New scan compression approach to reduce the test data volume

Pralhadrao V. Shantagiri1 | Rohit Kapur2 | Chandrasekar Shastry2

1Department of Computer Science, Jain University, Bangalore, India
2Department of Computer Science, Research Guide, Jain University, Bangalore, India

Abstract
The test data volume (TDV) increases with increased target compression in scan compression and adds to the test cost. Increased TDV is the result of a dependency across scan flip-flops (SFFs) that resulted from compression architecture, which is absent in scan mode. The SFFs have uncompressible values logic-0 and logic-1 in many or most of the patterns contribute to the TDV. In the proposed new scan compression (NSC) architecture, SFFs are analysed from Automatic Test Pattern Generation (ATPG) patterns generated in a scan mode. The identification of SFFs to be moved out of the compression architecture is carried out based on the NSC. The method includes a ranking of SFFs based on the specified values present in the test patterns. The SFFs having higher specified values are moved out of the compression architecture and placed in the outside scan chain. The NSC is the combination of scan compression and scan mode. This method decides the percentage (%) of SFFs to be moved out of compression architecture and is less than 0.5% of the total SFFs present in the design to achieve a better result. The NSC reduces dependencies across the SFFs present in the test compression architecture. It reduces the TDV and test application time. The results show a significant reduction in the TDV up to 78.14% for the same test coverage.

1 | INTRODUCTION

Increased adoption of embedded cores and the complexity of an integrated circuit (IC) is surging as technology processes scale up. The increasing size and complexity of an IC contributes to the increase in the test application time (TAT) and test data volume (TDV). The TAT and TDV impact the test cost of an IC. The scan-based manufacturing test is a cost-effective method to test the circuit under test (CUT) for structural faults. The commercial scan compression schemes are exploiting many compressible bits present in the test patterns set for the detection requirements of the targeted faults. In the scan compression technique, the scan chain is partitioned into multiple internal scan chains and connected between compressor and decompressor (codec). The scan compression technique supplies the test data into multiple internal scan chains from a smaller number of scan-data-input ports and observing the test data from many internal scan chains at a few scan outputs. The ratio of the number of internal scan chains to the external scan-input ports decides the targeted compression. In a traditional full scan design, all SFFs are controllable and observable. The desired values are shifted into the scan chain serially. The full scan design in scan mode has no dependency introduced by the scan and achieves high test coverage.

The structural dependencies that exist in the scan and scan compression architecture cannot be changed. But scan compression technique introduces dependencies as many internal scan chains are fed from a few data bits of decompressor when target compression is increased. Increased scan compression increases the expansion ratio in the decompressor to internal scan chains. This increased compression ratio reduces controllability and observability in the scan chains [1], increasing patterns required to detect the faults in the CUT. Increased targeted compression ratio increases free variable dependencies and causes loss of test coverage, pattern inflation, and TAT. This phenomenon is called pattern inflation. Scan compression fails to detect all the targeted faults even though many test patterns are applied. The reason is the loss of controllability, and observability. The test coverage is very important output parameter of the scan compression. The increased test coverage helps to improve the
yield ramp and quality of the manufactured IC and vice-versa. The proposed method’s objective is to address this issue without compromising the test coverage, scan input, and output ports budget.

The detection of a fault in the scan compression is carried out by specifying a value in each test pattern. The uncompres-sible values in the test pattern are decided by the ATPG tool based on the fault picked from the fault list to detect. The uncompres-sible value of CUT is a very small number and hence a specified value to SFFs also small in number. The majority of the bits of the test pattern are compressible bits. The dependencies introduced by the scan compression architecture do not affect pattern inflation when a target compression ratio is a small number. It starts showing effect when compression ratio is increased. The result leads to pattern inflation. An initial outcome of such a collision in uncompres-sible values is solved by reducing the ATPG compaction such that the tests can be applied in the scan compression architecture. As long as the increased pattern count is less than the increase in targeted compression, there are gains to be achieved by increasing the number of internal scan chains that are connected to the scan terminals through the codec. If the target compression is increased to a point where it interferes with the compressibility of an uncompacted test pattern then the result is a coverage loss. At this point, the target compression has to be limited as test coverage is usually not negotiable.

Various scan compression schemes have been researched in the past decades. These are categorised based on the supply of free variables into the internal scan chains, compressor, and decompressor architecture to detect the structural defects in the CUT. These can be classified as: Code-based, Linear-decomposition based, and Broadcast scan-based architectures.

The different scan compression architectures are briefly explained in the following sections.

### 1.1 | Code-based scan compression architecture

The code-based scan compression architecture divides the test data into symbolic codes. Each symbol represents a certain group of bits to form the compressed test pattern. In [2], bit reversion-based scan compression architecture is proposed. This method flips some of the specified bits of the test pattern. The run time of code-based architecture is more as it has to process the pattern set and flip the bits in many patterns. The run-length based fixed code length scan compression is proposed in [3]. It does the compression by picking two or more same length compatible patterns to encode. Fixed-to-variable based selective Huffman coding scan compression architecture is proposed in [4] to reduce the TDV with an area overhead in the range of 2.9% to 17.1%. The various code-based schemes are proposed in [5–8]. The code-based test compression methods are usually inefficient in TAT reduction [25] and require complex control logic [9].

A number of scan compression techniques have been researched in the past two decades. Scan compression involves the manipulation of scan input and output data. Some of the studies focus on only one aspect of the problem. For example, code-based scan compression methods that encode the data stream solve only the scan input part of the problem. As a result, such methods have not been adopted in the industry.

### 1.2 | Linear decompressor-based scan compression architecture

In the linear decompressor-based scan compression architecture, the decompressor uses combinational logic or sequential logic or both sequential and combinational logic. This decompressor circuit is represented in linear equations and supplies the test data into the internal scan chains. The computation can be done faster to generate test patterns by the ATPG tool. The demerit of these architectures is the need for modifications in the pattern generator. The linear decompressor sharing free variables across multiple cores to reduce the TDV and TAT is shown in [10]. It suffers when non-identical cores are present in the CUT. In [11] linear decompressor-based scheme which combines compression constraint with the ATPG engine to achieve the reduced TDV is proposed. In [12], feed-forward-based compression architecture is proposed to encode the test cube with dynamic compaction. The various linear decompressor-based schemes are researched in [13–16].

### 1.3 | Broadcast-based scan compression architecture

A methodology with a combinational decompressor is presented in [17] with a few inputs. The same test data is broadcasted into multiple circuits in fan-out mode. The disadvantage of this method is that it is not practically usable as each CUT may have chains with varying lengths and also may not be identical ones. Illinois scan [5, 18] broadcasts free variable into many internal scan chains through the single scan input port. In this case, free variable dependencies are introduced across SFFs in the same row of different chains. In CircularScan [19], scan chains are configured in the circular form. The next pattern is generated from the previous pattern’s capture response and succeeded in overcoming the limitation of the test channel bandwidth. VirtualScan [5] also supplies free variables into the CUT through a broadcast network to reduce the correlation in the TDV. The scan tree or scan forest is another broadcast scan scheme [5] with multilevel of scan partitions. The effectiveness of these schemes is improved by supporting different configurations (static and dynamic) to allow chains to drive from different configurations. FCSCAN [20] is a broadcast scan compression scheme which encodes minority specified uncompres-sible value zero, or one in the scan slices for compression. The LFSR (Linear Feedback Shift Register) deterministic pattern-based broadcast compression technique has been proposed in [21].

Dynamic Scan [22] divides the scan chain into multiple scan chain groups. The [22] proposed mechanism is to bypass
unused scan groups during reconfiguration of the scan architecture. This method helps to decrease the TDV by skipping certain SFFs in each configuration. This makes the length of the scan chain dynamic and needs more ATPG effort to generate patterns. In [23], streaming broadcasting scan compression architecture is proposed to reduce the TDV. This architecture handles dependency across SFFs using different load modes and shifting test data in both the directions into the internal scan chains. As load modes are limited, some detectable faults in scan mode cannot be detected in this architecture.

In [24], broadcasting-based scan compression architecture is proposed based on the capture-X analysis. It showed TDV could be improved by excluding SFFs capturing Xs from the compression architecture and placing them in an external scan chain. In [25], profiling of SFFs and excluding them from the compression architecture is studied and the study was limited to the last 95% of the patterns generated and created a higher number of the external scan chains to improve the TDV. In [26], a reconfigurable broadcasting technique using Internal Joint Task Action Group (IJTAG) to test identical cores is proposed. In [27], scan compression architecture with a reconfigurable scan chain group is proposed with the multiple test session. But, this architecture suffers from the area overhead and routing congestion. The various broadcast-based scan compression architectures have been proposed in [5, 15, 17–22, 28–36].

Furthermore, a literature review on uncompressible bits density is carried out. In [20, 37–40], the average uncompressible bits density in the CUT varies from 0.1% to 15%. In [40], compression and uncompressible bits density limits are reviewed.

In a broadcast-based scan compression technique, the scan chain is divided into multiple internal scan chains and connects them between the decompressor and the compressor, which is shown in Figure 1. Figure 1 shows how a single scan chain is partitioned into six internal scan chains in the compression architecture to reduce the number of shift cycles. The scan compression scheme uses an external scan-input port to broadcast scan load test patterns into internal chains help to reduce the TAT. These broadcasters can be combinational or sequential logic.

The scan compression architecture proposed by the authors is called as new scan compression (NSC). The ‘NSC’ method is validated using the DFTMax Ultra by Synopsys Inc. Though the results are obtained by using this technology, the analysis can be applied to other compression architectures also.

The authors proposed the ‘NSC’ method by analysing the free variable dependency across the SFFs in the compression architecture. The main contributor to the TDV is the free variable dependency introduced by the test compression architecture. The results shown by the authors for ‘NSC’ architecture are encouraging and able to achieve patterns count reduction up to 78.14%. The ‘NSC’ method exploited the advantages of both scan compression and scan having SFFs with more free variable dependency within the compression architecture or outside the compression architecture in the external chain. While doing dependency analysis, uncompressible values logic-one and logic-zero present in the scan load test pattern are considered. The ‘NSC’ method can be implemented in any compression scheme that exists today. The decision to move out SFFs outside test compression architecture (external chain) is based on the dependency ranking of the SFFs shown in Section 2. As shown in Figure 2, NSC architecture having 16 scan-in and scan-out terminals are considered. In this, the external chain is connected to the scan terminals Si_16 and So_16, which is shown in Figure 2. This external chain helps to reduce dependency and patterns count and more details are provided in Section 2. The compression architecture is connected to the scan-in from Si_1 to Si_15 and scan-out to So_1 to So_15, as shown in Figure 2. These Si ports supply test data to the compression scheme and the external chain. The So ports are used to observe the result.

The ‘NSC’ architecture has been described in Section 2. The algorithm to select SFFs to be moved out of the compression architecture is described in Section 2. The results are presented in Section 3 and the conclusion of the work in Section 4.

\[ F I G U R E \ 1 \] Partitioning of scan chain into multiple internal scan chains connected between codec in the scan compression architecture
decrease the patterns count to achieve desired test coverage without the application of top-off patterns, which are expensive. If the user wants to use top-off patterns, it can still be used with the ‘NSC’ architecture.

In the ‘NSC’, the free variable dependencies are exploited that are introduced by the scan compression architecture. In the scan mode, all the SFFs are controllable to load the desired values into them and observable by shifting out the captured value at scan-out during structural testing to detect the faults. There is no free variable dependency introduced by scan mode. Free variable dependencies exist in the scan compression architecture as a few data bits or scan terminals supply test data to many internal scan chains. This free variable dependency adds to the pattern inflation and increased TDV. The structural dependency exists in the scan, and scan compression architecture cannot be changed.

In the ‘NSC’ architecture, a scan mode pattern set is analysed to understand the SFFs contributing to the pattern inflation. The SFFs which need to have uncompressible value in most of the test patterns or many patterns are the ones contributing to the pattern inflation. Also, these SFFs are creating many free variable dependencies because of fan-out cone connection introduced by the compression architecture, as shown in Figure 3. To detect each fault, certain SFFs need to have uncompressible value (logic-0, logic-1). These SFFs are usually present in the different internal scan chains within the codec. So, as compression architecture feeds test data from ATE channels to many internal scan chains, dependencies are created. To address this issue in the compression architecture, it needs to load multiple test patterns to detect the faults list's targeted faults. In the proposed ‘NSC’ architecture, such SFFs are moved out from the compression scheme and placed into the external chain. These moved out SFFs are less than 0.5% of the total SFFs present in the CUT. The detailed illustration of SFFs being moved out from the compression architecture is depicted in Section 2.4. The calculation of the percentage of SFFs to be moved out of the compression architecture is shown in Section 2.4.

In ‘NSC’, patterns considered for analysis are generated for scan mode in assumed scan mode or ATPG generated. These patterns are made up of binary logic-1, logic-0, and X. The X is the unknown value, which is compressible. The logic-
0 and logic-1 are uncompressible bits. Each pattern in the pattern set has a few uncompressible values in it. These values are required to detect the targeted faults from the fault list. The ‘NSC’ architecture is made based on the pattern analysis for dependencies across the SFFs in the compression architecture. The details of free variable dependencies resulted from the compression architecture are depicted in Section 2.1. The SFFs ranking and selection algorithm is depicted in Sections 2.2 and 2.3. The detailed procedure to analyse and to select few SFFs to be moved out of the compression architecture is explained in Section 2.4. The complete flow of execution of the ‘NSC’ architecture along with existing compression architecture to generate patterns is depicted in Section 2.5.

### 2.1 Free variable dependencies across scan flip-flops in compression scheme

The ‘NSC’ architecture is validated using the DFTMax-Ultra and built on it. This reduces internal scan chains length and needs additional registers to feed multiple internal scan chains from a few ATE channels. The decompressor register comprises control register bits to supply control value to the decompressor and compressor. The control value includes load and unload direction bit, masking bits etc. It also provides data registers which supply the test data into the multiple internal scan chains based on the grouping specific to the load modes [23]. The scan compression scheme introduces free variable dependency which cannot be seen in a scan mode. The free variable dependencies occur as few data bits supply test data to many internal scan chains connected between codec. Though decompressor supports multiple load modes it is not sufficient to handle all the free variable dependencies. Dependencies are handled with shifting input test data in both the direction [23]. Per shift dynamic mode change is available for defined modes based on grey code as it has less freedom to change from one mode to another. This architecture has multiplexers in the decompressor to help share the scan data inputs.

Figure 3 shows the fan-out cone dependencies introduced by the scan compression architecture. This dependency increases patterns inflation as the ATPG engine tries to generate test patterns to detect the faults in each combinational logic cone. The data bit of the decompressor register D0 and D1 are fan-out to many internal scan chains and each scan cell has fan-out to combinational logic cone. The larger the combinational cone, scan cells fan-out to these cones need uncompressible values in most of the scan load test patterns. This leads to pattern inflation, TAT, and increased test cost. The fan-out cone dependency has been shown in Figure 3(b) and captured in Table 1. The scan flip-flop 1 and 2 are connected to five combinational logic cones requiring uncompressible values in many load test patterns.

The scan cells presented in Figure 3(a) and (b) range from SC1 to SC6 and feed the test data to the combinational overlapping and non-overlapping fan-out logic. The dependency exists in blocks numbered from 7 to 15 as shown in Figure 3(a). The fault detection in the non-overlap block is easy. The scan cell SC4, SC5, and SC6 are required to have uncompressible values in many test patterns to detect the faults in the combinational block 13. To detect the faults present in blocks 5, 10, 11, 12, 13, 14, 15, and 11, the scan cell SC5 is required to have the uncompressible values in many test patterns because of dependency. Many faults in the cones are undetectable if SC1 to SC6 are fan-out from the same data bit and hence causes an increase in patterns count and decreases fault coverage.

The representative dependencies across the SFFs of this architecture are shown in Figure 4. Figure 4(a) shows the data bit feeding many internal scan chains. Here, each scan slice is shown in the same colour to show that those SFFs get the same test data. This creates dependencies across those SFFs. Figure 4(b) shows streaming test data into the internal scan chains. The dependencies are shown in diagonal fashion. These dependencies are introduced by the scan compression architecture causing loss of test coverage and patterns inflation. The coverage loss scenario has been depicted in Figure 4(c) and (d). In Figure 4(c), AND gate gets data from two SFFs connected to two different chains in the diagonal fashion. The DFTMax-Ultra, has diagonal dependency of free variables. Assume that both the chains are connected to the same data bit in the decompressor. These two SFFs get the same value and cause controllability loss. This results in non-detection of fault at AND gate and results in coverage loss. Also, compression scheme tries to detect the fault by generating more test patterns and leads to pattern inflation. If these two chains are driven by different data bits of the decompressor in another mode, then the fault on AND gate can be detected. This kind of situation does not appear in a scan mode because of the absence of the free variable dependency.

In the scan compression schemes, fault coverage is limited by the non-reachability of the circuit states when the scan data input ports are shared in the compression architecture as shown in Figure 4(e). The circuits driven by the scan chains are independent, hence the loss of controllability. The loss of coverage is seen in the compression scheme, as pointed out in Figure 4(e) and (d). Here scan elements SC31, SC22, and SC13 get the same test data value that can be either logic-0 or logic-1. Notice that gate G2 which is Exclusive-OR always produces zero as an output. The detection of s-a-0 fault at gate G2 output fails. The logic gate G1 always produces zero and cumbersome to detect s-a-0 fault in it. The dependency introduced by the scan compression is the main cause of loss of coverage.

Generally, top-off serial test patterns are used in scan mode to overcome loss of fault coverage and patterns inflation. The cost of the serial scan test data patterns is much higher and hence it is advisable to keep the top-off serial vectors count low. It means faults that cannot be detected in the scan compression mode still can be tested with the serial
scan mode because of the absence of free variable dependencies in a scan mode. So, though scan compression reduces the TDV and TAT, it cannot detect all the faults being detected by the serial scan mode. To overcome these issues, there is a need for a solution that detects such faults without compromising the test coverage. The ‘NSC’ architecture proposed here is the solution to overcome such a problem. The ‘NSC’ method identifies the SFFs that contribute to the patterns inflation and moves out such SFFs from the compression architecture and places them into the external scan chain in a compression mode. This external scan chain has no free variable dependencies. All the SFFs present in the external scan chain are controllable and observable. Hence ‘NSC’ method exploits the advantages of both scan mode and scan compression mode to reduce the TDV by reducing dependencies across the scan cells in the compression mode. The ‘NSC’ architecture detects additional faults which are not detectable in the scan compression mode and helps to overcome this test coverage loss, pattern inflation, and reduces the top-off patterns required to improve the yield.

### 2.2 Ranking and selection of scan flip-flops

The pattern inflation occurs in scan compression because of increased uncompressible values and free variables dependencies created by the scan compression across the SFFs in scan chains. This usually happens when targeted compression is increased. The SFFs need to have a certain value to detect the targeted fault from the faults list. These cells are required to have uncompressible value in a greater number of the test patterns to detect multiple faults. Such SFFs contribute to pattern inflation. This also increases the TAT and TDV. The increased targeted compression ratio also reflects in loss of the test coverage and increased TDV. The proposed ‘NSC’ solution overcomes this problem. At first, the SFFs contributing to the pattern inflation are found out. It is carried out by analysing sample scan load patterns or all ATPG generated scan patterns for uncompressible value in each scan cell. It can be accomplished either by generating patterns using Assumed-scan or generating ATPG patterns after doing scan synthesis.

The algorithm to select scan cells that have to be excluded from the compression architecture and to create an external scan chain is depicted as follows:

| Scan flip-flop | Number of fan-outs | Combinational logic cone number |
|---------------|--------------------|---------------------------------|
| SC1           | 5                  | 1, 2, 3, 4, 5                  |
| SC2           | 5                  | 2, 4, 6, 7, 9                  |
| SC3           | 4                  | 5, 6, 7, 8                    |
| SC4           | 3                  | 5, 8, 9                      |
| SC5           | 2                  | 8, 10                        |
| SC6           | 6                  | 3, 5, 7, 9, 11, 12            |
The above algorithm can be understood with an example. Table 2 shows the representative data having captured for the scan chain with 10 SFFs and 9 patterns in the pattern set. The first row shows SFFs number, and the first column represents pattern number. Row number 2 to 10 shows a representative test pattern set. Each pattern has uncompressible value logic-0 and logic-1. The ‘X’ is the unknown value present in the test pattern. The ranking of SFFs based on the probability of uncompressible value present in the pattern set is carried out. In Table 2, SC1 is the first scan cell and SC10 is the last scan cell present in the scan chain. The scan cell SC1 has uncompressible value logic-1 in 6 patterns and logic-0 in 2 patterns (refer row 11, ‘#Uncompressible’). Hence the occurrence probability of SC1 is 0.88. Similarly, the probability calculation for all the SFFs of the chain is calculated. Based on probability, scan cell occurrence in the percentage of test patterns count ‘P’ is calculated. The ‘P’ indicates the scan cell needs to have uncompressible value in the percentage of patterns in the pattern set. If P is closer to 100%, it needs to have uncompressible value in most of the patterns or all patterns based on the value. This indicates the dependency across scan cell and cells having higher P are the ones that contribute to patterns count and patterns inflation. Hence, if these SFFs are put outside the scan compression, it becomes easy to load the desired value without disturbing the value in dependent SFFs across the SFFs within the scan compression. This leads to the detection of the targeted faults with the reduced number of scan test patterns. The test coverage is not negotiable. Hence achieving the same test coverage with a smaller number of test patterns reduces the overall cost of an IC testing and TAT. Then, SFFs are arranged in decreasing order based on the probability ranking ‘R’ or percentage of occurrence P of the scan cell. First, ‘L’ SFFs are picked and put into the chain outside the scan compression. These SFFs will not be part of the compression architecture. The sum of the uncompressible values of each scan flip-flop is shown in Equation (1). The probability ranking calculation of SFFs is shown in Equation (2). The calculation of the occurrence of the scan cell in the percentage of test pattern count is shown in Equation (3). The calculation of the value of ‘L’ is shown in Equation (4) of Section 2.4.

\[
v = \sum_{c=1}^{N} Pat_{rc} \quad \text{Where} \begin{cases} 1 & \text{Pat}_{rc} = 1 \\ 1 & \text{Pat}_{rc} = 0 \\ 0 & \text{otherwise} \end{cases}
\]

\[
v = (\text{\#Uncompressible})_{c=1}^{Len}
\]
where $N$ is the total number of patterns present in the pattern set $P$, and $v$ holds the uncompressible value for each scan cell. The ranking of SFFs is carried out based on the below given equation:

$$ R = \frac{v}{N} \quad (2) $$

The ranking of each scan cell (FF) in the CUT is carried out based on the scan load test patterns, which is shown in Table 2. The ranking of scan cells in circuit C1 is shown in Figure 5. The rank of each scan cell is in the range of 0 to 1. Note 1 is the highest and 0 is the lowest.

The percentage of occurrence of scan cell in the pattern set is calculated as follows:

$$ P = R \times 100 \quad (3) $$

### 2.3 | Scan flip-flop ordering based on ranking

Once the ranking and $P$ occurrence of each scan cell is calculated, then those SFFs are ordered in a descending order based on the probability ranking value. The SFFs from Table 2 are ordered and is as follows:

- SC4, SC1, SC2, SC5, SC9, SC3, SC6, SC7, SC8, SC10

This order will be used while picking the SFFs to be moved out of the scan compression scheme and include them in an external chain.

### 2.4 | Percentage of scan flip-flops to be moved out of the compression architecture

The ‘NSC’ method needs to execute the decision to move out the number of SFFs from the scan compression architecture. This count decides the length of the external scan chain, which holds that moved out SFFs from the compression architecture to help reduce inter-dependencies across SFFs in the compression architecture. The length of the external scan chain should be less than or equal to the shift cycles required to shift-in each test pattern. Shift-in each test pattern into the compression architecture needs many shift cycles which is always equal to the length of the decompressor serial register plus length of the longest internal scan chain present within the codec. This has been shown in Figure 6.

The decompressor register’s length is ‘$Dl$’ and the longest internal scan chain as ‘$Sl$’. The value of ‘$Dl$’ and ‘$Sl$’ can be obtained from the protocol file generated by the scan insertion flow.

The length ‘$L$’ of the external scan chain is as follows:

$$ L = Dl + Sl \quad (4) $$

### Table 2

| Pattern number | SC1 | SC2 | SC3 | SC4 | SC5 | SC6 | SC7 | SC8 | SC9 | SC10 |
|----------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|------|
| 1              | 1   | 1   | 1   | 1   | 1   | 1   | 1   | 0   | 1   | X    |
| 2              | X   | 0   | X   | 1   | 0   | X   | 0   | 0   | 1   | X    |
| 3              | 0   | X   | 0   | 1   | 0   | 1   | X   | 0   | X   | X    |
| 4              | 1   | 1   | 0   | 0   | 0   | 1   | X   | X   | X   | X    |
| 5              | 1   | 0   | 1   | 1   | X   | 0   | 0   | X   | 0   | X    |
| 6              | 0   | X   | X   | 1   | X   | 1   | X   | 0   | X   | X    |
| 7              | 1   | 0   | 1   | 0   | 1   | X   | X   | 0   | X   | X    |
| 8              | 1   | 1   | X   | 0   | 1   | 0   | X   | X   | 0   | X    |
| 9              | 1   | 0   | 1   | 1   | 1   | 0   | 1   | 1   | 0   | X    |
| #Uncompressible | 8   | 7   | 6   | 9   | 7   | 6   | 5   | 4   | 7   | 0    |
| Ranking (R)    | 0.88 | 0.77 | 0.66 | 1.00 | 0.77 | 0.66 | 0.55 | 0.44 | 0.77 | 0    |
| Percentage of occurrence (P) | 88.88 | 77.77 | 66.66 | 100 | 77.77 | 66.66 | 55.55 | 44.44 | 77.77 | 0    |

**FIGURE 5** Ranking of scan cells in circuit C1
the targeted compression. For example, consider the scan compression architecture having 8 scan-ins and scan-outs with 800 internal scan chains connected between the decompressor and the compressor, considering a compression ratio of 100. The length of each scan chain is say 200. So, the total number of scan cells (200 × 800) present in the design is equal to 1, 60,000. Then the length of the external scan chain is equal to L = 200 + 25 = 225, where 25 is the decompressor serial register length. So, the percentage of SFFs being moved out is 225 × 100 / 160000 = 0.14%. If the same is considered for a compression ratio of 50, then the number of scan chains is equal to 400. The length of each chain is equal to 160000 / 400 = 400. The external scan chain length is = 400 + 20 = 420, where 20 is the length of the decompressor register. So, the percentage of SFFs being moved out is 420 × 100 / 160000 = 0.26%.

The percentage of SFFs being moved out the scan compression is calculated as follows:

PSc = (L × 100)/Tsc \hspace{1cm} (5)

where Tsc is the total number of SFFs present in the CUT. The Psc is the percentage of SFFs being moved out of the scan compression to reduce the TDV and TAT in the ‘NSC’ architecture.

2.5 | The flow of execution of new scan compression’ architecture

The complete flow of execution of ‘NSC’ architecture has been shown in Figure 7. It includes the flow of scan insertion and patterns generation for the scan compression architecture, creation of chain outside scan compression, and scan insertion and patterns generation for the ‘NSC’ architecture. Figure 7 shows three main steps numbered 1, 2, and 3. In the first step, scan insertion and patterns generation are carried out for the scan compression architecture based on the user’s input parameters as Design For Test (DFT) configuration. The DFT configuration for codec includes the number of scan-in ports, scan-out ports, number of internal scan chains to be created within the codec, etc. The outcome of this step is the creation of scan inserted design and test protocol. The ATPG patterns are generated to detect the manufacturing faults in the CUT. The recording of TDV and TC (test coverage) is carried out.

In Step 2, a detailed analysis of each scan cell based on the probability ranking procedure depicted in Sections 2.2, 2.3, and 2.4. It includes selecting the first L number of SFFs as per Equation (4) from the ordered list shown in Section 2.2. The percentage of SFFs to be selected is shown in Equation (5). Such selected SFFs are placed in the scan chain being created outside the scan compression architecture in compression mode. The ‘NSC’ architecture is shown in Figure 2. This is shown in Step 2 of Figure 7.

In the last step (Step 3), scanning of insertion and patterns generation is done for the ‘NSC’ method based on the input parameters provided by the user along with the external scan chain specification. The input parameters include the number of internal scan chains, input and output ports budget, external scan chain specification, the CUT, and relevant libraries required to do scan insertion. Generate scan inserted design and test protocol file. The ATPG engine is invoked to generate test patterns for the ‘NSC’ architecture. The recording of the TDV and TC is carried out.

Compare the test coverage and TDV measured in Steps 1 and 3. The comparison of the patterns count at the same test coverage in both Step 1 and Step3 is carried out. The percentage of patterns count reduction achieved is recorded. The equation to calculate the percentage of patterns count reduction is shown in Equation (6).

\[ \text{PercPatImpr} = 100 - \left( \frac{(Psc \times 100)}{Psc} \right) \hspace{1cm} (6) \]

where Psc is the patterns count recorded for ‘NSC’ and Psc is the patterns count recorder at same coverage as ‘NSC’.

![Figure 6](image6.png) *Figure 6* The length of the decompressor register and the longest internal scan chain length

![Figure 7](image7.png) *Figure 7* Scan synthesis and patterns generation for both scan compression and ‘New Scan Compression’ method [24]
It is important to have the chain lengths of the scan match the scan operation of the compression scheme, which is the basic architectural constraint in this method. Any mismatch in chain lengths results in a sub-optimal situation.

3 | EXPERIMENTAL RESULTS

The authors considered CUTs having a different number of SFFs. The name of the circuit is shown in column 1 of Table 3. The CUTs used in our experiment with SFFs, gates count, and the ratio of gates to SFFs have been shown in columns 2, 3, and 4, respectively, in Table 3. The results are also generated for different compression ratios with input and output ports.

Table 4 shows the percentage of test patterns count reduction for each CUT. The CUT used is shown in column 1, the minimum and maximum reduction in patterns count achieved using ‘NSC’ method is also shown in columns 2 and 3, respectively. The pattern count reduction achieved varies according to the compression ratio. Compression ratio changes dependencies across scan cells and impacts pattern inflation and test coverage and the range of pattern count reduction achieved from minimum to maximum is shown in the Table 4. The scan synthesis, writing out output net list and protocol file is done using DFTMax-Ultra. The generation of test pattern set covering all the faults including stuck-at, transition, bridging, delay faults etc. is done using the TetraMax tool. The ‘NSC’ method can be used on any type of test pattern set. It can be used on stuck-at or transition faults also. The ‘NSC’ method is the hybrid approach which is the mixture of scan and scan compression. This method does not change in the way ATPG test patterns are generated. The ‘NSC’ method proposed imposes a very little congestion as less than 0.5% of the SFFs that are moved out of the scan compression architecture.

Table 5 shows the results captured for the scan compression and ‘NSC’ method which is mixture of both the scan and scan compression. Each column in Table 5 is numbered from 1 to 8. Column 1 represents the name of the CUT; Column 2 represents the number of input and output ports assigned to the compression architecture. In the ‘NSC’ method, 1 scan-input and scan-output port out of total ports budget is assigned to the external scan chain and remaining ports to the scan compression architecture. Hence over-all scan-input and scan-output ports budget remains the same. Column 3 represents the internal scan chains present within the codec to generate results. Columns 4 and 5 represent the total test patterns count and test coverage respectively achieved for each configuration for scan compression architecture. The results shown in Columns 4 and 5 are generated using the DFTMax-Ultra [41] scan synthesis tool. Columns 6 and 7 represent the total test patterns count and test coverage respectively achieved for the proposed ‘NSC’ architecture for the same configuration. The proposed ‘NSC’ results are compared with the results generated using DFTMax-Ultra. Though the results are obtained from this technology, the analysis is applicable to other compression architecture also. Finally, column 8 represents the benefits of the ‘NSC’ method achieved in the percentage of the patterns count reduction on comparing Column 4 with column 6. The reduction of the patterns count for the same coverage is important in reducing the overall test cost of an IC. Reduced patterns count reduces the overall testing time of an IC. The significant reduction in the patterns count reduction up to 78.14% is achieved with the ‘NSC’ method. The ‘NSC’ method introduces small congestion as less than 0.5% of the SFFs that are moved out of the scan compression scheme. There is no extra area overhead of the ‘NSC’ architecture compared to scan compression. The ‘NSC’ architecture is built using scan compression architecture [41]. It uses the same decompressor and compressor hardware. Hence area overhead of ‘NSC’ is equal to the compression architecture used. The ‘NSC’ also reduces TAT as patterns count reduction results in a reduction in TAT. If ChL is the scan configuration chain length, the compression chain length is ChL/100, and with Tp is the number of test patterns, then the TAT is calculated as shown in Equation (7).

$$TAT = \left(\frac{1}{100}\right) \times ChL \times Tp$$ \hspace{1cm} (7)

| CUT | #scan flip-flops | #Gates | #Gates/scan flip-flops ratio |
|-----|------------------|--------|-----------------------------|
| C1  | 147K             | 1158K  | 7.87                        |
| C2  | 28K              | 395K   | 13.90                       |
| C3  | 86K              | 896K   | 10.35                       |
| C4  | 530K             | 4166K  | 7.86                        |
| C5  | 383K             | 2932K  | 7.65                        |
| C6  | 330K             | 3078K  | 9.32                        |
| C7  | 232K             | 3032K  | 13.04                       |

| CUT | Minimum | Maximum  |
|-----|---------|----------|
| C1  | 11.00%  | 78.14%   |
| C2  | 36.00%  | 72.27%   |
| C3  | 26.00%  | 43.42%   |
| C4  | 8.22%   | 33.15%   |
| C5  | 14.75%  | 22.52%   |
| C6  | 37.12%  | 37.12%   |
| C7  | 12.40%  | 12.40%   |
4 | CONCLUSION

The proposed ‘NSC’ architecture exploited the advantages of the full scan chain and scan compression technique. The full scan technique has no dependency across the SFFs values, whereas the compression technique has free variable dependency introduced by the test compression architecture as many internal scan chains are fed test data from few ATE Channels. Only compression-introduced dependency is addressed here and authors tried to reduce this correlation by moving out few percentages of SFFs contributing to the higher percentage of uncompressible values from the compression architecture. These are the SFFs those contributing to higher correlation and increased pattern count to detect the fault in an IC. The ‘NSC’ architecture moves such SFFs out of the scan compression architecture and embedding them into the chain which is outside the compression architecture. The SFFs being moved out of the compression architecture are less than 0.5% of the total SFFs present in the CUT. The ‘NSC’ architecture proposed reduces patterns count up to 78.14% for the same test coverage. This also helps to reduce the overall cost of an IC manufacturing. The total budget of scan-in and scan-out ports remains same. One input port and one output port is taken from the total scan-in and scan-out ports budget and is assigned to the scan chain formed outside the compression architecture. We used DFTMax-Ultra for scan insertion and TetraMax for test patterns generation. The ‘NSC’ architecture can be implemented with any other scan synthesis and patterns generation tools exist in the industry.

ACKNOWLEDGEMENTS

We would like to thank the Editor-in-Chief, Associative Editor, and reviewers who provided constructive review feedback to improve this article.

ORCID

Pralhadrao V. Shantagiri © https://orcid.org/0000-0002-4934-9540

REFERENCES

1. Lee, T., Touba, N.A., Yang, J.S.: Enhancing test compression with dependency analysis for multiple expansion ratios. IEEE Trans. Comput. Aid. Design Integr. Circuits Syst. 36(9), 1571–1579 (2017)

### TABLE 5

| CUTs | #SI/SO | #Chains | Scan compression method [41] | Proposed NSC method | Improvement |
|------|--------|---------|-------------------------------|----------------------|-------------|
|      |        |         | #TPC (TDV) | %TC | #TPC (TDV) | %TC | %TPC (TDV) | #TPC (TDV) | %TC | %TPC (TDV) |
| C1   | 8      | 400     | 10009 | 95.40 | 8907 | 95.16 | 11.00 |
|      | 8      | 800     | 12636 | 94.42 | 8310 | 94.42 | 34.23 |
|      | 16     | 1600    | 11984 | 93.01 | 5917 | 93.01 | 50.62 |
|      | 16     | 2000    | 12663 | 92.28 | 2768 | 92.30 | 11.00 |
| C2   | 6      | 300     | 3221  | 74.62 | 2062 | 74.62 | 36.00 |
|      | 8      | 400     | 3599  | 74.62 | 998  | 74.62 | 72.27 |
|      | 8      | 800     | 3299  | 77.47 | 1496 | 74.48 | 54.65 |
| C3   | 8      | 1200    | 4374  | 93.49 | 2475 | 93.49 | 43.42 |
|      | 8      | 2000    | 5949  | 93.54 | 2971 | 93.54 | 41.16 |
|      | 16     | 3200    | 4085  | 93.53 | 3024 | 93.53 | 26.00 |
| C4   | 8      | 400     | 9447  | 92.87 | 6315 | 92.87 | 33.15 |
|      | 8      | 800     | 11301 | 92.89 | 9638 | 92.89 | 14.72 |
|      | 16     | 400     | 9149  | 92.86 | 6417 | 92.86 | 29.86 |
|      | 16     | 1200    | 7539  | 92.91 | 6919 | 92.91 | 8.22 |
| C5   | 8      | 400     | 10879 | 95.35 | 9148 | 95.35 | 6.00 |
|      | 8      | 800     | 14164 | 95.77 | 11710| 95.76 | 17.33 |
|      | 16     | 1200    | 14789 | 95.70 | 11458| 95.70 | 22.52 |
| C6   | 4      | 800     | 19576 | 95.96 | 12310| 96.00 | 37.12 |
| C7   | 24     | 800     | 11947 | 84.58 | 10466| 84.58 | 12.40 |

Note: TPC is test patterns count or TDV generated to achieve TC.
