Mehran University Research Journal of Engineering and Technology
Vol. 39, No. 2, 353 - 364, April 2020
p-ISSN: 0254-7821, e-ISSN: 2413-7219
DOI: 10.22581/muet1982.2002.11

X-Sand Filter: An X-Tolerant Response Compaction Technique for Faster-Than-At-Speed Testing

Salman Ahmad[1a], Kashif Iqbal[1b]

RECEIVED ON 11.01.2019, ACCEPTED ON 03.05.2019

ABSTRACT

Faster-than-at-speed testing provides an effective way of detecting small delay defects but at the cost of increased number of unknown logic values on longer paths of the circuit under test. For efficient testing, these unknown logic values need to be filtered out of the circuit under test output. In past, different compaction hardware schemes were presented to minimize these unknown logic values, all these schemes were effective in handling a limited number of unknown values arising due to design imperfections, processing problems manufacturing problems material problems etc. but no effective compaction scheme is available to handle large number of these logic values arising due to faster-than-at-speed testing. This paper presents "X-sand filter", a compaction technique, an extension of already presented idea of "X-tolerant signature analysis". Here, the idea of "X-tolerant signature analysis" with modifications has been applied and has attained a considerable improvement in the X-tolerance. X-sand filter is a hierarchical structure that handles gradual X-density reduction in an efficient manner. Simulation results obtained show that we can achieve up to 90 % reduction in the X-density if we use X-sand filter. Extensions to the work of X-sand filter can be carried out in future to enhance its capabilities and make its configuration more flexible in terms of layer designing.

Key Words: FAST, Small-Delay Defects, MISR, BIST (Built-in Self Test).

1. INTRODUCTION

A defect is any flaw/imperfection in integrated circuits (IC's) which causes them to deviate from their desired output. Defects in ICs may arise due to design imperfections, processing problems, manufacturing problems, material problems, aging factors or packaging issues. For testing purposes, defects are represented by fault models. Fault models identify and classify different defects. Different fault models like, delay fault models, transistors short and open fault models, stuck at fault models, and memory fault models are in use to detect and analyze faults e.g. defects in circuits in which one of the nodes of the circuit is stuck at some specific value which are modeled under stuck at faults model. For the detection of the fault, this model requires a test pattern, to activate and then propagate the fault information to the output.

"Delay defects" or "timing related defects" refer to any type of physical defects or an interaction of defects that adds enough signal-propagation delay in a device to produce an invalid response when the device operates at the targeted frequency [1]. With the advances in the field of technology, the dimensions (feature size) of the transistor and wires are with scaling down, as a result there is a decrease in the IC size. With this scaling down, the deep submicron effects are becoming prominent, resulting an increase in probability of timing related defects [2]. Delay defects are modeled under delay fault models.

Small-delay defects (SDDs) are one type of delay defects, which introduce a small amount of extra delay

[1] Department of Electrical Engineering, Wah Engineering College, University of Wah, Wah Cantt, Pakistan.
Email: [a]salmanahmad@wecuw.edu.pk (Corresponding Author), [b]kashifiqbal@wecuw.edu.pk

to the design [2] and are modeled as Small Delay Faults (SDFs) [3]. Small delay defects in IC's may also arise due to process variation, power supply noise, and cross talk. With technology shrinking, operating frequencies are increasing, as a result the allowable timing margin (time slack) for a path is decreasing and hence the margin for small delays defects to escape is also decreasing. This poses a serious threat for SDD's as they can cause an invalid output. SDD's can cause immediate failure if they occur on critical path, and may cause a quality risk if they occur on non-critical paths [3]. SDDs need to be detected to improve the overall quality of the product. The main problem with Small Delay Faults is that it is not always possible to detect them with at-speed tests; in fact, some SDFs show resistance against state-of-the-art timing-aware tests [4-6]. Since, they can only be propagated over paths with very large slacks, such faults are also called Hidden Delay Faults (HDFs) and pose a reliability issue, as they can cause an early life failure of the device [7].

Timing aware ATPGs (Automatic Test Pattern Generators) are designed to target faults on paths with minimum time-slack (longer paths), path 1 in Fig. 1 is the path with minimum time slack, as a consequence of this, SDD's on paths with maximum time slack (shorter paths) are not targeted, consequently considerable number of SDD's are undetected. To detect these SDD's on shorter paths delay testing with faster-than at speed testing (FAST) [8-10] is used to overclock the circuit under test (CUT). Faster-than speed testing is a technique to address the problem of time slack, where time slack is the difference between the times of clock frequency minus the time-taken by the signal to reach the output. Time slack provides tests escapes to delay defects, the greater the time slack, the greater the chances for the delay defects to escape the test. Increasing clock frequency as in case of faster-than-at-speed tests, time slack can be decreased, consequently decreasing test escapes.

In FAST, test frequency plays an important role. It involves testing circuit under test with a frequency above the nominal clock frequency for which the IC is designed, so that small delay defects on paths with longer time slack can be detected, which can be an advantage, but with a disadvantage of having a large number of unknown states at the output of the longer paths. These unknown states are called as X-states, X-values or Xs in short. For efficient testing, these unknown logic values need to be filtered out of the circuit under test output.

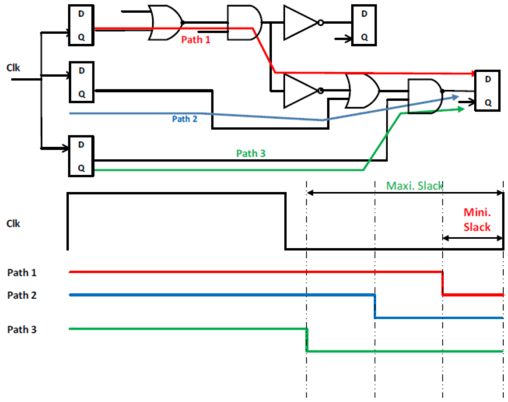

FIG 1: TIME SLACK AND ITS DIFFERENCES FOR SHORT PATHS AND LONGER PATHS

FAST increases fault detection by increasing the testing frequency, but at the cost of increase in switching activity of the circuit under test far higher than the normal operation, adding side effects like artificial increased gate delays, making exact timing analysis difficult [11-12]. Techniques for Automatic test pattern Generation for FAST can be used to help high switching activities by using single path sensitization criteria [13]. Furthermore, FAST uses multiple different frequencies to maximize the fault coverage and hence increase efficiency. Technique for determining a minimum number of frequencies can be used to increase the efficiency of the HDF with minimum number of test frequencies [14]. One design for testability technique specially tuned for faster-than-at-speed testing is discussed in [15]. The paper presented a special scan-chain configuration tailored to "X profile" along with an adaptive masking scheme followed by X tolerant compactor.

Response compaction in Built-in-Self Tests (BIST) application requires computation of reference signatures, but due to the presence of X-values during simulation it is not possible to compute the reference signature. Hence these X-values need to be filtered out/masked/blocked. Over a period of time different techniques have been developed to eliminate X's. These techniques usually employ one of the following approaches

(1) Fix the X-values to either of the logic 0 or 1. X-values need to be filtered before they enter the compaction hardware.
(2) Post processing of the response data, to find whether the tested IC is defective or not.
(3) A compaction hardware can be used that has the capability to tolerate X's and identity defective ICs.

Multiple Input Signature Registers (MISRs) and exclusive-OR trees are classical response compactors used in BIST applications, but there is one problem with these compactors, they are not X-tolerant i.e. in the presence of X-values at the output of the circuit under test, while performing the compaction for such an output, MISR will corrupt the output, and hence no output compaction is possible. This makes them useless to be used as output response compactors when the response has X-values. Different solutions to compact/mask/block x-values have been presented so far. Compaction hardwares have set a milestone in the field of X-tolerance and compaction [16-17].

The remainder of this paper is organized as follows. Section II presents the base of the proposed scheme, Section III presents the proposed system model and overview of the algorithm, Section IV presents the results obtained, showing the effectives of the system model in X-tolerance for FAST and Section V then concludes the paper.

2. BACKGROUND

X-compact for combinational circuits is presented in [16]. X-compact design can be used to design X-compactors for scan-out data with or without X-values both design techniques are discussed below. The proposed compactor is called as X-compact.
X-compact design can be used to design X-compactors for scan-out data with or without X-values. Both design approaches will be discussed with examples in this section of the document. X-compactors are implemented using exclusive-OR (XOR) gates. XOR gates. XOR gate has information loss-less property. XOR of an "X" with any other value is "X".

2.1 Compactor Design in the Absence of X-Values:

Fig.2 shows an example of compactor design based on X-compact, with better error detection in the absence of X-values. This design is capable of propagating all the errors present at its inputs (in the absence of X-value), with the condition that the errors originate from one, two, three or odd number of scan-out. Now consider the same situation as discussed above, if flip flop 10 of scan chain 5 and 6 have errors, this compactor will propagate both the errors, error from scan-out 5 will be propagated to either Out 1, Out 3 or Out 5, and error from scan-out 6 will be propagated to either Out 1, Out 4, or Out 5. As the hardware is flexible, hence propagation of error if it occurs at the same clock is propagated to one of the output.

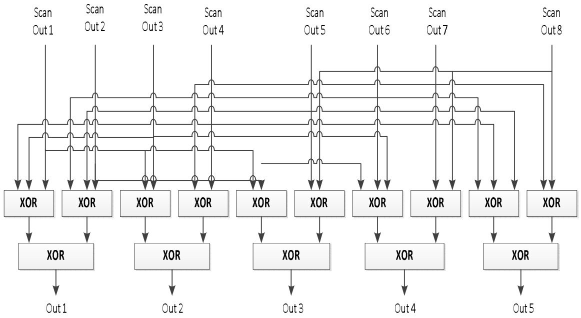

FIG. 2: X-COMPACTOR DESIGN WITHOUT X-VALUES

X-compact design can also be presented in the form of a matrix, called as X-compact matrix. X-compact matrix has number of rows equal to the number of un-compacted test responses at the input of the compactor and number of columns equal to the number of compacted output bits. For n- bits at the output of circuit under test, to be compacted to m- bits, X-compact matrix, can be represented by a matrix with n rows and m columns, for n > m. The matrix entry corresponding to i^{th} row and j^{th} column is 1, if and only if j^{th} bit of the compacted test response depends on i^{th} bit of the un-compacted test response. Otherwise, the entry is 0. A bit in the compacted response is obtained by calculating the XOR of those bits in the un-compacted test response that have 1s in the compacted response bit's column. X-compact matrix for Fig. 2 can be written as:

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \end{bmatrix} \qquad (1)$$

Compactor Design in the Presence of X-Values: [16] also presents the idea of an X-Compactor design in the presence of X-values. For the modified X-compact following theorem is considered

Theorem: For the same scan out cycle, error values from S_1 or fewer scan chains and X-values from S_2 or fewer scan chains, where $S_1 + S_2 \leq n$ (total number of scan chains), the error is guaranteed to be produced at the compactor output if and only if :

(1) None of X-compact matrix row contains all zeros.
(2) For any X-compact matrix, for any set k of S_1 and any set S_2 of the sub-matrix obtained by removing that rows in k and the X-compact columns having ones in that row are linearly independent.

X Compact matrix for Fig. 3 can be written as

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix} \qquad (2)$$

Fig. 3 presents an example of a modified X-compact design, having the same number of scan-outs i.e. 8 but the number of compactor outputs have increased from 5 to 6. The hardware is made more flexible to tolerate the X-values with error detection capability. The corresponding X-compact matrix is give below, as now the number of compactors outputs have increased from 5 to 6; hence the modified X-compact matrix has 6 columns instead of 5. Suppose for a particular scan-out cycle, scan chain 4 produces X whereas scan chain 1 & 8 produce errors. As XOR-ing of each value with X, has a result of X, therefore X produced by scan chain 4, will appear on outputs 2, 3 and 4. Error E1 cannot be observed at Out 2 and Out3. Similarly, E8 cannot be seen at Out 4 due to the presence of X. But as we can see errors appear at the outputs, E1 at Out 1 and E8 can be seen at Out 5 and Out 6. Therefore, this X-compact circuit is capable to detect all the errors even in the presence of X-values, which is indeed a big advantage of X-compact design.

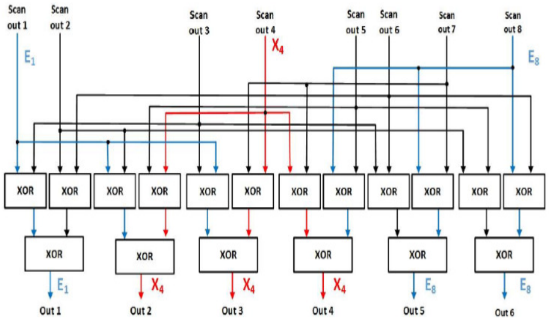

FIG. 3: X-COMPACTOR DESIGN WITH X-VALUES

The idea of the X-compactor (space compaction) presented in [16] is reutilized for a time compaction technique [17] and is named as X-Tolerant Signature Analyzers.

X-Tolerant Signature Analysis [17] presents a stochastic coding approach to fill X-compact matrix. With this approach X-compact matrix entries are filled with 1's and 0's randomly. X-compact matrix is a representation of X-compact, where number of rows represents inputs to the compactor and number of columns represents outputs of the compactor. With n inputs and m outputs of a compactor where n > m, then n represents number of rows and m number of columns in X-compact matrix. For the X-compact case, entries in matrix were made based on the dependencies of inputs on outputs i.e. a matrix entry corresponding to i^{th} row and j^{th} column is 1 if and only if j^{th} bit of the compacted test response depends on i^{th} bit of the un-compacted test response. Otherwise, the entry is 0. On the other hand, to fill this X-compact matrix for the time compactor, stochastic coding approach is adopted. That is entries in the matrix are filled at random, if we assign p as the probability of 1, then 1-p is the probability of 0 for any entry. The expected number of 1's in a row, the expected weight of the row, is then m × p [17].

A bit is interpreted as X-value, if in the expected compacted response the logic value of that bit is unknown, otherwise a non X-value. Error is assumed to be masked if the output compacted response has no non X-value bits with errors. The probability that an error in an output response bit gets masked when there are k X's in the other un-compacted test response bits, is given by the following expression:

$$\left[1 - \mathrm{p}\mathrm{x}(1-\mathrm{p})^{\mathrm{k}}\right]^{\mathrm{m}} \tag{3}$$

where p stands for the probability of 1s, k number of X's in the un-compacted response, and m compacted output response bits. Derivation of equation 3 is based on the fact that for any erroneous bit, if the corresponding entry in the row is zero, then error does not affect the corresponding bit in the compacted response. On the other hand, for any erroneous bit, if the corresponding entry is one, then error affects the corresponding compacted response bit if and only if none of the k rows producing X's have ones in that column otherwise the error is masked [17]. Equation 3 reaches its minimum when the following condition is satisfied

$$\mathrm{p} = \frac{1}{k+1} \tag{4}$$

Equation 4 gives the formula for calculating the probability of 1's for each entry in the X-compact matrix.

Based on the results obtained in [17], it is clear that X tolerant signature analyzers have limited capability of handling X's on the other hand looking into the scenario of faster-than-at speed testing, there is a tremendous increase in the number of Xs. X-compaction hardware schemes like [16-17] cannot handle such a large number of delay defects. Hence a better compaction and X-tolerant technique is needed that can handle large number of X-values.

3. SYSTEM MODEL

The idea presented in [17] is a time compactor, reutilizing the same idea with some modifications to form a space compactor is shown in Fig. 4 and this modified structure is called as stochastic X-compactor.

As can be seen in the Fig. 4, for each scan cycle *n* input bits are compacted into *m* output bits, where *n*>*m*, mathematically this compactor can be equated as:

$$\begin{bmatrix} I_1 & I_2 & \dots & I_n \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1m} \\ a_{21} & a_{22} & \dots & a_{2m} \\ . & & & \\ . & & & \\ a_{n1} & a_{n2} & \dots & a_{nm} \end{bmatrix} = \begin{bmatrix} O_1 & O_2 & \dots & O_m \end{bmatrix}$$

First matrix represents entries from I_1 to I_n that are inputs to the compactor (circuit under test outputs). The second matrix represents the weighted LFSR (Linear Feedback Shift Register) output; entries of this matrix are filled at random by weighted-random test patterns generated by weighted logic. Weight assignment to each entry is done on probabilistic approach, by assigning probability *p* to 1 and 1-*p* to 0, hence probability of 1 per row is p×m. Probabilities are calculated by the formula defined in equation 4. The third matrix is for the compacted output response. Depending on the output compaction required and the inputs to the compactor, dimensions of the matrices can be worked out. Weighted logic unit of X-compactor produces weighted random bits based on the X-density at its input.

This paper presents a space compaction technique that can handle an increased number of Xs arising due to faster-than-at-speed testing. The proposed design technique uses the X-compact in a hierarchical configuration to achieve a gradual reduction in compaction and X-density, one impact of increased X-tolerance could be a reduction in the fault coverage. However, fault coverage of this X-tolerant compactor is not considered for this presented work.

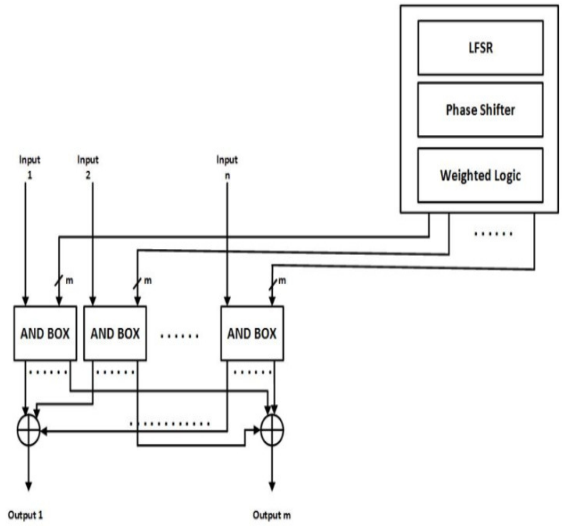

FIG 4: STOCHASTIC X-COMPACTOR

This configuration has multiple layers, each layer having multiple number of stochastic X compactors and is called as “X-sand filter”. “X-sand filter” filters out X’s from the output data in a layer after layer configuration like a sand filter does. Each unit in the configuration is a space compactor (stochastic X-compactor or X-compactor) with a limited capability of compaction and X-tolerance, for the overall configuration this capability adds up and the overall impact is a considerable increase in the compaction ratio and a decrease in the number of X’s at the output of the X-sand filter, which is a requirement of compaction hardware for FAST testing for SDD’s. First layer of the space compactor has the highest number of scan chains and X’s as input and hence highest number of stochastic X-compactors are required with gradual decrease in the number of scan chains and X’s. Last layer of the space compactor has the least number of stochastic X-compactors. First layer of the space compactor process the input scan chains, compacts the input data, reduces X-density and provides output that acts as input to the second layer, thereafter same process is repeated for all other layers, with a predefined compaction ratio for each layer.

The simulations of the research work are carried out with the help of two tools. Java as a programming language for the implementation of the algorithm and VHSIC_HDL (Very High Speed Integrated Circuits Hardware Description Language) or (Simply called as VHDL) for the implementation/simulations of the hardware components. In the first stage, the data is processed by the grouping algorithm to make scan chains with constant X-density per scan chain, once that is done, the data is then sent as input the X sand filter first layer for compaction and X values reduction.

The algorithm implemented accepts as input two different file formats

1. File format for the first layer (Time stamp files).

These files are in .txt extension, with test patterns information per row. They have last transition time stamp information in units of pico–seconds for each output signal.

2. File format for layers other than first layer (Time stamp files).

These files are in .txt extension, with scan chains information per row. First row of the file has output signals names, rest of the rows have output signals information

either as 0 or X. Due to the difference in input file format, algorithm uses two different classes for file parsing.

Algorithm implemented uses two types of data structures, Array and ArrayList. At start the algorithm scans through the number of rows and columns of the test file provided and based on this information makes a two dimensional array. This array stores test pattern information, in addition to that last column of the array is dedicated for storing the number of X count to be calculated by the algorithm. ArrayLists are used for storing group’s information..

As shown in Fig. 4, a Stochastic X Compactor has different components like LFSR, phase shifter, weighted logic unit and XOR modules. These components are individually implemented in VHDL and then combined to form one stochastic compactor.

The resulting Stochastic X compactor is a generic, having generic inputs and outputs. The next step in VHDL is to carry out the instantiation of different X compactors to form one layer. Total of 256 units of Stochastic X compactor can be instantiated per layer.

Manual configuration of the X-sand filter layers is considered, that is based on the final required compaction ratio and X-density, number of layers of X-sand filter are configured manually. An example of X-sand filter is given in Fig. 5.

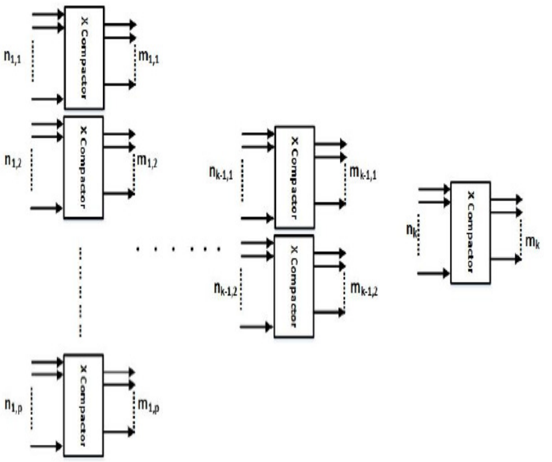

FIG. 5: X-SAND FILTER

Fig. 5 shows an example of X-sand filter with k layers, where k is a configurable parameter. Each layer has different number of X-compactors. First layer of the X-sand filter has p X-compactors with configurable inputs from $n_{1,1}$ to $n_{1,p}$ and configurable outputs $m_{1,1}$ to $m_{1,p}$. Similarly, kth layer has one X-compactor with configurable inputs n_k and configurable outputs m_k. As already mentioned basic building unit of X-sand filter is an X-compactor, integration of different units of X-compactor forms one layer. Each unit of X-compactor in itself is configurable, with generic inputs and outputs. In addition to that, number of X-compactor units per layer are also flexible. All these give flexibility to each layer, having flexible number of inputs, outputs and flexible number of units per layer. Multiple layers are used in one X-sand filter, again the number of layers are kept flexible as to suit different requirements.

As X-sand filter has multiple scan chains at input and multiple units of X-compactors per layer, hence weight calculation for each X-compactor is required, so that each X-compactor has a known X-density at its input. This can be done by grouping different scan chains, each with a known X-density. Grouping can be done in two ways

(1) Groups with a constant X-density. Where each unit of X-compactor in a layer has a constant X-density at its input.
(2) Groups with variable X-density. Where each unit of X-compactor in a layer has a variable X-density at its input.

Grouping with a constant X-density is considered for the presented work, to give an overall symmetrical structure to the X-sand filter and also to have an optimized solution. Different scan chains are grouped based on the number of X-density, such that each group has equal number of X-density. For this purpose, an algorithm is implemented, that computes X-density for each scan chain and then based on this information, group these scan chains. The nature of the problem is same like a bin packing problem where items of different volumes need to be packed into a container with a finite volume, with the goal to pack the items in as less bins as possible. This is an NP-complete problem, where to find an optimal solution is hard.

The algorithm implemented is “first fit algorithm”. First fit algorithm considers at start that all bins are empty and start sequentially with items to be placed. It tries to accommodate an item in the first available bin if possible, if this is not the case, it opens a new bin and tries to accommodate that item in that bin and so on. This is a straight forward greedy approach and provides a fast, but often non optimal solution to the problem. Groups are like bins and X-densities as items to be placed in the bins. Assume n scan chains at the output of circuit under test, each having variable number of Xs. Additionally, assume that overall X-density (total number of Xs in n scan chains) is *k*. Number of X’s in each scan chain are different, hence not a fixed number of scan chains can be accommodated in each group, resulting group size can vary. This proposed algorithm calculates X-density per scan chain, and then based on the predefined numeric values group scan chains, with constant X-density per scan chain.

Flow chart for the algorithm implemented is shown in Fig. 6.

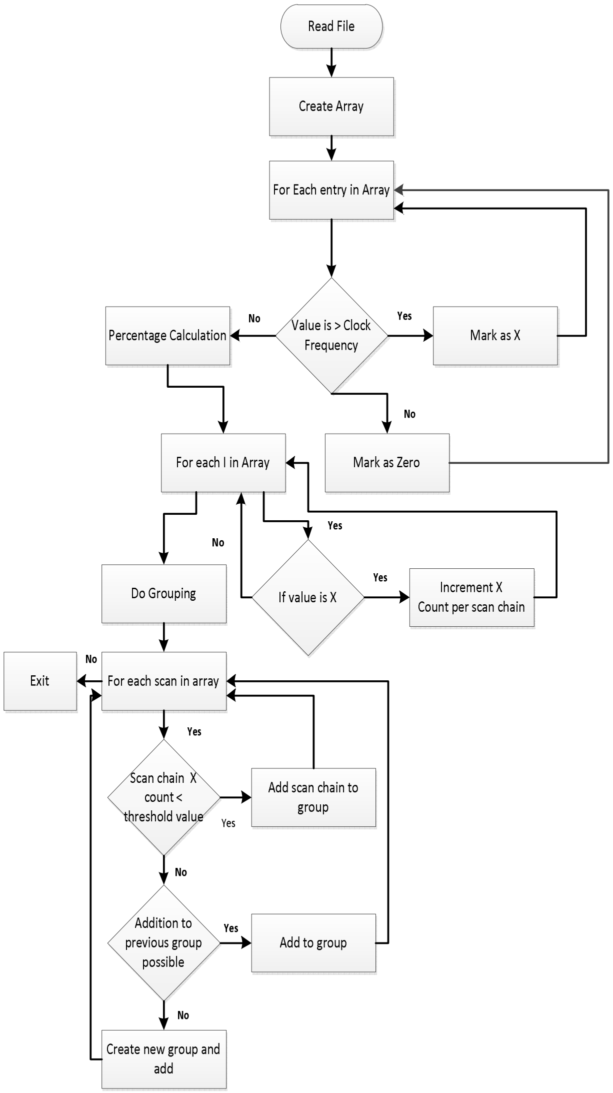

FIG 6: FLOW CHART

The grouped scan chains, with constant X-density per scan chain serves as input to the first layer of the X sand filter. The first layer of X Sand filter compacts the data as per requirement and provides an output with reduced number of X density. The same file is then given as an input the next layers of the X sand filters.

4. RESULTS

This section presents results for different time stamp files for test bench circuits b14_1, b21_1, b20_1. Initial input to the X-sand filter is the output of circuit-under-test, a time-stamp file with timing information in picoseconds. The first layer of the X-sand filter accepts this time-stamp file, the algorithm processes the information, makes groups and forwards it to the X-compactor for further processing. For a compactor with multiple layers, the same process is repeated multiple times for each layer separately. A threshold time is defined in the algorithm, all values above the threshold time are treated as X-states. Different time thresholds like 1000 ps, 2000 ps, 1073 ps have been selected for different test bench circuits and their results recorded. Each test bench file is tested for the following parameters:

(1) X-tolerance for different configurations.
(2) Compaction ratio obtained for different configuration.
(3) Effect of variation of X-tolerance with variation of number of units per layer.

Test bench file b14_1 is tested for time thresholds 1000 ps and 2000 ps. Initial input X-density for test bench circuit b14_1 ranges between 8.37% for threshold time of 1000 ps to 0.96% for threshold time of 2000 ps. This test bench file b14_1 is tested for 64 scan chains and 32 scan chains. As a standard procedure each layer of the compactor compacts the input bits into half, so for 64 scan chains the sequence of compaction is 64 to 32 bits for the first layer, 32 to 16 bits for the second layer and 16 to 8 bits for the third layer. Similarly, for the 32 scan chains the sequence is 32 to 16 bits for the first layer, 16 to 8 bits for the second layer and from 8 to 4 scan chains for the third layer. The test bench file b14_1 is processed for three layers. Fig.7 shows results obtained with test bench file b14_1 with 64 scan chains.

Fig. 7 shows that three layers of compactor are used for threshold time of 1000 ps and 2000 ps. At threshold time of 1000 ps, the initial X-density at the input of the first layer is 8.37 % of the total output response bits. First layer of the compactor compacts input bits into half, the X-density is reduced from 8.37 % to 2.49 % by the first layer giving 70.25% reduction. Second layer of the compactor again compacts the input bits into half from 32 scan chains to 16 scan chains, with X-density reduction from 2.49% to 1.27%, giving 48.99% reduction in the X's. Third layer of the compactor compacts input bits into half from 16 scan chains to 8 scan chains, with X-density reduction from 1.27% to 0.54%, giving 57.48 % reduction. Overall the compactor has reduced the X's from 8.3% to 0.5% with 93.54 % reduction in the X's.

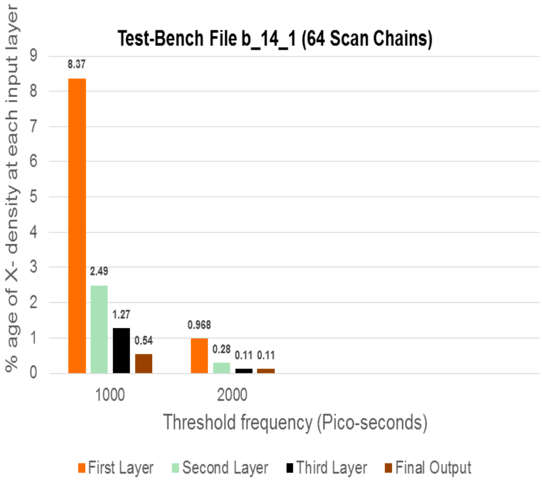

FIG. 7: TEST BENCH FILE B14_(64 SCAN CHAINS)

Similarly, for threshold time of 2000 ps initial input to the first layer is 0.968% of the total number of input bits. As can be seen that only two layers of compactor are used in this configuration to reduce the X-density from 0.968 % to 0.11%. First layer of the compactor compacts the input bits into half from 64 scan chains to 32 scan chains, with X-density reduction from 0.968 % to 0.28%, giving 71.07 % reduction. Second layer has input X-density of 0.28% and it reduces it to 0.11%, giving 60.71 % reduction in X-density. Overall the X-density reduction is 88.63% for two layers of compactor.

Fig. 8 shows results for test bench file b14_1 with 32 scan chains as initial input, test bench file is tested for threshold time of 1000 ps and 2000 ps. As can be seen for the threshold time of 1000 ps initial input X-density at the input of the compactor is 8.37 %. First layer of the compactor compacts the input bits into half with 32 scan chains into 16 scan chains; X-density is reduced from 8.37% to 3.059%, giving 63.45 % reduction. Second layer again compacts the input bits into half from 16 scan chains to 8 scan chains, with X-density reduction from 3.059% to 1.25%, giving 59.13% reduction in X-density. Similarly, the third layer compacts the input bits into half again from 8 scan chains to 4 scan chains. X-density is reduced from 1.25 % to 0.42 %, giving 64% reduction in X-density. Overall 94.98% reduction in X-density is achieved by three layer of compactor. Similar results for 2000 ps threshold times can be seen in Fig. 8.

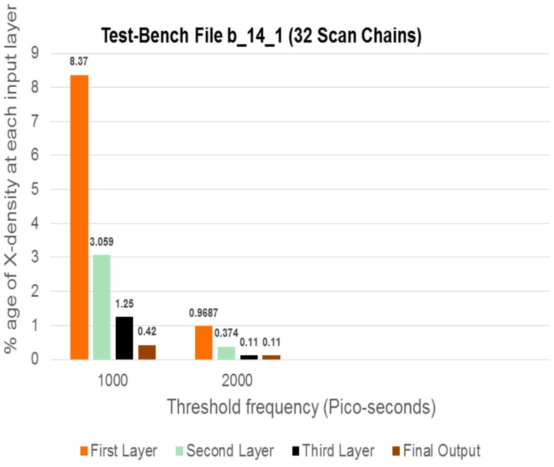

FIG. 8: TEST BENCH FILE B14_1(32 SCAN CHAINS)

Fig. 9 shows test bench file b21_1 with 64 scan chains as initial configuration. This test bench file is tested for threshold time of 1000 ps and 2500 ps with initial X-density ranging from 10.33 % for 1000 ps threshold time to 0.19 % for 2500 ps threshold time. For 1000 ps clock frequency compactor has three layers with X-reduction from 10.33% to 3.93 % by the first layer, giving 61.95% reduction. Second layer reduces X-density from 3.93% to 1.69%, giving 57% X reduction. Third layer reduces the X-density from 1.69 % to 0.68 %, giving 59.76% reduction in X-density. Overall X-density reduction for the tested bench mark circuit is 93.41%. Similarly, for threshold time 2500ps initial input X-density is 0.19% and is reduced in one layer to 0.1% with 47.37% reduction in X-density.

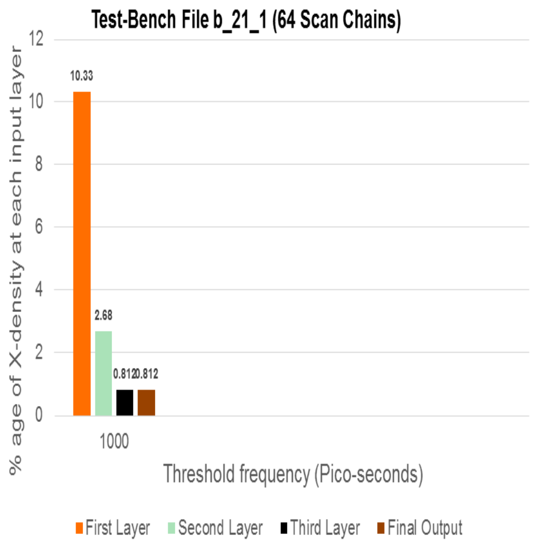

FIG. 9: TEST BENCH FILE B21_1(64 SCAN CHAINS)

Fig. 10 shows results recorded for the test bench file b21_1 for two different threshold times 1073 ps and 2500 ps with initial input X-density for these threshold times as 8.37% and 0.19% respectively.

As observed for the threshold time of 1073 ps initial input X-density at the input of the compactor is 8.37 %. First layer of the compactor compacts the input bits into half with 32 scan chains into 16 scan chains; X-density is reduced from 8.37% to 2.49%, giving 70.25 % reduction in X-denisty. Second layer again compacts the input bits into half from 16 scan chains to 8 scan chains, with X-density reduction from 2.49% to 1.27%, giving 49% reduction in X-density. Similarly the third layer compacts the input bits into half again from 8 scan chains to 4 scan chains. X-density is reduced from 1.27 % to 0.54%, giving 57.48% reduction in X-density. Overall 93.955% reduction in X-density is achieved by three layer of compactor. Similar results for 2500 ps threshold times can be seen in Fig. 10.

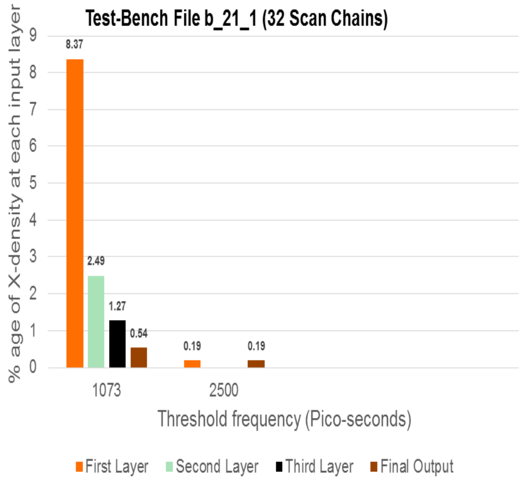

FIG. 10: TEST BENCH FILE B21_1 (32 SCAN CHAINS)

Fig. 11 shows that three layers of compactor are used for threshold time of 1000 ps and 2000 ps. At threshold time of 1000 ps threshold time the initial X-density at the input of the first layer is 10.33 % of the total output response bits. First layer of the compactor compacts input bits input bits into half, the X-density is reduced from 10.33 % to 3.95 % by the first layer giving 61.76% reduction. Second layer of the compactor again compacts the input bits into half from 32 scan chains to 16 scan chains, with X-density reduction from 3.95% to 1.99%, giving 49.95% reduction in the X's. Third layer of the compactor compacts input bits into half from 16 scan chains to 8 scan chains, with X-density reduction from 1.99% to 0.63%, giving 68.34 % reduction. Overall the compactor has reduced the X's from 10.33% to 0.63% with 93.91 % reduction in the X's.

Similarly, for threshold time of 2000 ps initial input to the first layer is 0.803% of the total number of input bits. As can be seen that only two layers of compactor are used in this configuration to reduce the X-density from 0.803 % to 0.21%. First layer of the compactor compacts the input bits into half from 64 scan chains to 32 scan chains, with X-density reduction from 0.803% to 0.36%, giving 55.17 % reduction. Second layer has input X-density of 0.36% and it reduces it to 0.21%, giving 41.66 % reduction in X-density. Overall the X-density reduction is 75.1% for two layers of compactor.

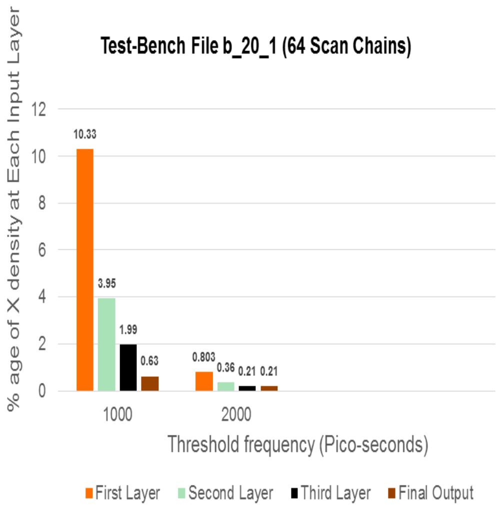

FIG. 11: TEST BENCH FILE B20_1(64 SCAN CHAINS

Fig. 12 shows results for test bench file b20_1 with 32 scan chains as initial input, test bench file is tested for threshold time of 1073 ps and 2000 ps. As can be seen for the threshold time of 1073 ps initial input X-density at the input of the compactor is 8.11%. First layer of the compactor compacts the input bits into half with 32 scan chains into 16 scan chains; X-density is reduced from 8.11% to 3.26%, giving 59.8 % reduction. Second layer again compacts the input bits into half from 16 scan chains to 8 scan chains, with X-density reduction from 3.26% to 1.89%, giving 42% reduction in X-density. Similarly third layer compacts the input bits into half again from 8 scan chains to 4 scan chains. X-density is reduced from 1.89 % to 0.39 %, giving 79.4% reduction in X-density. Overall 95.2% reduction in X-density is achieved by three layer of compactor. Similar results for 2000 ps threshold times can be seen in Fig. 11.

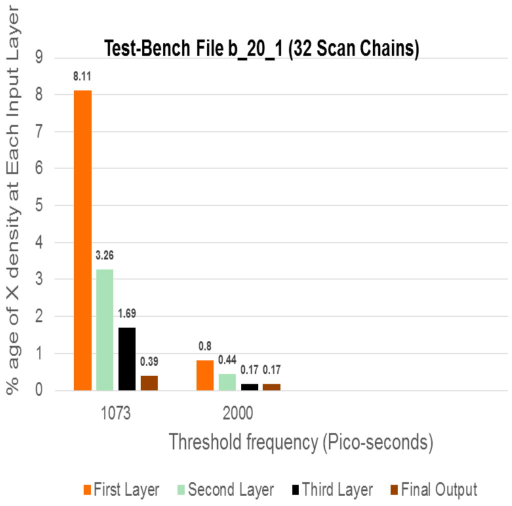

FIG. 12: TEST BENCH FILE B20_1(32 SCAN CHAINS)

5. CONCLUSION

Faster-than-at-Speed Test (FAST) is a good testing method to detect even very small hidden delay faults. However, FAST also produces a much larger amount of X-values than conventional at-speed tests. Hence, test response compaction gets increasingly difficult.

Simulation results for the X-sand filter obtained in this paper shows X-sand filter can achieve high reduction in X-values, making detection of very small hidden delay faults. Simulation results show that for the considered three layers configuration as standard, X-sand filter reduces the X-density from 85 % to 93 %, making it a considerably good option for compaction for faster-than-at-speed testing. With consideration of increasing number of units per layer, that is, by decreasing the number of layers from three to two it is again recorded that almost same reduction in X density from 77% to 91 % is attained, this on one end shows the flexibility of the system and on the other hand also saves the time for processing and hardware cost.

ACKNOWLEDGEMENT

We greatly acknowledge Prof. Dr. Sybille Helebrand, Head Department of Computer Engineering, University of Paderborn, Germany for her support, guidance and cooperation during our research work.

REFERENCES

[1] Mattiuzzo, R., Appello, D., and Allsup, C., “Small delay defects” EDN (Electrical Design News), Vol. 54, no. 13, p.28, 2009.

[2] Tehranipoor, M., Peng, K. and Chakrabarty, K., Test and Diagnosis for Small-Delay Defects, Springer, Chapter 3, pp. 21-36, New York, 2012.

[3] Yilmaz, M., Chakrabarty, K., and Tehranipoor, M., “Test-pattern grading and pattern selection for small-delay defects,” VLSI Test Symposium (VTS), San Diego, CA, USA, pp. 233–239, May 2008.

[4] Eggersgluß, S. and Drechsler, R., “As-robust-as-possible test generation in the presence of small delay defects using pseudo-boolean optimization,” *Design, Automation & Test in Europe Conference & Exhibition (DATE), Grenoble, France,* pp. 1–6, March 2011.

[5] Sauer, M., Kim, Y. M., Seomun, J., Kim, H.-O., Do, K.-T., Choi, J., Y., Kim, K., S., Mitra, S., and Becker, B., “Early-life-failure detection using SAT-based ATPG,” *Test Conference (ITC), Anaheim, CA, USA,* pp. 1–10, September 2013.

[6] Zolotov, V., Xiong, J., Fatemi, H. and Visweswariah, C., “Statistical path selection for at-speed test,” IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 29, no. 5, pp. 749–759, May 2010.

[7] Kim, Y. M., Kameda, Y., Kim, H., Mizuno, M., and Mitra, S., “Lowcost gate-oxide early-life failure detection in robust systems,” EEE Symposium on VLSI Circuits (VLSIC), Honolulu, HI USA pp. 125–126, June 2010.

[8] Amodeo, M., and Cory, B., “Defining faster-than-at-speed delay tests,” Cadence Nanometer Test Quarterly e-Newsletter, vol. 2, no. 2, May 2005.

[9] Yoneda, T., Hori, K., Inoue, M., and Fujiwara, H., “Faster-than-at-speed test for increased test quality and in-field reliability,” IEEE International Test Conference (ITC), pp. 1–9, September 2011.

[10] Yan, H., and Singh, A. D., “Experiments in detecting delay faults using multiple higher frequency clocks and results from neighboring die,” Proceeding of International Test Conference, Chalotte, NC, USA, Vol. 1, pp. 105-111, October 2003.

[11] Ahmed, N., Tehranipoor, M., and Jayaram, V., “A novel framework for faster-than-at-speed delay test considering ir-drop effects,” Proceedings of IEEE/ACM international conference on Computer-aided design (IC-CAD), San Jose, CA, USA pp. 198–203, November 2006.

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

[13] Fu, X., Li, H., and Li, X., “Testable path selection and grouping for faster than at-speed testing,” IEEE Transaction on VLSI Systems, vol. 20, no. 2, pp. 236–247, 2012.

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

[15] Matthias, K., & Sybille, H., “Design-for-FAST: Supporting X-tolerant compaction during Faster-than-at-Speed Test”. IEEE 20th International Symposium on Design and Diagnostics of Electronic Circuits and Systems (DDECS), pp. 35-41, Dresden, Germany, May 2017.

[16] Mitra, S., and Kim, K. S., “X-compact: an efficient response compaction technique,” IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 23, no. 3, pp. 421–432, 2004.

[17] Mitra, S., Lumetta, S. S., and Mitzenmacher, M., “X-tolerant signature analysis,” Proceeding of International Conference on Test (ITC) Charlotte, NC, USA, pp. 432–441, October 2004.