are assigned to inputs 1 through 6 respectively. □

4.1.3 Experimental Results

Table 3 presents the cone-dependent bounds on test lengths for LFSR/XORs for the partitioned versions of ISCAS combinational benchmark circuits [12] and unpartitioned versions of a few ISCAS sequential benchmark circuits [13]. Columns 2 and 3 present the original (n, m, k) and reduced (n, m', k) characteristics of these circuits. The last three columns present the bounds on test lengths (in terms of the number of independent test signals) by considering the reduced circuits. The cone-independent bounds are determined using Theorem 4. The cone-dependent bounds with the default permutation of inputs are determined using Theorem 6. The algorithm *XORBound* achieves tighter bounds on pseudo-exhaustive test lengths by determining one of the best permutation of inputs. The improvement of the bounds by allowing permutation of inputs is evident from the table. The circuit-specific bounds determined by allowing for the permutation of inputs are optimal for all these circuits except for circuit *c6288*.

4.2 LFSR/SRs

An (n, m, k) circuit can be pseudo-exhaustively tested by a simple LFSR/SR if there exists a primitive feedback polynomial of degree $k^*(\geq k)$ such that the residues assigned to the inputs driving each output are linearly independent as stated by Theorem 1.

Definition 7 *The primitive feedback polynomial of an LFSR/SR considered for a given circuit is said to be* **inapplicable** *if the polynomial results in a set of linearly dependent residues for the set of inputs driving some output of the circuit.*

Theorem 9 *[15] The total number of primitive polynomials of degree k^* is given by $\Phi(2^{k^*} - 1)/k^*$, where Φ is Euler's phi function.*

4.2.1 Bounds based on Default Permutation

We shall assume the default permutation of inputs where input θ_i is fed by the ith stage of an LFSR/SR. Input θ_i is assigned the residue $x^i \bmod P(x)$, where $P(x)$ is the primitive feedback polynomial of the LFSR/SR.

Theorem 10 *For an (n, m, k) circuit, let $p_{i,j}$ and $f_{i,j}$ be the circuit parameters (defined earlier) characterizing the cone dependencies. Let k^* be the smallest number satisfying the*

Table 3: Circuit-specific bounds for LFSR/XORs

Benchmark Circuit	Original (n, m, k)	Reduced (n, m', k)	Cone-independent Bound	Cone-dependent Bound with default permutation	Cone-dependent Bound with best permutation
c432	(56,27,20)	(56,20,20)	24	20	20
c499	(49,40,14)	(49,40,14)	19	16	14
c880	(70,36,17)	(70,29,17)	21	18	17
c1355	(49,40,14)	(49,40,14)	19	16	14
c1908	(47,39,20)	(47,26,20)	24	20	20
c2670	(262,169,20)	(262,117,20)	26	20	20
c3540	(108,80,20)	(108,57,20)	25	21	20
c5315	(215,160,20)	(215,91,20)	26	21	20
c6288	(99,98,20)	(99,39,20)	25	22	21
c7552	(286,187,20)	(286,69,20)	26	20	20
s27	(7,2,6)	(7,2,6)	6	6	6
s208	(19,10,18)	(19,3,18)	19	18	18
s298	(17,19,8)	(17,10,8)	11	9	8
s344	(24,21,13)	(24,9,13)	16	13	13
s349	(24,21,13)	(24,9,13)	16	13	13
s382	(24,15,14)	(24,10,14)	17	14	14
s386	(13,6,12)	(13,2,12)	12	12	12
s420	(35,18,34)	(35,5,34)	35	34	34
s444	(24,15,14)	(24,10,14)	17	14	14
s510	(25,5,20)	(25,2,20)	20	20	20
s526	(24,24,14)	(24,10,14)	17	14	14
s641	(50,35,28)	(50,23,28)	32	28	28
s713	(50,35,28)	(50,23,28)	32	28	28
s820	(23,15,21)	(23,3,21)	22	21	21
s832	(23,15,21)	(23,3,21)	22	21	21
s838	(67,34,66)	(67,9,66)	67	66	66
s953	(22,20,18)	(22,6,18)	20	18	18

following inequality

$$\sum_{j=1}^{m}\sum_{i=k^*}^{n} i \times f_{i,j}(2^{p_{i,j}-1}-1) < \Phi(2^{k^*}-1) \approx 2^{k^*}-1. \tag{5}$$

Then a simple LFSR/SR based on a degree k^* primitive polynomial is sufficient for pseudo-exhaustive testing of the circuit.

Proof Sketch: The proof can be derived as an extension of the arguments presented in [5]). The LHS expression (divided by k^*) forms an upper bound on the number of inapplicable primitive polynomials based on the $p_{i,j}$ values for all inputs θ_i and outputs O_j. The RHS expression (divided by k^*) gives the total number of primitive polynomials of degree k^* as per Theorem 9 stated in [15]. □

Theorem 11 *For any (n,m,k) circuit, our bound on the degree of LFSR/SR given by Theorem 10 is tighter than the bound derived in [5].*

Proof: It has been shown in [5] that a simple LFSR/SR of degree $\hat{k}^*$ is sufficient for pseudo-exhaustive testing of an (n,m,k) circuit if $\hat{k}^*$ satisfies the equation

$$n \times m \times (2^k - 1) < \Phi(2^{\hat{k}^*}-1) \approx 2^{\hat{k}^*}-1. \tag{6}$$

We shall show that the value of k^* in Equation 5 is bounded above by the value of $\hat{k}^*$ in Equation 6.

Let E_j denote the expression $\sum_{i=k^*}^{n} i \times f_{i,j} \times (2^{p_{i,j}-1}-1)$. The LHS expression of Equation 5 can be expressed as $\sum_{j=1}^{m} E_j$. Let us consider the input dependencies for output O_j given by the ordered set $\{\theta_{i_1}, \theta_{i_2}, \ldots, \theta_{i_k}\}$. For this output we compute E_j as

$$\begin{aligned} E_j &= \sum_{i=k^*}^{n} i \times f_{i,j} \times (2^{p_{i,j}-1}-1) \\ &\leq \sum_{q=1}^{k} i_q \times (2^{q-1}-1) \\ &\leq n \times \sum_{q=1}^{k}(2^{q-1}-1) \quad \text{(since } i_q \leq n\text{)} \\ &< n \times (2^k - 1). \end{aligned}$$

Summing up E_j for all values of j, we get

$$\sum_{j=1}^{m} E_j < \sum_{j=1}^{m} n \times (2^k-1) = m \times n \times (2^k - 1).$$

Thus the LHS expression of Equation 5 is smaller than the LHS expression in Equation 6. Hence k^* value in Equation 5 is bounded above by $\hat{k}^*$ value in Equation 6. □

4.2.2 Improvement on Bounds by Permutation

Given an (n, m, k) circuit, the bound on the degree of the applicable primitive polynomial for an LFSR/SR given by Theorem 10 can be improved by permuting the inputs. We shall attempt to minimize the total number of inapplicable primitive polynomials given by the LHS expression in Equation 5. Thus an improvement on the bound can be obtained for the degree of the applicable primitive polynomial for LFSR/SR. This is similar to the improvement on the bound achieved for LFSR/XORs.

Algorithm SRBound

Input: Output cone dependencies of (n, m, k) circuit.
Output: Upper bound on the degree k^* ($\geq k$) of applicable primitive polynomial.

1. Determine all dominating outputs and consider only the reduced circuit.
2. $k^* \leftarrow k$.
   /* k^* is the degree of the primitive polynomial */
3. Assign indices to inputs according to the input permutation determined by the algorithm *XORBound*.
4. While Equation 5 is not satisfied do
   (a) If $k^* = n$, go to Step 5.
   (b) $k^* \leftarrow k^* + 1$.
5. Output the degree of the applicable primitive polynomial (k^*).

For a given circuit, the algorithm *SRBound* usually determines an applicable primitive polynomial of smaller degree than the default permutation. Only dominating outputs are considered as per Lemma 4. The input permutation determined by the algorithm *XORBound* is used to minimize the LHS expression of Equation 5. The satisfiability check involves computing $p_{i,j}$ values for all inputs driving each output. Since the input permutation determined by the algorithm *XORBound* is used again in the algorithm *SRBound*, the complexity of the algorithm *SRBound* is the same as that of the complexity of the algorithm *XORBound*. However, the algorithm *SRBound* for LFSR/SRs does not guarantee the tightest possible bound unlike the algorithm *XORBound* for LFSR/XORs.

Example 9 Consider again the $(6, 6, 3)$ circuit described in Example 5. For LFSR/SRs, Barzilai's bound determined by Equation 6 requires *eight* test signals. The bound computed using Equation 5 without allowing permutation of inputs requires a primitive polynomial of degree *five*. The algorithm $SRBound$ still requires a degree *five* polynomial. However, the circuit can be tested with an LFSR/SR using the polynomial $x^4 + x + 1$. □

4.2.3 Experimental Results

Table 4 presents the cone-dependent bounds on test lengths for LFSR/SRs for the partitioned versions of ISCAS combinational benchmark circuits and unpartitioned versions of some ISCAS sequential benchmark circuits [13]. The last three columns present the bounds on test lengths (in terms of number of independent test signals) by considering only the reduced circuits. Barzilai's bounds are determined using Equation 6 and our bounds with default permutation of inputs are determined using Equation 5. The algorithm $SRBound$ results in tighter bounds by using the same permutation of inputs that were originally determined for LFSR/XORs. The improvement of the bounds by allowing permutation of inputs is evident from the table. It should be noted that our LFSR/SR bounds represent test lengths that are a few orders of magnitude smaller than those given by Barzilai's bounds.

5 Conclusion

In this paper we have first derived a few important algebraic results on the set union and intersection operations between vector subspaces. We have determined (a) the minimum overlap between distinct subspaces and (b) the minimum number of distinct subspaces contained in a vector space. These algebraic results are used in the derivation of the bounds on pseudo-exhaustive test lengths.

We have determined a few generic bounds on test lengths that are independent of the structural information about the circuit output cones. We have shown that any circuit with less than six outputs is maximal test concurrent. We have derived an expression for the number of independent test signals that are sufficient for pseudo-exhaustive testing of any given (n, m, k) circuit. The expression is based on the number of outputs (m) and the maximum cone size (k) of the circuit.

We have also derived a few circuit-specific bounds utilizing the structural information about the circuit output cones. We have derived tight upper bounds on the test sets generated by LFSR/XORs and LFSR/SRs and shown that our bounds are better than those derived in [3] and [5]. We have developed algorithms of polynomial complexity to permute circuit

Table 4: Circuit-specific bounds for LFSR/SRs

Benchmark Circuit	Original (n, m, k)	Reduced (n, m', k)	Cone-dependent Bound Barzilai [5]	Cone-dependent Bound with default permutation	Cone-dependent Bound with good permutation
c432	(56,27,20)	(56,20,20)	31	28	26
c499	(49,40,14)	(49,40,14)	25	23	22
c880	(70,36,17)	(70,29,17)	28	26	25
c1355	(49,40,14)	(49,40,14)	25	23	22
c1908	(47,39,20)	(47,26,20)	31	27	26
c2670	(262,169,20)	(262,117,20)	35	30	29
c3540	(108,80,20)	(108,57,20)	33	31	30
c5315	(215,160,20)	(215,91,20)	35	32	31
c6288	(99,98,20)	(99,39,20)	32	30	30
c7552	(286,187,20)	(286,69,20)	35	32	30
s27	(7,2,6)	(7,2,6)	7	7	6
s208	(19,10,18)	(19,3,18)	19	19	18
s298	(17,19,8)	(17,10,8)	16	14	13
s344	(24,21,13)	(24,9,13)	21	18	15
s349	(24,21,13)	(24,9,13)	21	18	15
s382	(24,15,14)	(24,10,14)	22	19	18
s386	(13,6,12)	(13,2,12)	13	13	12
s420	(35,18,34)	(35,5,34)	35	35	34
s444	(24,15,14)	(24,10,14)	22	18	14
s510	(25,5,20)	(25,2,20)	25	24	20
s526	(24,24,14)	(24,10,14)	22	18	14
s641	(50,35,28)	(50,23,28)	39	33	31
s713	(50,35,28)	(50,23,28)	39	33	31
s820	(23,15,21)	(23,3,21)	23	23	21
s832	(23,15,21)	(23,3,21)	23	23	21
s838	(67,34,66)	(67,9,66)	67	67	66
s953	(22,20,18)	(22,6,18)	22	22	18

inputs to obtain good improvements on these bounds. Our bounds provide good estimates of pseudo-exhaustive test lengths and can be used as guiding factors in designing circuit-specific TPGs. The computed theoretical bounds for the partitioned benchmark circuits comply well with the pseudo-exhaustive test lengths generated by circuit-specific TPGs as reported in [11].

References

[1] E. J. McCluskey. Verification Testing — A Pseudoexhaustive Test Technique. *IEEE Trans. on Computers*, C-33(6):541–546, June 1984.

[2] M. Abramovici, M. A. Breuer, and A. D. Friedman. *Digital Systems Testing and Testable Design*. IEEE Press, 1994.

[3] S. B Akers. On the Use of Linear Sums in Exhaustive Testing. In *Proc. 15th Int'l. Symp. on Fault-Tolerant Computing*, pages 148–153, June 1985.

[4] C. H. Chen. BISTSYN - A Built-In Self-Test Synthesizer. In *Proc. Int'l Conf. on Computer Aided Design*, pages 240–243, 1991.

[5] Z. Barzilai, D. Coppersmith, and A. Rosenberg. Exhaustive Bit Pattern Generation in Discontiguous Positions with Applications to VLSI Testing. *IEEE Trans. on Computers*, C-32(2):190–194, February 1983.

[6] D. Kagaris and S. Tragoudas. Cost-Effective LFSR Synthesis for Optimal Pseudo-Exhaustive BIST Test Sets. *IEEE Trans. on VLSI Systems*, 1(4):526–536, December 1993.

[7] L. T. Wang and E. J. McCluskey. Circuits for Pseudoexhaustive Test Pattern Generation. *IEEE Trans. on Computer-Aided Design*, 7(10):1068–1080, October 1988.

[8] J. G. Udell. Reconfigurable Hardware for Pseudo-Exhaustive Test. In *Proc. Int'l Test Conf.*, pages 522–530, September 1988.

[9] W. B. Jone and C. A. Papachristou. A Coordinated Approach to Partitioning and Test Pattern Generation for Pseudoexhaustive Testing. In *Proc. Design Automation Conf.*, pages 525–530, June 1989.

[10] S. Hellebrand, H-J. Wunderlich, and O. F. Haberl. Generating Pseudo-Exhaustive Vectors for External Testing. In *Proc. Int'l Test Conf.*, pages 670–679, September 1990.

[11] R. Srinivasan, S. K. Gupta, and M. A. Breuer. Novel Test Patttern Generators for Pseudo-Exhaustive Testing. In *Proc. Int'l Test Conf.*, pages 1041–1050, October 1993.

[12] F. Brglez and H. Fujiwara. A Neutral Netlist of Ten Combinational Benchmark Circuits and a Target Translator in FORTRAN. In *Proc. Int'l. Symp. on Circuits and Systems*, pages 663–698, June 1985.

[13] F. Brglez et al. Combination Profiles of Sequential Benchmark Circuits. In *Proc. Int'l. Symp. on Circuits and Systems*, pages 1929–1934, May 1989.

[14] R. Srinivasan, S. K. Gupta, and M. A. Breuer. An Efficient Partitioning Strategy for Pseudo-Exhaustive Testing. In *Proc. Design Automation Conf.*, pages 242–248, June 1993.

[15] S. W. Golomb. *Shift Register Sequences.* Aegean Park Press, 1982.

[16] I. N. Herstein. *Topics in Algebra.* Xerox College Publishing, 1975.

Appendix: Proofs of Lemmas and Theorems

Lemma 1 *Consider a k-dimensional space S and any two distinct subspaces S_1 and S_2 of dimensions k_1 and k_2 contained in S. The set $S_1 \cap S_2$ is a subspace contained in S and consists of at least $\lceil 2^{k_1+k_2-k} \rceil$ elements.*

Proof: Let $S_3 = S_1 \cap S_2$. Consider any two elements a and b such that $a, b \in S_3$. Since $S_3 \subset S_1$ and $S_3 \subset S_2$, we have $a, b \in S_1$ and $a, b \in S_2$. Since S_1 and S_2 are subspaces, we have that $(a + b) \in S_1$ and $(a + b) \in S_2$ implies $(a + b) \in S_3$. Thus S_3 forms a subspace contained in S. Let x be the dimension of subspace S_3.

Let S_1, S_2 and S_3 be spanned by the bases B_1, B_2 and B_3 respectively. Since $S_3 \subset S_1$ and $S_3 \subset S_2$, we can choose B_1 and B_2 such that $B_3 \subset B_1$ and $B_3 \subset B_2$. Since $|B_1| = k_1$, $|B_2| = k_2$ and $|B_3| = x$, we have

$$|B_1 \cup B_2| \;=\; |B_1| + |B_2| - |B_1 \cap B_2| \;=\; |B_1| + |B_2| - |B_3| \;=\; k_1 + k_2 - x$$

Let S_4 be the $(k_1 + k_2 - x)$-dimensional subspace spanned by the basis $B_1 \cup B_2$. Since $S_4 \subseteq S$, we have

$$k_1 + k_2 - x \;\leq\; k \;\Rightarrow\; x \;\geq\; k_1 + k_2 - k$$

Hence $S_1 \cap S_2$ is a subspace of dimension at least $(k_1 + k_2 - k)$. Although the term $(k_1 + k_2 - k)$ could be negative, $S_1 \cap S_2$ always contains the additive identity element (zero). Hence $S_1 \cap S_2$ is a subspace contained in S and has at least $\lceil 2^{k_1+k_2-k} \rceil$ elements. □

Lemma 2 *A k-dimensional space is composed of at least $(2^i + 1)$ distinct subspaces of dimensions less than or equal to $(k - i)$, where $1 \leq i \leq (k - 1)$.*

Proof: We shall prove the theorem in two parts. First we will show that a k-dimensional space is composed of at least $(2^i + 1)$ distinct $(k - i)$-dimensional subspaces. Then we will generalize the dimensions of the $(2^i + 1)$ distinct subspaces to less than or equal to $(k - i)$.

Part I: Consider any two distinct $(k - i)$-dimensional subspaces S_1 and S_2 contained in a k-dimensional space S. From Lemma 1, the two subspaces S_1 and S_2 must have a common subspace of dimension at least $(k - 2i)$. Hence we have

$$\begin{aligned} |S_1| &= |S_2| = 2^{k-i} \\ |S_1 \cap S_2| &\geq 2^{k-2i} \\ |S_1 \cup S_2| &= |S_1| + |S_2| - |S_1 \cap S_2| \leq 2^{k-i} + 2^{k-i} - 2^{k-2i}. \end{aligned}$$

Let $S_1, S_2, \ldots, S_x$ be x distinct $(k - i)$-dimensional subspaces such that $\cup_{j=1}^{x} S_j = S$. Each of these subspaces can have at most $(2^{k-i} - 2^{k-2i})$ elements unique to them. Hence we have

$$|S| = \left| \bigcup_{j=1}^{x} S_j \right| = 2^k \leq 2^{k-i} + (x-1)(2^{k-i} - 2^{k-2i})$$

Multiplying throughout by 2^{2i-k} we get

$$\begin{aligned} 2^{2i} &\leq 2^i + (x-1)(2^i - 1) \\ \Rightarrow \quad x &\geq \frac{2^{2i} - 2^i}{2^i - 1} + 1 \\ \Rightarrow \quad x &\geq 2^i + 1. \end{aligned}$$

Thus a k-dimensional space is composed of at least $(2^i + 1)$ distinct $(k - i)$-dimensional subspaces.

Part II: Now we shall generalize the dimensions of the $(2^i + 1)$ distinct subspaces. Let $S_0^*, S_1^*, \ldots, S_{2^i}^*$ be $(2^i + 1)$ distinct subspaces contained in a k-dimensional space S with dimensions $k_0, k_1, \ldots, k_{2^i}$ respectively. Let $k_j \leq (k - i)\ \forall\, j = 0, 1, \ldots, 2^i$. Let $S_0, S_1, \ldots, S_{2^i}$ be $(2^i + 1)$ distinct $(k - i)$-dimensional subspaces contained in S such that $S_j^* \subseteq S_j\ \forall\, j = 0, 1, \ldots, 2^i$. From Part I we know that

$$\begin{aligned} & \bigcup_{j=0}^{2^i} S_j \subseteq S. \\ \Rightarrow \quad & \bigcup_{j=0}^{2^i} S_j^* \subseteq \bigcup_{j=0}^{2^i} S_j \subseteq S. \end{aligned}$$

Thus a k-dimensional space is composed of at least $(2^i + 1)$ distinct subspaces of dimensions less than or equal to $(k - i)$. □

Lemma 3 *Consider a k-dimensional space S and any three distinct $(k-1)$-dimensional subspaces S_1, S_2 and S_3 contained in S. Let $S^* = S_1 \cap S_2$. The subspace S_3 satisfies the relation $S_1 \cup S_2 \cup S_3 = S$ if and only if $S_1 \cap S_2 \cap S_3 = S^*$.*

Proof: Since S_1 and S_2 are distinct $(k-1)$-dimensional subspaces contained in S, S^* is a $(k-2)$-dimensional subspace as per Corollary 1. Consider an element a such that $a \in S_1$ and $a \notin S^*$. Let $T_1 = \{a + s \mid \forall s \in S^*\}$. Then we have $T_1 \subset S_1$ and $|T_1| = |S^*| = 2^{k-2}$. The set $S^* \cap T_1 = \emptyset$ since $a \notin S^*$. The set $S^* \cup T_1$ contains 2^{k-1} elements and hence $S^* \cup T_1 = S_1$. Consider another element b such that $b \in S_2$ and $b \notin S^*$. Let $T_2 = \{b + s \mid \forall s \in S^*\}$. Then we have $T_2 \subset S_2$ and $|T_2| = |S^*| = 2^{k-2}$. The set $S^* \cap T_2 = \emptyset$ since $b \notin S^*$. The set $S^* \cup T_2$ contains 2^{k-1} elements and hence $S^* \cup T_2 = S_2$. The Venn diagram of these sets are shown in Figure 2. Thus we have

$$
\begin{aligned}
S_1 &= S^* \cup \{a + s \mid \forall s \in S^*\} = S^* \cup T_1 \\
S_2 &= S^* \cup \{b + s \mid \forall s \in S^*\} = S^* \cup T_2 \\
S_1 \cup S_2 &= S^* \cup \{a + s \mid \forall s \in S^*\} \cup \{b + s \mid \forall s \in S^*\} = S^* \cup T_1 \cup T_2
\end{aligned}
$$

Let $T_3 = \{a+b+s \mid \forall s \in S^*\}$. Since $a, b \notin S^*$, we know that $a \notin S_2$, $b \notin S_1$ and $a+b \notin S_1 \cup S_2$. Therefore $T_3 \cap S_1 = T_3 \cap S_2 = \emptyset$. We know that $T_3 \subset S$ and $|T_3| = |S^*| = 2^{k-2}$. The sets S^*, T_1, T_2 and T_3 are disjoint to each other and the set $S^* \cup T_1 \cup T_2 \cup T_3$ contains 2^k elements and hence $S^* \cup T_1 \cup T_2 \cup T_3 = S$. The elements of S are partitioned into four equal sized subsets S^*, T_1, T_2 and T_3 (the subsets are called cosets in algebra terminology [16]) as shown in Figure 2.

The set T_i $(i = 1, 2, 3)$ does not form a subspace and $S^* \subset L(T_i)$. If a subspace (say S_x) contains S^* and an element from T_i, then $T_i \subset S_x$. The subsets S^*, T_1, T_2 and T_3 are unique to any given two subspaces S_1 and S_2,

(If):: Assume that $S_1 \cap S_2 \cap S_3 = S^*$. The set $S_3 \cap T_1 = \emptyset$ since if $S_3 \cap T_1 \neq \emptyset$, then $T_1 \subset S_3$ and $S_3 = S_1$. Similarly the set $S_3 \cap T_2 = \emptyset$ since if $S_3 \cap T_2 \neq \emptyset$, then $T_2 \subset S_3$ and $S_3 = S_2$. Hence $S_3 \cap T_3 \neq \emptyset$. Since $S^* \subset S_3$ and $T_3 \cap S_3 \neq \emptyset$, we have $T_3 \subset S_3$ and $S_3 = S^* \cup T_3$. Therefore $S_1 \cup S_2 \cup S_3 = S^* \cup T_1 \cup T_2 \cup T_3 = S$.

(Only If):: Assume that $S_1 \cup S_2 \cup S_3 = S$. This implies $T_3 \subset S_3$. Since S_3 is a subspace, $S^* \subset L(T_3) \subseteq S_3$. Therefore $S_1 \cap S_2 \cap S_3 = S^*$. □

Lemma 5 *For an (n, m, k) circuit, let k^* $(\geq k)$ independent test signals be sufficient to assign proper residues for all inputs in I_i for all $i > 2^{k^*-k+1}$. Then these test signals are also sufficient to assign proper residues for all inputs in I_j for all $j \leq 2^{k^*-k+1}$.*

Proof: Let S be the k^*-dimensional space generated by k^* independent test signals. Assume that all inputs in I_i for all $i > 2^{k^*-k+1}$ have been assigned proper residues from S. Let input $\theta \in I_{2^{k^*-k+1}}$ drive output O_j. Assume that k_j inputs (where $k_j \leq (k-1)$) driving O_j have been already assigned proper residues and the residue assignment for θ is under consideration. The residues assigned to k_j inputs span a k_j-dimensional subspace and none of the elements from this subspace can be assigned as a proper residue for θ. In other words, all the elements in this subspace are prohibited residues for θ. Since θ drives exactly 2^{k^*-k+1} outputs, there are at most 2^{k^*-k+1} distinct subspaces of dimensions less than or equal to $(k-1)$ whose elements are prohibited residues for θ.

Lemma 2 states that S is composed of at least $(2^{k^*-k+1}+1)$ distinct subspaces of dimensions less than or equal to $(k-1)$. Thus the total number of prohibited residues for θ is less than 2^{k^*}. Hence θ can be assigned a proper residue from S. Since θ is arbitrary, all inputs in $I_{2^{k^*-k+1}}$ can be assigned proper residues from S.

Similarly, it can be shown that the total number of prohibited residues is less than 2^{k^*} for any input in I_j for all $j < 2^{k^*-k+1}$. Hence all inputs in I_j for all $j \leq 2^{k^*-k+1}$ can be assigned proper residues by k^* test signals. □

Lemma 6 *For an (n, m, k) circuit with $m < 6$, let k independent test signals be sufficient to assign proper residues for all inputs in I_5 and I_4. Then these test signals are also sufficient to assign proper residues for all inputs in I_3.*

Proof: Let S be the k-dimensional space spanned by the k independent test signals. Assume that all inputs in I_5 and I_4 have been assigned proper residues from S. We shall show that all inputs in I_3 can also be assigned proper residues from S for any $(n, 5, k)$ circuit. The lemma follows for any (n, m, k) circuit with $m < 6$.

Let the five outputs of the circuit be denoted as O_1, O_2, O_3, O_4 and O_5 respectively. Let us sequentially assign proper residues to inputs in I_3 and assume that input $\theta \in I_3$ is under consideration for residue assignment. Let θ drive outputs O_1, O_2 and O_3. Each of these three outputs can have at most $(k-1)$ inputs that are already assigned proper residues. Let k_1, k_2 and k_3 inputs driving O_1, O_2 and O_3, respectively, be already assigned proper residues. Without loss of generality, assume $k_1 \leq k_2 \leq k_3 \leq (k-1)$. Let S_i $(i = 1, 2, 3)$ be the subspace spanned by the residues assigned to k_i inputs driving O_i. The subspaces S_1, S_2 and S_3 are of dimensions k_1, k_2 and k_3 respectively. The elements in $S_1 \cup S_2 \cup S_3$ are prohibited residues for θ. As per Lemma 1, we have

$$\begin{aligned} |S_1 \cap S_3| &\geq 2^{k_1+k_3-k} \\ |S_2 \cap S_3| &\geq 2^{k_2+k_3-k} \end{aligned}$$

Hence the total number of prohibited residues for θ is given by

$$|S_1 \cup S_2 \cup S_3| \leq 2^{k_1} + 2^{k_2} + 2^{k_3} - 2^{k_1+k_3-k} - 2^{k_2+k_3-k} \leq 2^k \qquad (7)$$

The equality in Equation 7 is satisfied only for $k_1 = k_2 = k_3 = (k-1)$. That means any input θ in I_3 can be assigned a proper residue from S, provided the values of k_1, k_2 and k_3

are not simultaneously equal to $(k-1)$. Since the circuit has only five outputs, there can be at most only one input in I_3 with $k_1 = k_2 = k_3 = (k-1)$. Let θ^* be the unique input in I_3 satisfying the condition $k_1 = k_2 = k_3 = (k-1)$. Therefore all the inputs in I_3 except θ^* can be assigned proper residues from S. Input θ^* appears in O_1, O_2 and O_3 as shown below.

$$\begin{array}{lll} O_1 & :: & \cdots\cdots\ \theta_1' \cdots\cdots \theta^* \\ O_2 & :: & \cdots\cdots\ \theta_2' \cdots\cdots \theta^* \\ O_3 & :: & \cdots\cdots\ \theta_3' \cdots\cdots \theta^* \\ O_4 & :: & \cdots\ \theta_1'\ \theta_2'\ \theta_3' \cdots \\ O_5 & :: & \cdots\ \theta_1'\ \theta_2'\ \theta_3' \cdots \end{array}$$

Let $T = S_1 \cup S_2 \cup S_3$. Input θ^* can be assigned a proper residue as long as $T \subset S$. Let $T = S$ under a residue assignment for inputs in $I_3 - \{\theta^*\}$ so that θ^* cannot be assigned a proper residue from S. We shall show that there exists another residue assignment for inputs in $I_3 - \{\theta^*\}$ such that $T \subset S$ and θ^* can also be assigned a proper residue from S.

Let R_1, R_2 and R_3 be the sets of $(k-1)$ residues assigned to the remaining $(k-1)$ inputs driving O_1, O_2 and O_3 respectively. The $(k-1)$-dimensional subspaces S_1, S_2 and S_3 are spanned by the sets R_1, R_2 and R_3 respectively. Let $S^* = S_1 \cap S_2 \cap S_3$. Since $T = S$ by our assumption, S^* is a $(k-2)$-dimensional subspace as per Lemma 3. Since $T = S$, there exists residues r_1, r_2 and r_3 unique to R_1, R_2 and R_3, respectively, such that $r_1 \notin S_2 \cup S_3$, $r_2 \notin S_1 \cup S_3$ and $r_3 \notin S_1 \cup S_2$. Following similar arguments given in the proof of Lemma 3, we can show that

$$\begin{aligned} S_1 &= S^* \cup \{r_1 + s \mid \forall s \in S^*\} \\ S_2 &= S^* \cup \{r_2 + s \mid \forall s \in S^*\} \\ S_3 &= S^* \cup \{r_3 + s \mid \forall s \in S^*\} \end{aligned}$$

Let $T_1 = \{r_1 + s \mid \forall s \in S^*\}$, $T_2 = \{r_2 + s \mid \forall s \in S^*\}$ and $T_3 = \{r_3 + s \mid \forall s \in S^*\}$. Since $T = S$, Lemma 3 implies that the set $\{r_1 + r_2 + s \mid \forall s \in S^*\}$ must be equal to T_3. In other words, r_3 must be equal to $(r_1 + r_2 + s^*)$ where $s^* \in S^*$.

Let inputs θ_1', θ_2' and θ_3' drive outputs O_1, O_2 and O_3 (as shown above) and be assigned the residues r_1, r_2 and r_3 respectively. Since the residues r_1, r_2 and r_3 are unique to R_1, R_2 and R_3, the inputs θ_1', θ_2' and θ_3' are also unique to O_1, O_2 and O_3 respectively. Inputs θ_1', θ_2' and θ_3' cannot belong to I_4 or I_5 and hence must belong to I_3. This implies the last two outputs O_4 and O_5 must be driven by all three inputs θ_1', θ_2' and θ_3' as shown above.

We shall show that the residue $r_3' = (r_1 + s^*)$ instead of $r_3 = (r_1 + r_2 + s^*)$ is still a proper residue for input θ_3'. Input θ_3' drives O_3, O_4 and O_5. Let us consider O_3 and show that r_3' can also be assigned as a proper residue for θ_3' instead of r_3. Since $r_1 \notin S_3$, we infer that $r_3' \notin S_3$.

Since $L(R_3 - \{r_3\}) \subset S_3$, we know that $r_3' \notin L(R_3 - \{r_3\})$. Hence r_3' is linearly independent with the residues in $(R_3 - \{r_3\})$ and r_3' instead of r_3 can be assigned as a proper residue for θ_3' as far as O_3 is concerned. Next let us consider O_4. Let R_4 be the set of linearly independent residues assigned to the inputs driving O_4. Since the inputs θ_1', θ_2' and θ_3' appear together in O_4, $\{r_1, r_2, r_3\} \subset R_4$. Since $r_3 \notin L(R_4 - \{r_3\})$, $r_2 \in L(R_4 - \{r_3\})$ and $r_3' = (r_3 + r_2)$, we infer that $r_3' \notin L(R_4 - \{r_3\})$. Therefore r_3' instead of r_3 can be assigned as a proper residue for θ_3' as far as O_4 is concerned. Similarly, it can be shown that r_3' instead of r_3 can be assigned as a proper residue for θ_3' as far as O_5 is concerned. Thus we reassign r_3' instead of r_3 as a proper residue for θ_3'.

Let $R_3' = R_3 - \{r_3\} + \{r_3'\}$ and $S_3' = L(R_3')$. By the reassignment process R_3' instead of R_3 becomes the set of $(k-1)$ residues assigned to the remaining $(k-1)$ inputs driving O_3. Since $r_2 \notin L(R_3) = S_3$, we know that $r_2 \notin L(R_3 - \{r_3\})$. Since $r_2 \notin L(R_3 - \{r_3\})$, $r_3 \notin L(R_3 - \{r_3\})$ and $r_3' = r_2 + r_3$, we infer that $r_2 \notin L(R_3') = S_3'$ and $r_3 \notin L(R_3') = S_3'$. Since $r_3 \notin S_1 \cup S_2$, we infer $r_3 \notin S_1 \cup S_2 \cup S_3'$ and therefore θ^* can be assigned r_3 as a proper residue. Thus all inputs in I_3 can be assigned proper residues from S. □

Theorem 3 *Any (n, m, k) circuit with $m < 6$ is a MTC circuit.*

Proof: Consider any (n, m, k) circuit with $m < 6$. Since the maximum cone size of the circuit is k, it requires at least k independent test signals for pseudo-exhaustive testing. Let S be the k-dimensional space spanned by the basis $B = \{1, x, x^2, \ldots, x^{k-1}\}$ (representing k independent test signals). We only need to show that all inputs in I_5 and I_4 can be assigned proper residues from S. Inputs in I_3 are guaranteed of proper residues from S as per Lemma 6. Inputs in I_2 and I_1 are guaranteed of proper residues from S as per Corollary 2.

Case $m = 4$: Let $|I_4| = k_4$. Since each output is driven by all inputs in I_4 and the maximum cone size for the circuit is k, $k_4 \leq k$. Hence all inputs in I_4 can be assigned proper residues by selecting k_4 elements $\{1, x, x^2, \ldots, x^{k_4-1}\}$ of B. Hence the circuit is a MTC circuit.

Case $m = 5$: Let $|I_5| = k_5$ and $|I_4| = k_4$. Since each output is driven by all inputs in I_5 and the maximum cone size for the circuit is k, $k_5 \leq k$. Inputs in I_5 can be assigned proper residues by selecting k_5 elements $\{1, x, x^2, \ldots, x^{k_5-1}\}$ of B. We shall consider inputs in I_4 and assign proper residues from the subspace spanned by the remaining $(k - k_5)$ elements $\{x^{k_5}, x^{k_5+1}, \ldots, x^{k-1}\}$ of B.

Let the five outputs of the circuit be denoted as O_1, O_2, O_3, O_4 and O_5 respectively. Partition the inputs in I_4 into five subsets $I_{4,1}, I_{4,2}, \ldots, I_{4,5}$ such that $I_{4,i} = \{$inputs that do not drive O_i $\}$ $(i = 1, 2, \ldots, 5)$. Let $|I_{4,i}| = k_{4,i}$ $(i = 1, 2, \ldots, 5)$. Without loss of generality, assume that $I_{4,5}$ is the smallest subset among the five subsets. Select one input (say θ_i') from each $I_{4,i}$ and form the input set $I = \{\theta_1', \theta_2', \theta_3', \theta_4', \theta_5'\}$. Note that only four inputs from I

appear together in any output as shown below.

$$\begin{array}{lll} O_1 & :: & \cdots\cdots\ \theta'_5\ \theta'_4\ \theta'_3\ \theta'_2 \cdots\cdots \\ O_2 & :: & \cdots\cdots\ \theta'_5\ \theta'_4\ \theta'_3\ \theta'_1 \cdots\cdots \\ O_3 & :: & \cdots\cdots\ \theta'_5\ \theta'_4\ \theta'_2\ \theta'_1 \cdots\cdots \\ O_4 & :: & \cdots\cdots\ \theta'_5\ \theta'_3\ \theta'_2\ \theta'_1 \cdots\cdots \\ O_5 & :: & \cdots\cdots\ \theta'_4\ \theta'_3\ \theta'_2\ \theta'_1 \cdots\cdots \end{array}$$

The inputs in I completely occupy four columns in the cone dependencies. Consider a four dimensional subspace spanned by the four elements $\{x^{k_5}, x^{k_5+1}, x^{k_5+2}, x^{k_5+3}\}$ of B. We shall assign the five residues $\{x^{k_5},\ x^{k_5+1},\ x^{k_5+2},\ x^{k_5+3},\ x^{k_5} + x^{k_5+1} + x^{k_5+2} + x^{k_5+3}\}$ to the five inputs in I. Since only any four inputs from I appear together in any output, this assignment ensures proper residues to all inputs in I. This process is repeated until all inputs are selected from $I_{4,5}$. Thus $5k_{4,5}$ inputs in I_4 are assigned proper residues from the subspace spanned by $4k_{4,5}$ elements of B.

The remaining $(k_4 - 5k_{4,5})$ inputs in I_4 need to be assigned proper residues from the subspace spanned by the remaining $(k - k_5 - 4k_{4,5})$ elements in B. Since none of the remaining inputs in I_4 belong to $I_{4,5}$, all of them drive O_5. Also all k_5 inputs in I_5 and $4k_{4,5}$ inputs in I_4 drive O_5. Hence the total number of inputs driving O_5 must be greater than or equal to $(k_5 + 4k_{4,5} + k_4 - 5k_{4,5}) = (k_5 + k_4 - k_{4,5})$. Since the maximum cone size for the circuit is k, we have $k \geq k_5 + k_4 - k_{4,5}$ which implies $k - k_5 - 4k_{4,5} \geq k_4 - 5k_{4,5}$. Hence we have the number of remaining elements in B is greater than or equal to the number of remaining inputs in I_4 and we can assign each of the remaining elements in B to each of the remaining inputs in I_4. Thus all inputs in I_5 and I_4 can be assigned proper residues from S. Hence the circuit is a MTC circuit. □

Theorem 6 *For an (n, m, k) circuit, let $p_{i,j}$, p_i^* and $f_{i,j}$ be the circuit parameters (defined earlier) characterizing the cone dependencies. Let k^* be the smallest number satisfying the following inequality for all inputs θ_i, $1 \leq i \leq n$.*

$$\lceil 2^{2p_i^*-2-k^*} \rceil + \sum_{j=1}^{m} f_{i,j}\{2^{p_{i,j}-1} - \lceil 2^{p_i^*+p_{i,j}-2-k^*} \rceil\} < 2^{k^*} \tag{8}$$

Then k^* independent test signals are sufficient for pseudo-exhaustive testing of the circuit.

Proof: An (n, m, k) circuit can be pseudo-exhaustively tested by k^* independent test signals if all inputs can be assigned proper residues from the k^*-dimensional space (say S). Inputs θ_1 through θ_n are considered in succession for residue assignment. Let us assume that inputs θ_1 through θ_{i-1} have been successfully assigned proper residues and input θ_i is under consideration. We shall explore the feasibility of assigning a proper residue for θ_i from S.

Consider an output O_{j^*} in which θ_i appears at position p_i^* among the input dependencies. For this output, θ_i appears along with (p_i^*-1) inputs that have been already assigned proper residues. These (p_i^*-1) residues span a (p_i^*-1)-dimensional subspace (say S_{j^*}) and all the elements in this subspace are prohibited residues for θ_i. Consider another output O_j with $p_{i,j} > 0$ and hence $f_{i,j} = 1$. For O_j, θ_i appears along with $(p_{i,j}-1)$ inputs that have been already assigned proper residues. These residues span a $(p_{i,j}-1)$-dimensional space (say S_j) and all the elements in this space are prohibited residues for θ_i. From Lemma 1, we know that subspaces S_{j^*} and S_j have at least $\lceil 2^{p_i^*+p_{i,j}-2-k^*} \rceil$ common elements. Hence the number of prohibited residues for θ_i due to O_{j^*} and O_j is given by

$$\begin{aligned} |S_{j^*} \cup S_j| &= |S_{j^*}| + |S_j| - |S_{j^*} \cap S_j| \\ &\leq 2^{p_i^*-1} + 2^{p_{i,j}-1} - \lceil 2^{p_i^*+p_{i,j}-2-k^*} \rceil \end{aligned}$$

Considering all outputs driven by θ_i, the total number of prohibited residues for θ_i is given by

$$\begin{aligned} |\bigcup_{j=1;p_{i,j}>0}^{m} S_j| &\leq |S_{j^*}| + \sum_{j=1;j\neq j^*}^{m} f_{i,j}\{|S_j| - |S_{j^*} \cap S_j|\} \\ &\leq 2^{p_i^*-1} + \sum_{j=1;j\neq j^*}^{m} f_{i,j}\{2^{p_{i,j}-1} - \lceil 2^{p_i^*+p_{i,j}-2-k^*} \rceil\} \\ &= \lceil 2^{2p_i^*-2-k^*} \rceil + \sum_{j=1}^{m} f_{i,j}\{2^{p_{i,j}-1} - \lceil 2^{p_i^*+p_{i,j}-2-k^*} \rceil\} \end{aligned}$$

Thus the LHS expression of Equation 4 gives an upper bound on the total number of prohibited residues for θ_i. As long as this expression is less than 2^{k^*}, a proper residue from S is guaranteed for θ_i. Hence the satisfiability of Equation 4 for all inputs guarantees the existence of proper residues for all inputs in the space generated by k^* independent test signals. □

Theorem 8 *Algorithm XORBound is of polynomial complexity and determines the tightest possible bound on the number of test signals that can be achieved using Theorem 6.*

Proof: Let I denote the set of circuit inputs. Let k^* be the number of test signals considered during some iteration of the algorithm *XORBound*. Assume that a subset of inputs (say I_1 with $|I_1| = n_1$) are not assigned indices after the completion of the *while* loop. This implies that all the $(n-n_1)$ inputs in $I - I_1$ have been successfully assigned indices greater than n_1. Let Π_1 denote the partial permutation of circuit inputs in which n_1 inputs are not assigned indices and the remaining $(n-n_1)$ inputs are assigned indices greater than n_1. The *while* loop must have terminated after determining that none of the inputs in I_1 can be assigned the index n_1. We claim that I_1 is the minimum set under any partial permutation of inputs and shall prove the claim as follows.

Let Π_2 denote another partial permutation of circuit inputs that results in the *minimum* subset of inputs (say I_2 with $|I_2| = n_2$) such that (1) none of the n_2 inputs in I_2 can be assigned the index n_2 and satisfy Equation 4 and (2) all of the $(n - n_2)$ inputs in $I - I_2$ are assigned proper indices greater than n_2 and satisfy Equation 4.

We shall prove that $I_1 = I_2$ by contradiction. Let $I_1' = I_1 - (I_1 \cap I_2)$ and consider an input $\theta_1 \in I_1'$. Input θ_1 does not satisfy Equation 4 with index n_1 under Π_1 but satisfies the equation with an index greater than n_2 under Π_2. Hence for some output (say O_j), the $p_{1,j}$ value for θ_1 under Π_1 must be greater than the $p_{1,j}$ value under Π_2. This is possible only if there exists another input (say θ_2) that appears before θ_1 among the dependencies for O_j under Π_1 and appears after θ_1 among the dependencies for O_j under Π_2. This implies (1) $\theta_2 \in I_1$ under Π_1; (2) $\theta_2 \notin I_2$ under Π_2 and (3) θ_2 being assigned an index greater than that of θ_1 under Π_2. Repeating the argument for $\theta_2 \in I_1'$ leads to a third input $\theta_3 \in I_1'$ and θ_3 being assigned an index greater than that of θ_2 under Π_2. The argument can thus be repeated for all inputs in I_1'. The argument fails for the last input in I_1' since there are no more inputs left in I_1'. This is a contradiction. Hence there exists no $\theta_1 \in I_1'$ and $I_1 \subseteq I_2$. Since I_2 is the minimum set by definition, $I_1 = I_2$.

Thus algorithm *XORBound* determines the minimum set of inputs that cannot be assigned indices and iterates with an increment to the number of test signals. Thus the algorithm determines the tightest possible bound on the number of test signals that can be achieved using Theorem 6. □

Theorem 10 *For an (n, m, k) circuit, let $p_{i,j}$ and $f_{i,j}$ be the circuit parameters (defined earlier) characterizing the cone dependencies. Let k^* be the smallest number satisfying the following inequality*

$$\sum_{j=1}^{m} \sum_{i=k^*}^{n} i \times f_{i,j}(2^{p_{i,j}-1} - 1) < \Phi(2^{k^*} - 1) \approx 2^{k^*} - 1 \tag{9}$$

Then a simple LFSR/SR based on a degree k^* primitive polynomial is sufficient for pseudo-exhaustive testing of the circuit.

Proof: (The following is an extension of the arguments presented in [5]). Let us consider output O_j and determine an upper bound on the number of inapplicable primitive polynomials of degree k^* for this output. Let the input dependencies of O_j contain input θ_i in $p_{i,j}$th position. Let inputs $\theta_{i_1}, \theta_{i_2}, \ldots, \theta_{i_{p_{i,j}-1}}$ appear in positions 1 through $(p_{i,j} - 1)$ respectively for this output. An applicable primitive polynomial $P(x)$ of degree k^* should ensure that the residues $\{x^{i_1}$ *mod* $P(x)$, x^{i_2} *mod* $P(x)$, ..., $x^{i_{p_{i,j}-1}}$ *mod* $P(x)$, x^i *mod* $P(x)\}$ are linearly independent. In other words, each polynomial $Q(x)$ of the form $x^i + \sum_{q=1}^{p_{i,j}-1} a_q x^{i_q}$ (where $a_q = 0$ *or* 1 and not all of them are zeros) must not be divisible by $P(x)$. There are $(2^{p_{i,j}-1} - 1)$ such polynomials $Q(x)$ of degree i. Each one of the polynomials $Q(x)$ is divisible by no more than i/k^* distinct primitive polynomials of degree k^*. Therefore an upper bound on the number of inapplicable primitive polynomials of degree k^* that may assign linearly dependent residues to some inputs in the set $\{\theta_{i_1}, \theta_{i_2}, \ldots, \theta_{i_{p_{i,j}-1}}, \theta_i\}$ driving O_j is given by the expression $E_{i,j} = (i/k^*)(2^{p_{i,j}-1} - 1)$. Summing up $E_{i,j}$ for all values of $i \geq k^*$ yields an upper bound on the number of inapplicable primitive polynomials of degree k^* for O_j. There is no need to consider any $Q(x)$ polynomial of degree less than k^* since the primitive polynomial $P(x)$ is of degree k^*. The boolean variable $f_{i,j}$ ensures that only those inputs that drive O_j are considered.

Again summing up for all values of j yields an upper bound on the total number of inapplicable primitive polynomials of degree k^* for all circuit outputs. This double summation is given by the LHS expression of Equation 5.

Theorem 9 states that the total number of primitive polynomials of degree k^* is given by $\Phi(2^{k^*} - 1)/k^*$. To ensure that the total number of inapplicable primitive polynomials of degree k^* is less than the total number of primitive polynomials of degree k^*, we must have

$$\begin{aligned} \sum_{j=1}^{m} \sum_{i=k^*}^{n} i/k^* \times f_{i,j}(2^{p_{i,j}-1} - 1) &< \Phi(2^{k^*} - 1)/k^* \\ \Rightarrow \sum_{j=1}^{m} \sum_{i=k^*}^{n} i \times f_{i,j}(2^{p_{i,j}-1} - 1) &< \Phi(2^{k^*} - 1) \approx 2^{k^*} - 1 \end{aligned}$$

Thus the satisfiability of Equation 5 guarantees a primitive polynomial of degree k^* applicable to all outputs. □