

## Test Vector Optimization Using Pcofan-Poframe Partitioning

P. PattunnaRajam<sup>1, \*</sup>, Reeba korah<sup>2</sup> and G. Maria Kalavathy<sup>3</sup>

**Abstract:** This paper presents an automated POCOFAN-POFRAME algorithm that partitions large combinational digital VLSI circuits for pseudo exhaustive testing. In this paper, a simulation framework and partitioning technique are presented to guide VLSI circuits to work under with fewer test vectors in order to reduce testing time and to develop VLSI circuit designs. This framework utilizes two methods of partitioning Primary Output Cone Fanout Partitioning (POCOFAN) and POFRAME partitioning to determine number of test vectors in the circuit. The key role of partitioning is to identify reconvergent fanout branch pairs and the optimal value of primary input node  $N$  and fanout  $F$  partitioning using I-PIFAN algorithm. The number of reconvergent fanout and its locations are critical for testing of VLSI circuits and design for testability. Hence, their selection is crucial in order to optimize system performance and reliability. In the present work, the design constraints of the partitioned circuit considered for optimization includes critical path delay and test time. POCOFAN-POFRAME algorithm uses the parameters with optimal values of circuits maximum primary input cone size ( $N$ ) and minimum fan-out value ( $F$ ) to determine the number of test vectors, number of partitions and its locations. The ISCAS'85 benchmark circuits have been successfully partitioned, the test results of C499 shows 45% reduction in the test vectors and the experimental results are compared with other partitioning methods, our algorithm makes fewer test vectors.

**Keywords:** Pseudo exhaustive testing, POCOFAN (Primary Output Cone Fanout Partitioning), POFRAME partitioning, combinational digital VLSI circuit testing, critical path delay, testing time, design for testability.

### 1 Introduction

The recent advances in fabrication technology have led to tremendous increase in the count of gates that can be accommodated in digital ICs. The complexity of modern digital VLSI a (Very Large Scale Integration) circuit is developing rapidly in accordance with Moore's Law. The complexity of the design degrades the difficult testing problem. This brings about the need for an alternative method of testing VLSI circuits. This in turn

---

<sup>1</sup> Department of ECE, Anna University, Chennai, Tamil Nadu, India. pattunnarajamphd@gmail.com

<sup>2</sup> Department of ECE, Alliance University, Bangalore, Karnataka, India.

reeba26in@gmail.com

<sup>3</sup> Department of CSE, St. Joseph's College of Engineering, Chennai, Tamil Nadu, India.

maria\_kalavathy@yahoo.co.in

\*Corresponding author: P. PattunnaRajam. Email: pattunnarajamphd@gmail.com.

increases the testing time. The testing time associated with exhaustive testing increases exponentially with the number of inputs to the circuit. The number of test vectors required to test the circuit are based on number of primary inputs. Accordingly, many techniques have been developed for test pattern generation and test vector optimization. In exhaustive testing, the combinational circuit is tested with all possible  $2^n$  test patterns, where  $n$  is the number of inputs. Though it guarantees detection of all detectable non-transitional faults, its run time becomes too high for circuits with  $n > 20$ . The testing time associated with exhaustive testing increases exponentially with the number of inputs to the circuit. For circuit with a large number of inputs, exhaustive testing is very time consuming and may not be practical. Pseudo-exhaustive (PE) testing method retains almost all the advantages of exhaustive testing, but at a much reduced complexity.

In Pseudo-exhaustive testing, the combinational circuits are tested with  $2^n$  test patterns based on the optimal value of primary input node  $N$  and fanout  $F$  value. In our proposed method, an output cone with maximum fanout branch pair of the circuit is partitioned and exhaustively tested. Output cone is the portion of the circuit whose signals are reachable by backward trace of circuit topology, till primary inputs are reached, starting from a primary output. Edge partitioning at the primary output cone partitioning reduces the number of branches from the reconvergent fanout stems and changes the optimal value of  $N$  and  $F$ . Thus a test vector reduces by fixing the optimal value of primary input node  $N$  and fanout  $F$  partitioning.

Though, if the circuit is partitioned into separate partitions without any strategy it may lead to additional test vectors. Goel [Goel (1993)] valued the cost of testing by increasing more number of gates in the circuit. The time complexity to generate test vectors exhaustively by  $2^n$  input combinations is  $O(n^3)$ , where  $n$  is the number of gates in the circuit. It provides an in-depth test with much testing time and with many inputs. One alternative, proposed by McCluskey et al. [McCluskey and Bozorgui-Nesbat (1981)] is pseudo exhaustive testing. This requires that a circuit be divided into partitions, where each partition has a limit on the number of primary inputs which considerably reduces the necessary number of test vectors. Each partition is comparable in nature and can then be tested in a relatively small amount of time. Partitioning techniques can be broadly separated into two categories:

- Invasive approach
- Non-invasive approach

The invasive approach inserts DFT elements to separates one portion of a circuit from another. The example for an invasive approach is a multiplexer partitioning, which enables sub circuit testing using proper control signals set to multiplexers or DFT circuit. The non-invasive approach achieves sub circuit separation without any additional hardware components. Paths from the primary inputs to the sub circuit inputs and paths from the sub circuit outputs to the primary outputs must be sensitized. Using these paths, each sub circuit can be tested as an isolated sub block with more testing time.

The underpinning concept of the present work is to solve complex problems in VLSI design, by partitioning. Though many partitioning algorithms have been proposed, very few works analyse the partitioning process for optimizing system design constraints, from a testing. None of the previous works have analyzed the effect of edge partitioning on the

reconvergent fanout branch pairs is to reduce test vectors based on the optimal value of  $N$  and  $F$ .

The objective of this research study is to design testable reliable VLSI circuits using an optimized partitioning approach, in order to achieve simultaneous optimization of number of partitions, number of test vectors, and critical path delay (CPD). This optimization problem is proven to be NP-Hard. In this study, circuit partitioning is implemented using the improved primary input primary output fan-out (IPIFAN) algorithm. IPIFAN algorithm is used to determine the partitioned points based on two constant inputs: maximum node fan-in size  $N$  and minimum partitioning fan-out value  $F$ .

IPIFAN is hybridized with POCOFAN-POFRAME algorithm with adaptive parameter variations to enhance the search process for optimal  $N$  and  $F$  values, in order to obtain most optimal partitioned circuit. IPIFAN is based upon pseudo-exhaustive testing method. This testing method provides nearly reduced number of test vectors. Parallel testing of partitioned cones is achieved by inserting suitable invasive DFT logic at the partition points, to apply pseudo exhaustive test patterns.

Literature reveals that the previous optimization has been done using Particle Swarm Optimization (PSO) and exhaustive methods. To the best of our knowledge, edge partitioning has not been proposed so far for optimization of the partitioning problem. In this study, an analogy is drawn between partitioning the branches from the reconvergent fanout stem from the primary output and solving the partitioning problem to obtain the best optimal solution. Knowledge of the reconvergent fanout structure of a circuit is taken into account to produce a more accurate testability measure or assist in the test generation process.

In this article, a pseudo exhaustive cone partitioning approach is proposed to reduce the number of test vectors. I-PIFAN algorithm is used to find optimal values of  $N$  and  $F$ . The rest of the paper is organized as follows. In Section 2, the existing method using I-PIFAN partitioning algorithm is presented. The problem approach and the partitioning strategy of the proposed method are discussed in Section 3 and 4. Experimental results are shown in Section 5 and the paper is concluded in Section 6.

## **2 Related work**

Gruning et al. [Gruning, Mahlstedt, Daehn et al. (1990)] proposed an efficient cone oriented circuit partitioning method which significantly speeds up automatic test pattern generation for combinational circuits. The advantages gained by partitioning method are based on the increase in the number of dominators in the circuit graph. Srinivasan et al. [Srinivasan, Gupta and Breuer (1993)] presented a hill climbing heuristic procedure to partition large circuit into small circuit based on the optimal solution by solving ILP formulation. The heuristic measures adopted are not appropriate in guiding the search space. Jone et al. [Jone and Papachristou (1995)] have reported a graph theoretic model of VLSI circuits, and an algorithm for pseudo-exhaustive testing based on sub-circuit modification technique. The objective was to generate test patterns of limited length [Shaer (2005)]. Fault oriented pseudo exhaustive technique guarantees the number of unmodeled combinational faults (e.g. bridging faults) within the cone. As the cone sizes in the test mode become smaller, the number of bridging faults that are detected by

pseudo exhaustive testing also decreases. However, this increases the design effort and the number of stages in the resulting TPG [Chen and Gupta (1998)]. All detectable multiple stuck-at faults and all detectable combinational faults within individual cones are tested in pseudo exhaustive testing [Srinivasan, Gupta and Breuer (2000)]. An output cone that depends on the inputs is exhaustively tested if and only if the residues are linearly independent. LFSR/XOR circuit has been constructed as per the design procedure incur high area overhead due to the XOR network, but generate minimal pseudo exhaustive test sets by utilizing information about cone dependencies. Shaer [Shaer (2000)] developed a general partitioning algorithm (PIFAN), based on the Primary Input cone and Fanout values of each node. While this algorithm shows significant improvement over previously suggested methods, it lacks the flexibility to optimally partition all circuits. PIFAN algorithm was automated and improved by Shaer et al. [Shaer and Dib (2002); Shaer, Landis and Al-Arian (2000)], for all circuits by considering the effect of critical path called I-PIFAN (Improved Primary Input Fanout). I-PIFAN is based on maximum fanin size  $N$  and minimum fanout value  $F$ . The circuit is completely partitioned for each combination of  $N$  and  $F$ , from which best result of number of test vectors, number of cuts and optimal values of  $N$  and  $F$  are selected. The disadvantages of this approach are the increased computation time due to deterministic exhaustive search and less optimal results. Also, optimal results are found only if  $N$  and  $F$  values lie in the specified range.

Particle swarm optimization (PSO) is an evolutionary computation technique developed by Kennedy et al. [Kennedy and Eberhart (1995)]. The potential solutions, called “particles”, fly around in a multi-dimensional search space, to discover an optimal, or sub-optimal solution. Each particle accelerated with random velocity in an n-dimensional space. The position and velocity of each particle in n dimension space is updated based on its previous velocity, the previous best particle location ( $p_{id}$  or  $p_{best}$ ) and the previous global best location of a particle in the population ( $p_{gd}$  or  $g_{best}$ ). The basic concept of PSO lies in accelerating each particle towards its  $p_{best}$  and  $g_{best}$  locations at each time step.

Clerc et al. [Clerc and Kennedy (2002)] have addressed the stability and convergence issues of PSO. The optimal solution of PSO-PIFAN exists at any range there is no specified range of  $N$  and  $F$  values [Venayagamoorthy, Smith and Singhal (2007)] presented a PSO-PIFAN algorithm using swarm intelligence based approach for pseudo exhaustive testing. Here the optimal values of  $N$  and  $F$  is used to determine number of test vectors, number of partitions and critical path delay based on the value of fitness function  $f_1, f_2$  and  $f_3$  in Eqs. (1)-(3) without demanding specific range, where  $f_T, f_P$  and  $f_C$  are the number of test vectors, number of partitions and critical path delay required to do exhaustive testing.

$$f_1 = fp^2 + \sqrt{\frac{f_T^{initial}}{100} + f_T} \quad (1)$$

$$f_2 = fc^2 + \sqrt{\frac{f_C^{initial}}{100} + f_T} \quad (2)$$

$$f_3 = (fp^2 + fc^2) + \sqrt{\frac{f_{c^{initial}}}{100} + f_T} \quad (3)$$

Kumar et al. [Kumar, Bhaskar, Chattopadhyay et al. (2009)] proposed a PSO based pseudo exhaustive approach for circuit partitioning. The results indicate an improvement in number of partitions, but does not include optimization of crucial design constraints of reliability, CPD and test time. Exhaustive test of sub-circuit partitioning deals an attractive alternate to exhaustive approach, which leads to pseudo-exhaustive testing approach. In many cases, each output of  $(n, m, k)$  circuit under test (CUT) depends only on a subset of primary inputs, where 'k' represents the cone size or the maximum dependency among 'm' output cones. For an 'n' input CUT with 'm' outputs, pseudo-exhaustive testing approach involves applying exhaustive test to the m-output cones. If the CUT is partitioned into 'm' cones with cone size 'k', then many of these cones can be tested in parallel, reducing test vector length [Voyiatzis, Gizopoulos and Paschalis (2010)]. To optimize the partitioning process, the hybrid technique of ant colony optimization (ACO) and I-PIFAN method are incorporated with multiple performance characteristics. The reliability of the partitioned circuit was found by using probabilistic gate model (PGM) method for different values of  $N$  and  $F$ . ACO decides the number of partitions and its locations in the circuit by determining the optimal values of circuits maximum primary input cone size ( $N$ ) and minimum fan-out value ( $F$ ) [Jose, Kumar, Hussain et al. (2014)].

Authors in Nejadmoghadam et al. [Nejadmoghadam, Mahani, Kaviani et al. (2014)] proposed a transition probability based method to partition combinational circuits for pseudo exhaustive testing. This method optimizes the  $N$  and  $F$  value to increase the transition probability so as to minimize the number of test vectors, number of partitions and the critical path delay. The fitness function is given in Eq. (4) as:

$$f = \frac{\alpha \times P + \beta \times TV + \gamma \times CP + \text{avg}(\min(\text{contr } 0, \text{contr } 1))}{\text{avg}(\text{I.TP})} \quad (4)$$

where  $\alpha$ ,  $\beta$  and  $\gamma$  the weight of factors and  $P$  is the number of normalized partition and  $TV$  is the number of normalized test vector and  $CP$  is normalized longest path.  $\text{avg}(\min(\text{contr } 0, \text{contr } 1))$  is the average of minimum normalized 0-controllability or 1-controllability. Also,  $\text{avg}(\text{I.TP})$  is the average of improvement in the transition probability of gates which their transition probability is less than the threshold.

In digital circuits half of the nodes in the circuit are fanout nodes. The number of fanout branches in a stem differs from one stem to another. Xu et al. [Xu and Edirisuriya (2004)] proposed an algorithm mathematically by considering average and the maximum value of fanout branches as  $p$  and  $q$  and  $n$  as nodes in the circuit for FOBL (Fanout Branch Line) and RFOBL (Reconvergence Fanout Branch Line). The worst case time complexity achieves for maximum number of fanout branches  $O(n^2 \cdot q^3)$  than the average value  $O(n^2 \cdot p^3)$ . Test vectors are not only optimized by fault based pseudo exhaustive testing and also by test compaction scheme incorporated in Navabi [Navabi (2011)] by identifying compatible test vectors whenever identical child vectors detect different faults. Khera et al. [Khera, Sharma and Gupta (2017)] presented a heuristic fault based approach of independent and essential faults so as to reduce test vector count by extracting the compatible child test vectors from the test compaction scheme and merging them.

### 3 Problem approach

In PSO based optimal partitioning algorithm [Venayagamoorthy, Smith and Singhal (2007)] and I-PIFAN is used as the partitioning approach and the parameters maximum primary input (PI) cone size  $N$  and minimum fan-out (FO) value  $F$  are determined using PSO. There are two conditions for a node to be partitioned: (i) node should not be an inverter or buffer and (ii) PI value of the node should be greater than one. If a partitionable node's FO value is greater than or equal to  $F$ , a partition is inserted.

The problem approaches in the proposed method are:

- The nodes and the nets which has maximum fanout branches at primary output are selected using Cone partitioning.
- No two repeated nodes are carried out between nonoverlap cones.
- Fanout stem with reconvergent fanouts, edge partitioning gives better results than fanout partitioning [Srinivasan, Gupta and Breuer (1993)].
- Circuits without reconvergent fanouts, both edge and fanout partitioning gives best results.
- Exclusion of fanout stem partitioning reduces the area overhead by eliminating the inbuilt LFSR which generates test pattern for the following unpartitionable node.
- Any one of the edges in the cone partitioning must be immediate successor to the primary input node.
- Fewer numbers of node must be between edges and the primary input node.
- The nodal switching activity in the fanout stem is dependent on its fanout branches.
- The first occurrence of primary output with maximum fanout branches is considered as primary output cone partitioning when two or more PO has same number of fanout branches.

### 4 Partitioning strategy

In the combinational logic circuit, each node is labeled with its initial primary input ' $N$ ' and fanout branches ' $F$ ' values. This circuit is partitioned in a breadth first manner using the automation design tool with random ' $N$ ' and ' $F$ ' values fixed at (3, 2) value. There are two phases. Phase I begin from the initial node list and partition made at the output where  $FO \geq 2$ . The PI and FO values of each node are updated. Similarly, phase II also begins at the start of the node list and searches for the first node with a PI greater than  $N$ . The value of  $N$  should be greater than or equal to the maximum fan in of the circuit.

The pseudo code of primary output cone partitioning is as shown in Fig. 1.



- (a) Edge partition the circuit PO with maximum FO branches as cone.
- (b) If edge partitioning is at the FO branch node then
- (c) Partitioning at the optimal value ( $N, F$ ) does not exist.
- (d) Number of test vectors due to  $N$  &  $F$  reduces.
- (e) Repeat (b) to (d) until edge partitioning reaches PO to Pi.
- (f) If two or more PO has same number of fanout branches then
- (g) Priority on first occurrence of PO exists.
- (h) Repeat (b) to (d).
- (i) end.

Figure 1: Primary output cone partitioning

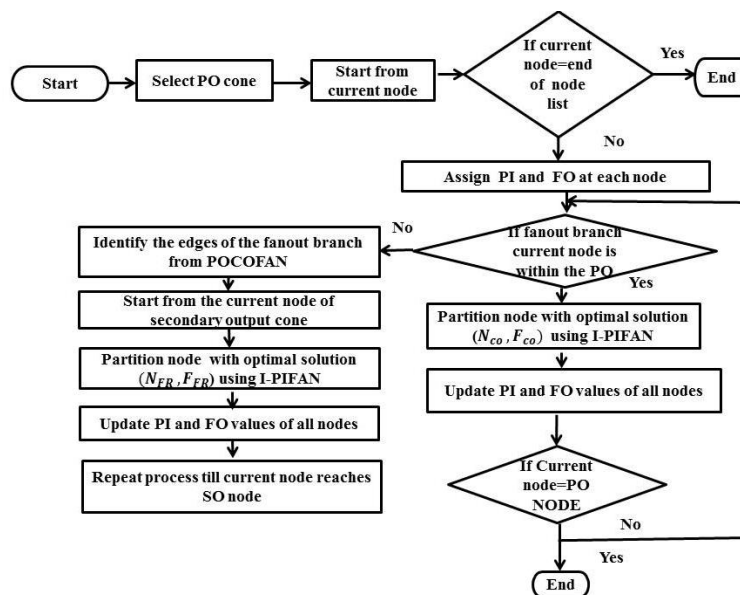


Figure 2: Flowchart of POCOFAN-POFRAME partitioning algorithm

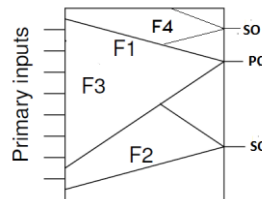
In this section, we outline the steps POCOFAN-POFRAME partitioning algorithm:

- (a) Partition the primary output cone with maximum fanout branches-POCOFAN (Fig. 2)
- (b) Start at the beginning of the primary input node.
- (c) Initialize the optimal solution of primary input node  $N$  and fanout  $F$ .
- (d) Partition the circuit node using I-PIFAN
- (e) If fanout branches is not within PO cone then
- (f) Identify fanout branch as edges-POFRAME
- (g) Repeat (b) to (d) until current node=SO node
- (h) Update the PI and FO values for all nodes
- (i) If current node≠PO node

- (j) Repeat (m)
- (k) end

#### **4.1 Proposed partitioning approach**

The present approach uses the modular cone partitioning method (POCOFAN-POFRAME), which provides highly accurate results with moderate computational complexity. The idea of this approach is primarily inspired by PSO-PIFAN method in order to minimize the number of test vectors, the number of partitions and the critical path simultaneously [Venayagamoorthy, Smith and Singhal (2007)]. In proposed method, large combinational circuits are partitioned into two partitioning circuits, POCOFAN and POCOFRAME partitioning. The single primary cone output  $PO$  with more fanout branches are called POCOFAN partitioning and fewer fanout branches with many secondary outputs are called POFRAME partitioning as shown in Fig. 3. Each sub circuit is further partitioned into sub partition circuits based on separate optimal solution of primary input cone size  $N$  and fanout branches  $F$ .

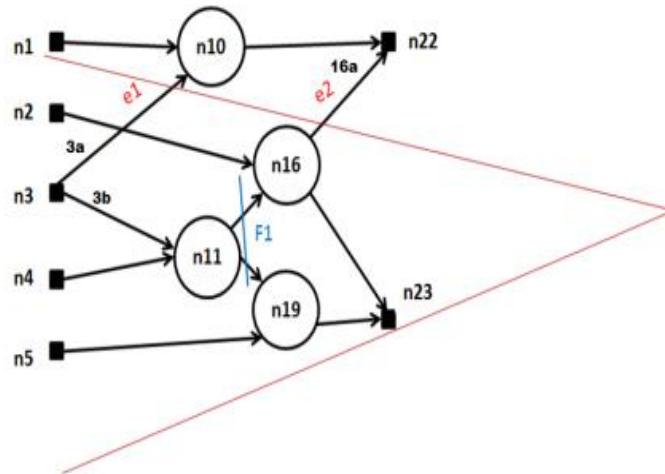


**Figure 3:** POCOFAN-POFRAME partitioning

Partitioning is the process of generating a logic hierarchy to carry out tests efficiently. Partitioning (POCOFAN) based on an analysis of primary output  $PO$ , fanout  $F$  and primary input cone size  $N$ . A cone is the fraction of the combinational logic network which includes all signal lines affecting one output. An efficient cone oriented circuit partitioning method is presented, which significantly speeds up automatic test pattern generation for combinational circuits [Goel (1980)]. The number of inputs in the largest cone is the lower bound and most of the compatible inputs and fanout branches are belong to the same cone with single primary output  $PO$ . As a consequence of having a fanout of one, fanout branches may never represent a primary output since primary outputs always have a fanout of zero. The circuit may have overlapping or non-overlapping cones but specified cone must have single primary output [Kantipudi and Agrawal (2006)].

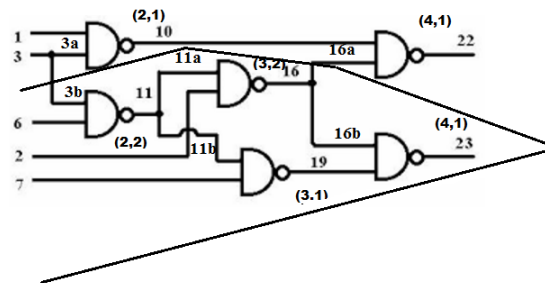
The combinational circuit of C17 is modeled as a circuit graph for algorithmic processing is illustrated in Fig. 4. The node represents inputs, output and gates. The signal lines between the gates are called edges. Two interesting problems arise depending on where partitioning to be placed in a circuit. In the first case, partitioning is allowed to be placed on to the edge where maximum fanout branches of the primary output take place. This is mentioned as edge partitioning, e1 and e2 from node 3a and node 16a. In the second case, partitioning retained on the fanout branches of the gate outputs are called fanout partitioning F1 at node 11.



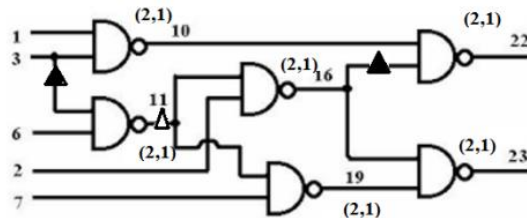


**Figure 4:** Circuit graph partitioning

The circuit has 5 inputs and 2 outputs. The largest cone size is 4. The node 11 and 16 have two fanout branches. Backtracking the gates and nets from primary output to primary input in the form of cone size partitions the circuit into two sub circuits based on maximum number of fanout branches. The nodes are a signal line which connects from primary input to selected primary output are called cardinal. Therefore, the cardinal nodes of primary output 23 are node 23 itself and nodes 2, 3, 3b, 6, 7, 11, 11a, 11b, 16, 16b, 19. The node 3 and 6 are cardinal nodes for fanout node 11. The partitioning noticeable with black lines are the portions of POCOFAN partitioning as shown in Fig. 5.



**Figure 5:** POCOFAN partitioning



**Figure 6:** POCOFAN fanout partitioning

POFRAME partitioning circuit is the frame of the POCOFAN partitioning circuit which carries out fewer number of primary inputs  $N$  and additional number of Fanout branches

originated from the leading partitioning circuit. The nodes 1, 3a, 16a, 10 and 22 are POFFRAME cardinal nodes  $\delta_{FR}$ .

The branches of node 3a and 16a are the edges called as offline inputs is revealed in dark triangle as shown in Fig. 6.

#### 4.2 Influence on cardinal nodes

To show the effect of the partitioning method on the number of cardinal nets a notation of the “cardinal net degree” and the “average cardinal net degree” is introduced [Jone and Papachristou (1995)]. The cardinal degree of a circuit node  $x$  is defined as,

$\delta(x)$  is the number of cardinal nodes of node  $x$ .

The cardinal node degree is considered as a guideline to estimate number of mandatory net assignments in the partitioning circuit from the whole circuit. This supports in speeding up the test pattern generation process.

The average cardinal POCOFAN cone size node degree  $\delta_{c\ avg}$  of a circuit is defined in Eq. (5) as

$$\delta_{c\ avg} = \frac{\sum_{k=1}^N \delta(k)}{N} \quad (5)$$

where  $N$  is the total number of circuit nodes. The value  $\delta_{avg}$  reflects the correlation between the number of cardinal nodes and mandatory node assignments corresponding to the whole circuit.

The average cardinal POCOFAN fanout size node degree  $\delta_{f\ avg}$  of a circuit is defined in Eq. (6) as

$$\delta_{f\ avg} = \frac{\sum_{k=1}^T \delta(k)}{T} \quad (6)$$

Where  $T$  is the number of cardinal POCOFAN cone size nodes.

Consider in Fig. 5, The maximum fanout branches related to the primary output of the circuit is deliberated in POCOFAN partitioning which has large cone size and it holds more number of primary input nodes and maximum fanout branches often need to be partition to reduce test application time [Chen and Gupta (1998); Srinivasan (2000)].

Fig. 7 shows the total number of nodes and the number of nodes of the cone partitioned circuit of ISCAS'85 benchmark circuits. Thereby the cardinal degree of the cone size partitioning (hatched bars) and fanout partitioning (blue bars) of C17, C432, C1355 circuit in the POCOFAN partitioning is approximately 2 times greater than the POFFRAME segment (black bars) is as shown in Fig. 8.

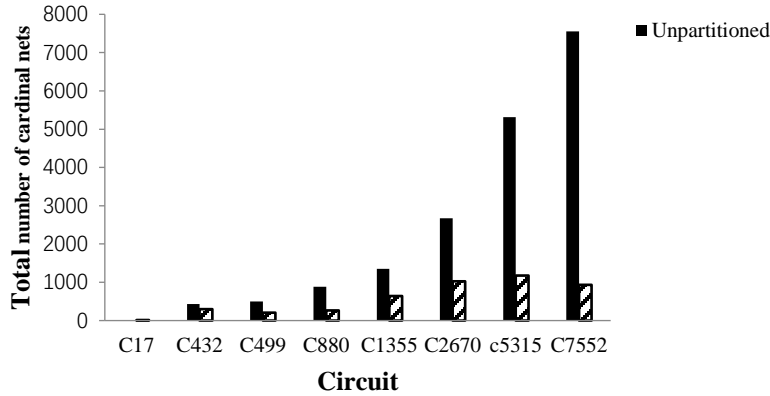


Figure 7: Cardinal nodes of POCOFAN partitioning on ISCAS'85 benchmark circuits

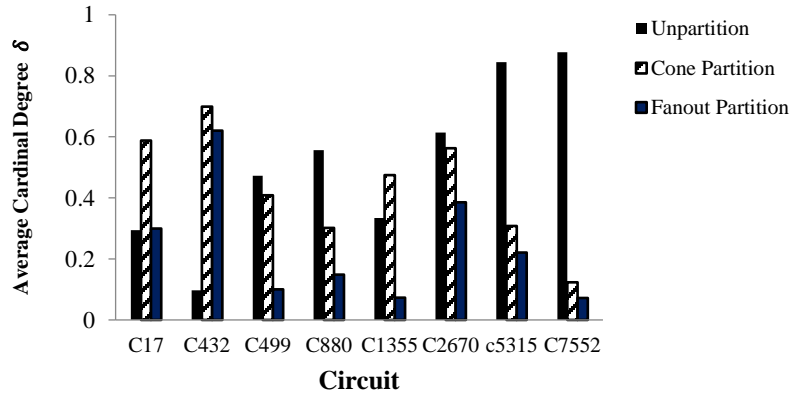


Figure 8: Effect of the POCOFAN partitioning on ISCAS'85 benchmark circuits

#### 4.3 Test vector optimization using partitioning

There are two partitioning method namely POCOFAN and POFRAME is adopted. Each partitioning has a unique optimal solution of primary input cone size  $N$  and fanout branch  $F$  separately as  $(N_{Co}, F_{Co})$  and  $(N_{FR}, F_{FR})$  where  $N_{Co}, F_{Co}$  are the primary input cone size and fanout branches of POCOFAN partitioning and latter for the POFRAME partitioning.

From Fig. 5, The nodes 11a, 16, 19 and 23 are POCOFAN partitioning cardinal nodes  $\delta_{Co}$  and node 11 is the fanout partitioning cardinal node  $\delta_{Fo}$ . The node 11a is one of the primary input of POCOFAN cone circuit after fanout partitioning at node 11 whose optimal solution is  $(2, 2)$  where  $F \geq 2$ . The fanout cardinal node  $\delta_{Fo}$  of node 11 and 16 has no dominator except for itself since there is no node which must be passed unambiguous on the way to a secondary output of node 22. In conventional method [Shaer, Landis and Al-Arian (2000); Shaer and Dib (2002); Shaer (2004); Kumar, Bhaskar, Chattopadhyay et al. (2009)], fanout stem at node 16 is partitioned into two

branches which gives rise to new primary inputs at node 22 and node 23. So, the number of  $2^n$  test patterns is calculated based on optimal solution of  $(N, F)$ . primary input cone size  $N \geq 3$  and fanout branch  $F \geq 2$ . Node 11 and 16 have fanout branches of  $F \geq 2$  where partition takes place. The total number of test patterns at the fanout stem of node 11 and 16 and primary output  $PO$  of node 22 and 23 is 24 ( $2^2+2^2+2^3+2^3$ ).

The nodes 11 and 16 must be passed unambiguous on the way to a primary output of node 23. Fanout Partitioning occurs only at node 11 whose fanout branches of  $F \geq 2$  and not in node 16 hence whose optimal solution of  $(N, F)$  is (2, 2). Consequently fanout partition trails at node 11 whose  $F \geq 2$  is symbolized with hollow triangle is shown in Fig. 6. The optimal solution of  $(N_{FR}, F_{FR})$  in the POFRAME is chosen as maximum primary input cone size  $N$  and minimum fanout branch  $F$ . The total number of test vectors needed pseudo exhaustively to test the whole circuit with 20 ( $2^2+2^3+2^3$ ) test vectors for  $(N_{Co}, F_{Co})$  and  $(N_{FR}, F_{FR})$  is (3, 2) as opposed to the 32 ( $2^5$ ) needed for the circuit.

The pseudo code to determine optimal test vector is shown in Fig. 9.

```
//Finding the Optimal Test vector using (N, F) Values
function optimal TV(cone)
    for i=1 to length(cone)
        Nco(i)=cone(i). fanin;
        Fco(i)=cone(i). fanout;
    end
    f=max(fco);
    for i= 1 to length(Nco)
        n=fin(i);
        [partarray]=split_cone(cone,n,f);
        plen=length(partarray);
        sum=0;
        for p= 1 to plen
            iplen=length(partarray(p). inputs);
            sum=sum+(2iiplen);
        end
        sumarray(i)=sum;
    end
    [minimal_vector_size,index]=min(sumarray);
    return(partarray(index), minimal_vectorsize);
end
```

**Figure 9:** Test vector with optimal  $(N, F)$  values

### 5 Experimental results

The automation tool for circuit partitioning is developed in C++. It reads the node list from the circuit description file (CDF) and then processes the circuit. The data inferred from each line of ISCAS'85 benchmark netlist file are the node address, node name, type of node and the fanin /fan-out values of each node. In POCOFAN partitioning, the netlist is processed and the circuit's nodes are leveled from primary output to primary input by tracing maximum number of fanout branches at the selected primary cone output. The edges of the POCOFAN partitioning connected to the nodes are the part of POFRAME. Two unique values of  $N$  and  $F$  have unique number of partitions. It is observed that it is not certain that the same number of partitions may yield the same number of test vectors. The location of the partition mainly plays a vital role to find number of test vectors for the circuit. It is also verified from experiments that with increase in number of partitions at the edges instead of fanout stem partitioning influences test vector reduction. The hardware overhead moderates merely by fewer number of segmentation cells in order to avoid inbuilt LFSR which generates test pattern generation. It is experimentally found that the level of variations of these parameters is circuit dependent.

In Tab. 1, two unique optimal solutions provide total number of partitions and the number of test vectors. Here the number of partitions and number of test vectors are calculated based on both edge and fanout partitioning nodes.

**Table 1:** Optimal ( $N$ ,  $F$ ) values of ISCAS'85 circuits

ISCAS'85 Circuit	# of PI	# of PO	Proposed Method								# of TVs without partitions
			POCOFAN Partitioning				POFRAME Partitioning				
			$N_{CO}$	$F_{CO}$	# P	#of TVs	$N_I$	$F_{FR}$	# of Par	# of TVs	
Sample	5	2	3	2	1	12	3	2	1	8	32
C432	36	7	16	9	13	7,88,736	8	2	8	1,796	6.87e+10
C499	41	32	6	5	10	66,308	$\frac{1}{2}$	4	2	256	2.20e+12
C880	60	26	18	8	5	10,48,832	9	3	6	2,064	1.15e+18
C1355	41	32	14	5	9	66,560	9	4	3	256	2.20e+12
C2670	233	140	9	11	19	1,51,838	8	7	6	1,408	1.38e+70
C5315	178	123	9	15	15	39,936	5	3	5	136	3.83e+53
C7552	207	108	8	15	10	67,584	7	2	6	644	2.05e+62

The effectiveness of the pseudo exhaustive partitioning presented by Shaer et al. [Shaer, Landis and Al-Arian (2000)] resulted in 2 partitions and 24 test vectors for C17 sample circuit. The partitioning based on POCOFAN-POFRAME resulted in 20 vectors with same number of 2 partitions nevertheless one at the fanout stem and another at the edges. It is worth pointing out that unique optimal solution can yield reduction in the test vectors with minimum hardware overhead. The hardware overhead due to the partitions, testing time and speed have to be taken into consideration to select the optimal values. For example, for C432 ISCAS'85 benchmark circuit, POCOFAN-POFRAME determined one of the maximum optimal values of  $N$  and  $F$  to be 16 and 9, respectively, which

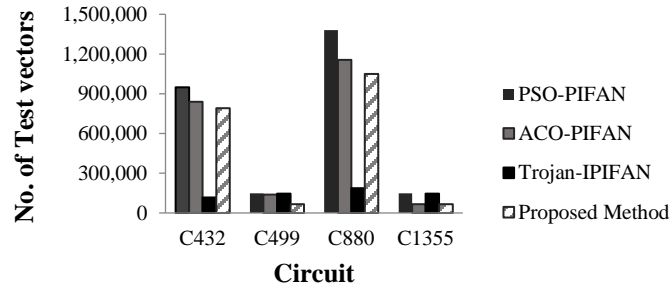
resulted in 13 partitions and 7,88,736 vectors at cone and 1,796 in the frame. The PSO-PIFAN approach generated 21 partitions and 9,49,000 test vectors with  $N$  and  $F$  equal to 18 and 11. The optimal solutions in the PSO-PIFAN algorithm are based on number of particles. A larger population of PSO particles could result in convergence to the optimal solution in less iteration and also the runtime per iteration would increase accordingly.

In existing method PSO-PIFAN and in ACO-IPIFAN, the reduction of test vectors is based on fitness functions [Venayagamoorthy (2007); Voyiatzis (2010)]. The two optimal values  $(N_{CO}, N_{FR}), (F_{CO}, F_{FR})$  of POCOFAN and POFRAME and the total number of partitions  $P$  is illustrated in Tab. 2. Hence the optimal value of  $(N, F)$  in the frame is preferred to be less than POCOFAN segment when the number of nets in the frame is greater than the cone.

**Table 2:** Comparison of proposed partitioning with other techniques

Benchmark Circuit	Method	N	F	P	No. of Test vectors	Increase in CPD (%)
C432	PSO-IPIFAN	18	11	21	9,49,000	41.18
	ACO-IPIFAN	20	7	10	8,40,000	5.66
	Proposed Approach	(16,8)	(8,2)	21	7,90,052	6.45
C499	PSO-IPIFAN	11	16	8	1,47,456	9.09
	ACO-IPIFAN	14	5	8	1,38,000	2.87
	Proposed Approach	(6,12)	(5,4)	12	66,564	2.33
C880	PSO-IPIFAN	11	16	20	13,80,000	20.83
	ACO-IPIFAN	6	18	27	11,56,000	2.70
	Proposed Approach	(18,8)	(8,3)	11	10,50,896	17.50
C1355	PSO-IPIFAN	10	16	8	1,47,456	4.17
	ACO-IPIFAN	12	5	16	65,500	2.07
	Proposed Approach	(14,9)	(5,4)	12	66,816	2.25
C2670	PSO-IPIFAN	NA	NA	NA	NA	NA
	ACO-IPIFAN	6	18	25	1,56,000	2.70
	Proposed Approach	(9,8)	(11,7)	25	1,53,246	2.19
C5315	PSO-IPIFAN	NA	NA	NA	NA	NA
	ACO-IPIFAN	5	14	22	31,888	5.71
	Proposed Method	(9,5)	(15,3)	20	40,072	4.67
C7552	PSO-IPIFAN	NA	NA	NA	NA	NA
	ACO-IPIFAN	5	12	16	65,540	2.90
	Proposed Approach	(8,7)	(15,2)	16	68,228	2.24

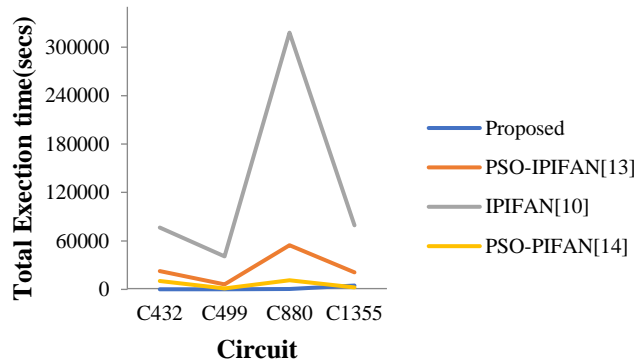




**Figure 11:** Comparison of test vectors with existing techniques

Fig. 11. shows variations in test vectors for ISCAS’85 benchmark circuits and compared with existing method. The optimal partitioning with greater transition probability using I-PIFAN for improving Trojan detection comparatively achieves less number of test vectors in C432 and C880 circuit but it occupies large area due to many partitions [Nejadmoghadam, Mahani and Kavian (2014)]. The execution time to predict a test vector of both cone and frame of ISCAS’85 benchmark circuits is shown in Fig. 12.

It is worth pointing out that different combinations of  $N$  and  $F$  can yield the same number of partitions and test vectors for a given circuit. Also, the same number of partitions can yield a different number of test vectors depending on the locations at which the partitions are made. However, there is no single global optimum solution possible for the problem. It varies according to the circuit arrangement, edge partitioning at the fanout branches from the fanout stem, maximum fanin, maximum fanout and the requirements of the circuit designer.



**Figure 12:** Time required to find test vector as a function of the problem dimension

The sample data obtained from our experiments for the benchmark circuit C880 are shown in the graphs of Figs. 13 and 14. It is seen that the variations of the critical path delay, partitions and number of test vectors are random, making solution searching an NP-Hard problem. The same is observed for the other benchmark circuits also. For

instance, when the number of partitions is 5, test vectors are comparatively less for the optimal solution (14, 8) when compare to (12, 8).

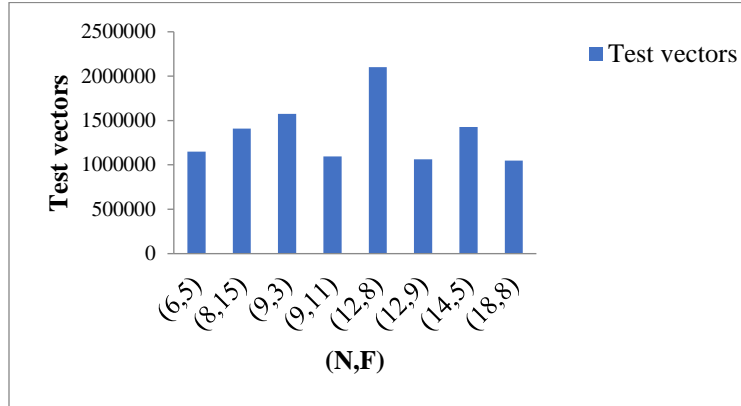


Figure 13: Optimal solution  $(N, F)$  vs. test vectors of C880

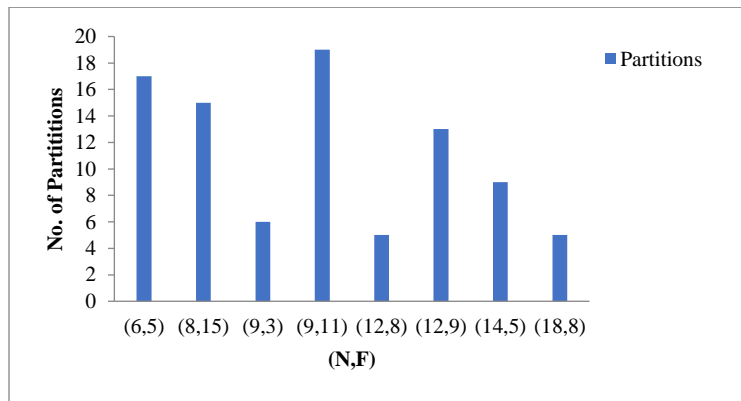


Figure 14: Optimal solution  $(N, F)$  vs. partitions of C880

## 6 Conclusion

In this paper, a modified method where a partitioned block could be used to reduce test patterns has been proposed. Two methods of partitioning are carried out based on two unique optimal solution of  $(N_{CO}, F_{CO})$  and  $(N_{FR}, F_{FR})$ . Each gate output is marked with primary input cone size  $N$  and fanout  $F$ . Each fanout stem has number of branches. Here in POCOFAN partitioning, partition takes place in the branches of gate fanout stem and the edges between the gates. It is recommended when node or gate has reconvergent fanouts within POCOFAN partitioning; only fanout partitioning is feasible. Similarly, when the circuit node has more reconvergent fanouts, edge partitioning provides better results with minimum number of vectors by avoiding fanout partitioning arise at the number of primary input node.

Typically in POFRAME partitioning, the selection of optimal solution  $(N_{FR}, F_{FR})$  is based on edges, number of gates and fanout branches. Due to more number of off-line

inputs only fewer nodes with fanout branches and partitioning is predictable. The ISCAS'85 benchmark circuits have been successfully partitioned, the test results of C499 shows 45% reduction in the test vectors and our experimental results are compared with other partitioning methods, our algorithm makes fewer test vectors.

Hardware manufacturers are increasingly subcontracting their IC fabrication work overseas due to their much lower cost structure. This stance a significant security risk for ICs used for critical military and business applications. Attackers can exploit this loss of control to substitute Trojan ICs for genuine ones or insert a Trojan circuit into the design or mask used for fabrication. Hardware Trojan are hidden in an IC and are located in low controllability and observability positions [Nejadmoghadam, Mahani and Kaviani (2014)]. Thus our method well suits to insert hardware Trojan where fanout branches separated from reconvergent fanout stem. The idea is to increase the transition probability of the potential points to the reachable output. It ensures that all nodes with low transition probability by reaching the shortest paths. The proposed algorithm reduces the test time of digital combinational circuits and will improve Trojan detection quality. Fault detection based test vector technique using pseudo exhaustive partitioning is considered as future work.

## References

- Chen, C. A.; Gupta, S. K.** (1998): Efficient BIST TPG design and test set compaction via input reduction. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 17, no. 8, pp. 692-705.
- Clerc, M.; Kennedy, J.** (2002): The particle swarm-explosion, stability and convergence in a multidimensional complex space. *IEEE Transaction on Evolutionary Computation*, vol. 6, no. 1, pp. 58-73.
- Goel, P.** (1980): Test generation costs analysis and projections. *Proceedings of ACM/IEEE 17th Design Automation Conference*, pp. 77-84.
- Gruning, T.; Mahlstedt, U.; Daehn, W.; Ozcan, C.** (1990): Accelerated test pattern generation by cone-oriented circuit partitioning. *Proceedings of European design automation EURO-DAC'90*, pp. 418-421.
- Jone, W. B.; Papachristou, C. A.** (1995): A coordinated circuit partitioning and test generation method for pseudo-exhaustive testing of VLSI circuits. *IEEE Transactions on Computer Aided Design Integrated Circuits and Systems*, vol. 14, pp. 374-384.
- Jose, D.; Kumar, P. N.; Hussain, A.; Shanker, P.** (2014): VLSI circuit partitioning using ant colony optimization to yield fault tolerant testable systems. *Arab Journal of Science and Engineering*, vol. 39, pp. 8709-8729.
- Kantipudi, K. R.; Agrawal, V. D.** (2006): On the size and generation of minimal n-detection tests. *Proceedings of the 19th International Conference on VLSI Design (VLSID'06)*.
- Kennedy, J.; Eberhart, R.** (1995); Particle swarm optimization. *Proceedings of IEEE International Conference on Neural Networks*, vol. 4, pp. 1942-1948.

**Khera, V. K.; Sharma, R. K.; Gupta, A. K.** (2017): A heuristic fault based optimization approach to reduce test vectors count in VLSI testing. *Journal of King Saud University-Computer and Information Sciences*.

**Kumar, K.; Bhaskar, S. U.; Chattopadhyay, P. S.; Mandal, P.** (2009): Circuit partitioning using particle swarm optimization for pseudo-exhaustive testing. *Proceedings of IEEE International Conference on Advances in Recent Technologies in Communication and Computing*, pp. 346-350.

**McCluskey, E. J.; Bozorgui-Nesbat, S.** (1981): Design for autonomous test. *IEEE Transaction on Computers*, vol. 30, no. 11, pp. 866-875.

**Navabi, Z.** (2011): *Digital system test and testable design using hdl models and architectures*. Springer, London.

**Nejadmoghadam, F.; Mahani, A.; Kaviani, Y. S.** (2014): A new testing method for hardware trojan detection. *Conference on Electronics, Telecommunications and Computers-CETC 2013*, vol. 17, pp. 713-719.

**Shaer, B.** (2004): Concurrent pseudo-exhaustive testing of combinational VLSI circuits. *Proceedings of the IEEE Computer Society Annual Symposium on VLSI Emerging Trends in VLSI Systems Design (ISVLSI'04)*, pp. 750-754.

**Shaer, B.; Dib, K.** (2002): An efficient partitioning algorithm of combinational CMOS circuits. *Proceedings of IEEE Computer Society Annual Symposium on VLSI*, pp. 142-146.

**Shaer, B.; Landis, D.; Al-Arian, S.** (2000): Partitioning algorithm to enhance pseudo exhaustive testing of digital VLSI circuits. *IEEE Transactions on Very Large Scale Integration Systems*, vol. 8, no. 6.

**Srinivasan, R.; Gupta, S. K.; Breuer, M. A.** (2000): Novel test pattern generators for pseudo-exhaustive testing. *IEEE Transaction on Computer Aided Design of Integrated and Systems*, vol. 49, pp. 1228-1240.

**Srinivasan, R.; Gupta, S. K.; Breuer, M. A.** (1993): An efficient partitioning strategy for pseudo-exhaustive testing. *Proceedings of ACM/IEEE 30th Design Automation Conference*, pp. 242-248.

**Venayagamoorthy, G. K.; Gudise, V. G.** (2004): Swarm intelligence for digital circuits' implementation on field programmable gate arrays platforms. *NASA/DOD Conference on Evolvable Hardware*, pp. 83-86.

**Venayagamoorthy, G. K.; Smith, S. C.; Singhal, G.** (2007): Particle swarm based optimal partitioning algorithm for combinational CMOS circuits. *Engineering Application of Artificial Intelligence*, vol. 20, pp. 177-184.

**Voyiatzis, I.; Gizopoulos, D.; Paschalis, A.** (2010): Recursive pseudo exhaustive two-pattern generation. *IEEE Transaction on Very Large Scale Integration Systems*, vol. 18, pp. 142-152.

**Xu, S.; Edirisuriya, E.** (2004): A new way of detecting reconvergent fanout branch pairs in logic circuits. *IEEE Proceedings of 13th Asian Test Symposium*.