Research Journal of Applied Sciences, Engineering and Technology 4(22): 4674-4677, 2012
ISSN: 2040-7467

Submitted: March 23, 2012 Accepted: April 14, 2012 Published: November 15, 2012

# Enhanced Test Data Compression of Conflict Bit Using Clustering Technique

[1]S. Saravanan and [2]Har Narayan Upadhyay
[1]School of Computing (SOC),
[2]School of Electrical and Electronics Engineering (SEEE), SASTRA University,
Thanjavur-613401, India

**Abstract:** The aim of this study is to implement enhanced test data compression of conflict bit using clustering technique. Huge test patterns, larger power consumption and more accessing time are the various challenges encountered by present System on Chip (SOC) design. Various compression techniques have been developed to minimize the huge test patterns by reducing the size of the data which saves space and transmission time. Test quality of the test pattern can be improved by test data compression. By finding the proper conflict bit ('U') the proposed algorithm generates test patterns having high reduction in test compression. Small numbers of test patterns are generated using clustering technique. With proper test pattern clustering it is possible to achieve high level of compression. Validation of the proposed method is found by experimental results on ISCAS'89 and shows that compression ratio is achieved by 79% with less conflict test pattern.

**Keywords:** Compression, conflict bit, test pattern clustering

## INTRODUCTION

The new challenges that System on Chip (SoC) mainly focuses are test data reduction, accessing time reduction and the amount of power consumption. Testing a complex chip requires huge amount of test data. The bandwidth required for the tester and the chip increases with increase in logic integration on a single chip. As a result the memory capacity of Automatic Test Equipment (ATE) exceeds to store large test data. So the external ATE has to face serious challenges. Nowadays compression techniques are used to reduce testing time and amount of test data stored on ATE. By 2013, as predicted by International Technology Roadmap for Semiconductor (IRTS, 2011), nearly 1000X compression will be required.

More different categories of test data compression can be classified based on test patterns. In Fixed to Fixed (F-F) category compression occurs on fixed number of test pattern and produces fixed number of compressed output. Similarly, if compression occurs on variable test pattern and produces variable number of compressed output, then it belongs to Variable to Variable (V-V) category. Fixed to Variable (F-V) test pattern and Variable to Fixed (V-F) test pattern are the other combination techniques.

Test cubes which are the combination of '0', '1' and 'X' (unspecified bit) are taken for each test pattern, identifying the specified bit and unspecified bit.

The collection of '0' and '1' marks the specified bits. Unspecified bits have only 'X' as identifiers. These unspecified bits dominate over the specified bits in test patterns when compared. Random values of '0' and '1' make the unspecified bits to be considered as specified, leading to less compression in scan based design. The conflict data 'U' is the combination of '0' and '1'. The objective of this study is to use clustering technique for enhanced test pattern compression of conflict bits. Initially 'X' is identified in the test pattern. Depending upon the 'X' bit count, test patterns are clustered. Conflict bit 'U' is observed with each cluster blocks. Thus test pattern is compressed with its conflict bit. Test pattern compression is also recognized as one of the major impact in cost of manufacturing testing (Rajski *et al*., 2004). It is noted that test patterns consist of a group of test data.

## EXISTING METHODOLOGY

Numerous techniques have been developed to compress test data. Lossless compression technique is one among them, where compression of test data is much more difficult. As test data volume has become an important problem, a lot of research has been done on compression for test data (Tauba, 2006). Existing studies deal with three different types of compression. In first type, a number of code words are used to encode test data in Code-based scheme based on specific properties which is embedded in bit strings of test data (Nourani and Tehranipour, 2005).

In second type it deals with Linear-decompression based Linear Feedback Shift Register (LFSR) and XOR circuits (Könemann, 1991; Mitra and Kim, 2006). In the

**Corresponding Author:** S. Saravanan, School of Computing (SOC), SASTRA University, Thanjavur-613401, India

last type, Broadcast-scan based type focuses on broadcasting the same test pattern to multiple scan chain (Hamzaoglu and Patel, 1999). Test pattern compression plays a major role in reducing cost per chip. Based on LFSR various compression techniques have been developed. By these methods efficient compression on scan based design was achieved (Könemann, 1991). Large amount of unnecessary transitions may occur in the scan chain due to the scanning in of the pseudo-random bits generated by the LFSR. A test cube is partitioned to several blocks logically and a hold flag is used to indicate the occurrence of a transition in a block. By increasing the non-transition blocks, test power can be reduced (Lee *et al*., 2009). Run length based techniques were discussed in Golomb (Chandra and Chakrabarty, 2001), FDR (Chandra and Chakrabarty, 2003). Nine coded compression is shown in (Nourani *et al*., 2004) and Block merging is in (El-Maleh, 2008).

A drawback of compression schemes based on LFSR reseeding is that the unspecified bits are filled with random values, which results in a large number of transitions during scan-in, thereby causing high-power dissipation. Data volume and test times are reduced primarily by compressing the don't-care (unspecified) bit information. The original specified bit density hence dominates the theoretical compression limits. Further compression can be achieved by focusing on opportunities to compress specified bit information in addition to the don't-care bit information (Könemann, 2003).

**Proposed system:** Test data compression is implemented for scan chain based design. It is noted that unspecified bits ('X') form more than 90% in a given particular test cube (Hamzaoglu and Patel, 1998). Hence test cubes are the collection of unspecified bits which are represented as test patterns. In these test patterns unspecific bits dominates the specific bits.

The proposed system is made up of two main steps. First step discuss about the identification of the test data and second step discuss about clustering the test pattern into various groups to find conflict bits. The main aim of the proposed system is to compress test pattern and to identify the number of conflict bit in the compressed test pattern.

Dividing the whole test data into two groups forms the first step apart from identifying all the conflict bits. Test patterns which have maximum number of '0' and '1' are clustered into a single group and maximum numbers of 'X' are clustered into another group. When a cluster group has maximum conflict bits, then compression of the test pattern is low, whereas when the cluster group has maximum unspecific bits then the compression of the test pattern is more. When the conflict bits are improper then the test pattern increases randomly. So proper clustering is required for conflict bits. Pattern clustering is found by the threshold value of specified bit with unspecified bit:

Table 1: Test pattern in normal form

| Cluster 1 test pattern |
|---|
| 1 1 X 1 0 0 |
| 1 X X 1 0 0 |
| X 1 X 0 X 0 |
| X X X 0 X X |
| Cluster 2 test pattern |
| 0 1 X 0 1 0 |
| 1 1 X 0 0 X |
| 0 X X X 1 0 |
| 0 X X X X X |

Table 2: Clustered test pattern

| Test pattern | No. of unknowns |
|---|---|
| 1 1 X 1 0 0 0 1 X 0 1 0 | 1 |
| 1 X X 1 0 0 1 1 X 0 0 X | 2 |
| X 1 X 0 X 00 X X X 1 0 | 3 |
| X X X 0 X X 0 X X X X X | 5 |

Table 3: Proposed conflict pattern clustering method

| | Conflict pattern clustering | | | | | |
|---|---|---|---|---|---|---|
| $D^K$ | $p_1$ | $p_2$ | $p_3$ | $p_4$ | $p_5$ | $p_6$ |
| $d^1$ | 1 | 1 | X | 1 | 0 | 0 |
| $d^2$ | 1 | X | X | 1 | 0 | 0 |
| $d^3$ | X | 1 | X | 0 | X | 0 |
| $d^4$ | X | X | X | 0 | X | X |
| $G_1^k$ | 1 | 1 | X | U | 0 | 0 |
| $d^5$ | 0 | 1 | X | 0 | 1 | 0 |
| $d^6$ | 1 | 1 | X | 0 | 0 | X |
| $d^7$ | 0 | X | X | X | 1 | 0 |
| $d^8$ | 0 | X | X | X | X | X |
| $G_2^k$ | U | 1 | X | 0 | U | 0 |

Threshold value (T) = ((Total test patterns–Specified test pattern) / Total test patterns) × 100 (1)

Here the threshold value is 0.3 as per the proposed system. As in Table 1, there are eight number of test pattern with two sets of test pattern in each group having various numbers of unspecified bits ('X'). The Threshold value ('T') is found by test division, which is performed as per Eq. (1). A group of test pattern is clustered together. The clustering method depends on the unspecified test data.

A pair of clustered test pattern is shown in Table 2. Every clustered group has a combination of 4 test patterns. Two such groups are formed, which is divided with respect to the threshold value calculated. Here each test pattern consists of 1, 2, 3 and 5 unspecified bits. The possibility of compression is found to be high. So the threshold value used is helpful to produce better result.

Second step is to cluster test pattern to find conflict bits, which is tested on each and every scan test patterns. Here large number of scan test pattern is divided into various groups of test pattern. Table 3 shows the basic idea behind the proposed compression. It has a set of n inputs obtained for a set of test cubes.

Let the set be taken as $G^k = (G_1^k, G_2^k, \ldots, G_n^k)$, where $G_i^k \in \{0, 1, X, U\}$. The input pattern is denoted as $p_i$ and assigned as X or 1 or 0. For the above mentioned data, the test pattern $D^k = \{d^1, d^2, d^3, d^4, d^5, d^6, d^7, d^8\}$ is a

deterministic test cube set. All these test cubes are combined into compressed pattern as $G_1^k$ and $G_2^k$. The compression operation gives the output as '0' when the entire corresponding test patterns are '0' and 'X' only. The combination operation gives the output as '1' when the entire corresponding test patterns are '1' and 'X' only. When 'X' alone comes in the input it produces the value 'X'. When the merge is taken between '0' and '1', it denotes 'U' as conflicting test pattern.

During clustering of test pattern consideration it is necessary to identify number of conflict bits which is produced in compression. In Table 3 test pattern compression method is shown by conflict pattern clustering. Here inputs $p_1, p_2,\ldots, p_6$ belongs to patterns for compression. The procedure for compression in Cluster A is as follows. For the input $p_1$ and $p_2$ the values are 1 and X. So the result is 1. The input $p_3$ contains only X, so the result is also X. In input $p_4$, the combination of 0 and 1 produces the result U. In inputs $p_5$ and $p_6$ the result for compression of 0 and X is 0. The results of Cluster A are shown as $G_1^k$. Similarly in Cluster B, for input $p_1$ the values to be compressed are 0 and 1 and produce the result U. For input $p_2$ the combination of 1 and X produces the result 1. For input $p_3$, only X comes and the result is also X. For the input $p_4$ and $p_6$ the values to be compressed are 0 and X and this produces the result 0. Input $p_5$ has 0 and 1 combination and it produces the conflict value U. The compressed values of Cluster B are shown in $G_2^k$.

In the proposed algorithm step 1 denotes the calculation of threshold value of specified and unspecified values. Proposed method is targeted for 0.3 threshold value in specified bit. Once the threshold value is found then preparation of different levels of list which is associated with more specified values and unspecified values is carried out. Thus values are given as L1 and L2 list. The list obtained is used to cluster the pattern in proper method and to reduce conflict bit in the compression. Thus the test pattern is reduced and compression is achieved.

Last two stages show the test pattern applied to proper Cluster Size $CS_{[x]}$. It can be denoted as CS 5, CS 10 etc., finally finding the value of conflict bit is calculated by proper clustering. Then the clustered test patterns are applied to scan chain design. The proposed algorithm identifies the number of conflict bit.

Algorithm ConflictPatternclustering (Patlist, n, l, $CS_{[x]}$)
Input: Patlist – list of patterns consist of 0's, 1's & X's
n – Number of patterns in Patlist
l – Length of each pattern
$CS_{[x]}$ – Cluster Size
Output: Calc Conflict

1. Find threshold as less number of X's
   Tcnt = countX (Patlist[1])
   For i = 2 to n
   Tmpcnt = countx (Patlist[i]);
   If tmpcnt < Tcnt then
   Tcnt = tmpcnt;
   Endif

2. Split the patterns into two lists L1 & L2
   L1 – Patterns contains xcnt numbers x's
   L2 – Patterns contains more than xcnt number of X's
   For I <- 1 to n
   Tmpcnt = countX(patlist [i])
   If tmpcnt = xcnt then
   Add patlist[i] into L1
   Else
   Add patlist [i] into L2
   Endif
   Next i.

3. Test pattern is applied to proper Cluster Size $CS_{[x]}$. This can be given as CS 5, CS 10, CS 15 etc.,
4. Calculate conflict bit in proposed $CS_{[x]}$

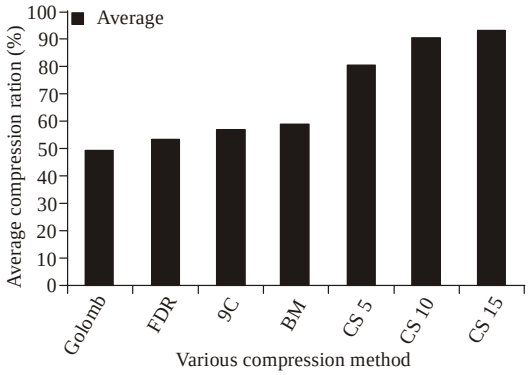


Fig.1: Proposed experimental results of average compression ratio in %

## EXPERIMENTAL RESULTS

The efficiency of achieving compression based conflict bit for scan test pattern is assessed. The proposed method is implemented in VHDL language and tested with full scan version of the ISCAS89 benchmark circuit. Test patterns used for these experiments are obtained by using Mintest dynamic compaction algorithm (Hamzaoglu and Patel, 1998). All these provided test patterns target on 100% fault coverage and its corresponding conflict activity.

The Fig. 1 shows the proposed experimental results of compression ration in average value. The Table 4 shows the experimental results of non-conflict test patterns. Table 5 shows proposed circuit, its original volume of test pattern, existing methods and proposed method. Various Cluster Sizes (CS) for ISCAS89 benchmark circuit is shown as CS 5, CS 10 and CS 15.

Table 4: Experimental results of non-conflict bits

| | Non- conflict bits in proposed method (%) | | | Compression ration of proposed method (%) | | |
|---|---|---|---|---|---|---|
| Circuit | CS 5 | CS 10 | CS 15 | CS 5 | CS 10 | CS 15 |
| S5378 | 83.9 | 65.0 | 56.3 | 79.2 | 89.1 | 92.7 |
| S9234 | 84.1 | 69.9 | 62.3 | 79.8 | 89.9 | 93.0 |
| S15850 | 93.0 | 80.1 | 71.3 | 79.3 | 89.6 | 92.8 |

Table 5: Compression ratio of proposed method

| Circuit | Original volume | Golomb% (Chandra and Chakrabarty, 2001) | FDR% (Chandra and Chakrabarty, 2003) | 9C% (Nourani *et al*., 2004) | BM% (El-Maleh, 2008) | Proposed method % | | |
|---|---|---|---|---|---|---|---|---|
| | | | | | | CS 5 | CS 10 | CS 15 |
| S5378 | 23754 | 37.11 | 47.98 | 51.64 | 54.98 | 79.2 | 89.1 | 92.7 |
| S9234 | 39273 | 45.25 | 43.61 | 50.91 | 51.19 | 79.8 | 89.9 | 93.0 |
| S15850 | 76986 | 62.82 | 66.21 | 66.38 | 69.49 | 79.3 | 89.6 | 92.8 |
| Average | ------- | 48.39 | 52.6 | 56.31 | 58.55 | 79.4 | 89.5 | 92.8 |

This also shows that cluster size of the test pattern is proportional to the compression technique. Due to less conflict bits higher compression is achieved.

Depending upon the occurrence of conflict bit in scan based test pattern, compression is carried out further. Table 5 shows that average value of compression ratio ranges from 79 to 92%. This also shows clustering size with less range can produce less number of conflict test patterns.

## CONCLUSION

The present challenge of reducing test pattern is one of the most important tasks in SoC testing. It is particularly identified in the field of scan cell test patterns. This study proposes a new algorithm of pattern clustering based on test pattern compression of conflict bit technique. In the proposed method, experimental results for the ISCAS89 benchmark circuit show significant reduction in compression ratio. It also shows clustering size is inversely propositional to conflict bits. Results show that 83 to 93% of non-conflict bit is achieved by using proper cluster size. Considerably low transition time and low data storage are achieved.

## REFERENCES


Chandra, A. and K. Chakrabarty, 2001. System-on-a-chip data Compression and decompression architecture based on Golomb codes. IEEE T. Comput. Aid. Des., 20(3): 355-368.

Chandra, A. and K. Chakrabarty, 2003. Test data compression and test resource partitioning for system-on-a-chip using Frequency-Directed Run-length (FDR) codes. IEEE T. Comput., 52(8): 1076-1088.

El-Maleh, A.H., 2008. Efficient test compression technique based on block merging. IET Comput. Digit. Tech., 2(5): 327-335.

Hamzaoglu, I. and J.H. Patel, 1999. Reducing test application time for full scan embedded cores. Digest of Papers 29th Ann. Int'l Symp. Fault-Tolerant Computing (FTCS 99), IEEE Press, pp: 260-267.

Hamzaoglu, L. and J.H. Patel, 1998. Test set compaction algorithms for combinational circuits. Proceeding of CAD, pp: 283-289.

International Technology Roadmap for Semiconductor (IRTS), 2011. Test and Test Equipment. San Francisco, CA.

Könemann, B., 1991. LFSR-coded test patterns for scan designs. Proceeding of Eur. Des. Test Conference, pp: 237-242.

Könemann, B., 2003. Care bit density and test cube clusters: Multi-level compression opportuneities. Proceeding of IEEE International Conference on Comput. Des., pp: 320-325.

Lee, L.J., W.D. Tseng, R.B. Lin and C.W. Yu, 2009. Deterministic built-in self-test using multiple linear feedback shift registers for low-power scan testing. Proceeding of Asian Test Symposium, pp: 111-116.

Mitra, S. and K.S. Kim, 2006. XPAND: An efficient test stimulus stimulus compression technique. IEEE T. Comput., 55(2): 163-173.

Nourani, M., M. Tehranipour and K. Chakrabarty, 2004. Nine-coded compression technique with application to reduced pin-count testing and flexible on-chip decompression. Proceeding of Design, Automation, Test in Europe, pp: 1284-1289.

Nourani, M. and M. Tehranipour, 2005. RL-Huffman encoding for test compression and power reduction in scan application. ACM Trans. Des. Autom. Electr. Syst., 10(1): 91-115.

Rajski, J., J. Tyszer, M. Kassab and N. Mukherjee, 2004. Embedded deterministic test. IEEE T. Comput. Aid. Des. Integ. Circuit. Syst., 23(5): 776-792.

Tauba, N.A., 2006. Survey of test vector compression techniques. IEEE Design Test Comput., 23(4): 294-303.