A Novel Method of Compression Based on Adaptive Encoding Technology for SoC Design

B.Sakthi Bharathi[1], S.Saravanan[2], R.Vijay Sai[3]
[1]M.Tech.VLSI Design, SASTRA University
[2]Assistant Professor, SASTRA University
[3]Assistant Professor, SASTRA University
Thanjavur, India.
[1]b.sakthibharathi@gmail.com,[2]saran@core.sastra.edu,[3]vijaysai@it.sastra.edu

***Abstract*- Test data compression is one of the main objectives to make reliable in testing System on chip design. The data compression mainly relies in the areas of hardware, time and power consumed. This paper presents an efficient test data compression method to achieve high compression ratio. The test patterns have 1, 0 and undefined (x) data. The test patterns are grouped such that to improve compression ratio. While grouping the undefined bits is also assigned as 1, 0 and conflict bit (c). The experimental result shows that the compression ratio achieved is highly desirable. It is targeted to ISCAS 89 bench mark circuit.**

Keywords- Test data compression, Conflict bits, Undefined bits.

I. INTRODUCTION

System on Chip (SoC) testing is one of the major issues to be considered in present design. Compression of test data is one of the methods to reduce its huge volume. The data compression involves encoding original information to fewer bits. Two types of compression are there, one is lossy and the other is lossless. Lossless compression means reducing redundant data's. In lossless, data's are never lost. In lossy, the data's are reducing by removing slightly valuable data's.

Compression is necessary to reduce area usage and transmission capacity in SoC design. In this paper, novel method is discussed, which uses an adaptive compression technology with Extended Frequency Direct Run length (EFDR). This method groups only compatible test data. The patterns that have unimportant conflict bits and undefined bits are grouped into a unique pattern and the chains are encoded. The test patterns are divided into number of groups for compression. It also encodes the chains of 1, 0 and Conflict (c) bits simultaneously. In the merged pattern the information regarding conflicting bits also stored.

II. RELATED WORK

The main challenge in System on Chip design (SoC) is large capacity of data. This large capacity of data exceeds the memory and automatic test data equipment channel capacity. To overcome this problem, data volume can be minimized. This in turn not only reduces the memory but also the time. In order to overcome these challenges many new techniques are proposed.

The concept of BIST has been proposed for testing particular cores. The BIST infix the test patterns in large sequences, but due to this technique the test time increases and is critical to meet out high fault coverage [1], then to overcome this problem considerable methods and various mechanisms have been suggested based on Linear Feedback Shift Register (LFSR) reseeding [2, 3]. LFSR reseeding have merit that patterns have very few defined bits and while compressing, the ratio of compression is direct proportional with number xs in pattern. If the specified bits go out of bound then LFSR reseeding is not efficient. To overcome this problem, distinct compression methods have been designed which includes Frequency directed run length coding (FDR) [4], Alternate run length coding using FDR [5], Statistical coding[6,7] RL-HC coding[12],9c Coding[10] and Extended Frequency Directed Run length coding(EFDR)[11].

The large number of cubes which has pre-determined mutually conflict bits are merged to increase the overall performance. Each array has one source pattern. Large number of patterns is obtained by appending additional bits to source pattern. To identify source pattern and information regarding location, limitation and accumulative pattern are analyzed [8].The Test patterns have both defined and conflict bits, so without neglecting the conflict bits it should also be considered for compressing. The conflict datas' are also specified with separate code words [9]. Instead of using different vectors the consecutive blocks can be merged by assigning compatible blocks inside, this is called block merging[13]

This paper, presents an innovative approach of adaptive compression technology which is made to compress the data efficiently. This method groups the Test patterns into different groups and then compression is done by compounding unimportant conflict bits and defined bits into a unique pattern. Then it encodes the chains of 1, 0 and C concurrently. This paper is based on the methodology of paper [9].

III. PROPOSED METHOD

If there are large number of test patterns and using all the data's at a time in architecture for compression consumes large area overhead so in order to minimize the area overhead, first the large number of test patterns are divided into small groups and compressed separately. Then the compressed values are again compressed to form the final output. This paper is the extension method of paper [9].

The test pattern has both conflict and undefined bits. The patterns are then divided and data's are compared and compounded by crediting 1 or 0 to undefined bits and C to conflict bits. Blended patterns have three kinds of chains i.e. 1s, 0s and Cs. Now the Compression needs two types of code to encode- One can encode 0s and 1s and the other can encode Cs. The chains of 0s and 1s are encoded using code word .The code word chains of 0's encode chains of 0's and code word chains of 1's encode chains of 1's. Two bits are said to be compatible if either the bits are same or any one bit is x. The most conflict bits are merged by crediting C. The chains of C are also encoded with code word. Consider test group which have five test patterns (TP):

TP1:00xxxx011xxxx10100xx

TP2:0xxxxxxx0xxxxxx00xxx

TP3:00xxxxxx1xxxx1011xxx

TP4:0xxxxxxx0xxxxxx00xxx

TP5:00xxxxxx1xxxxxx01xxx

The test patterns TP1 to TP5 are inputs which has 20 bits each. This TP1 to TP5 should be compressed to get output. These test patterns should be compressed by using following steps.

Compression is done by using above formula. The TP1 and TP2 are taken and compared to form TP and this is again compared with the consecutive test patterns to get T*.The compounded test pattern T* is {00xxxx01cxxxx10cc0xx}. Then crediting X with 0 or 1 by checking next immediate defined data. If the immediate defined data is 1 then previous x is replaced with 1 or if 0 then it is replaced with 0. We get fully specified test pattern T* is {00000001c111110cc000}. Then encoding T* with code words, we obtain the Encoded test CT_{pcode}= {0110000 100 10 11010 000 001 01000} which has 28 bits.

By using only the compressed value with code word, original data's can't be retrieved from the coded data, so to retrieve the original data we should know the conflict position. In order to identify the positions of conflict data's VI_c is derived. The value of VI_c is found by identifying number of conflict values in each test pattern VI_c = {110 000 111}. At last, the group can be encoded as code given below:

Let number of Test patterns is NT_p.

Let number of conflict bits be NI_c.

Let values of conflict bits be VI_c.

Let code of Test pattern is CT_{pcode}.

Concatenating the values {NT_p NI_c VI_c CT_{pcode}}

Therefore encoded group is {011 011 110 000 111 0110000 100 10 11010 000 001 01000}

A. ALGORITHM FOR COMPRESSION

Steps of compression

1. Get the Test patterns.
2. Divide the Test patterns into groups.
3. Take the first group of datas' and identify the compounded pattern by compressing each row of data.
4. Now replace x in compounded pattern by assigning 1 or 0 by identifying next immediate defined data.
5. Now the fully specified Test pattern has runs of 1, 0 and c.
6. Next replace the runs of 1, 0 and c with respective code words.
7. Repeat the above steps for whole Test patterns.

IV. EXPERIMENTAL RESULTS

This proposed method is compared with some bench mark circuits and the experimental results are shown here. The bench mark circuit taken here is **Tetramax Test cubes** for ISCAS89.

A particular test bench is taken and the patterns are divided into many cubes. These cubes are compressed separately and the compressed bits are taken. By this way all the cubes are compressed separately and the total compression ratio is achieved. Here for s5378 and s9234 the test patterns are grouped with 100 test patterns and for s13207 and s15850 the test patterns are grouped with 150 test patterns to achieve this compression ratio. This results shows that, by implementing the proposed method the compression ratio is high. Various compressions are tableted in Table1. Corresponding values are charted in Figure 1.

Compression ratio = ((original bits-compressed bits)/ original bits) × 100

Table 1: Comparison of different methods

Circuit	Number of Bits	Compressed Bits	Ratio [9]	Ratio [11]	Ratio [12]	Ratio [13]	Ratio [14]	Ratio [Proposed]
S5378	276274	2758	61.00%	51.64%	51.93%	53.75%	54.98%	99.00%
S9234	400387	4735	72.93%	50.91%	45.89%	47.59%	51.19%	98.81%
S13207	2294600	14640	89.65%	82.31%	81.85%	82.51%	84.89%	99.36%
S15850	2220374	19245	74.35%	66.38%	67.99%	67.34%	69.49%	99.13%

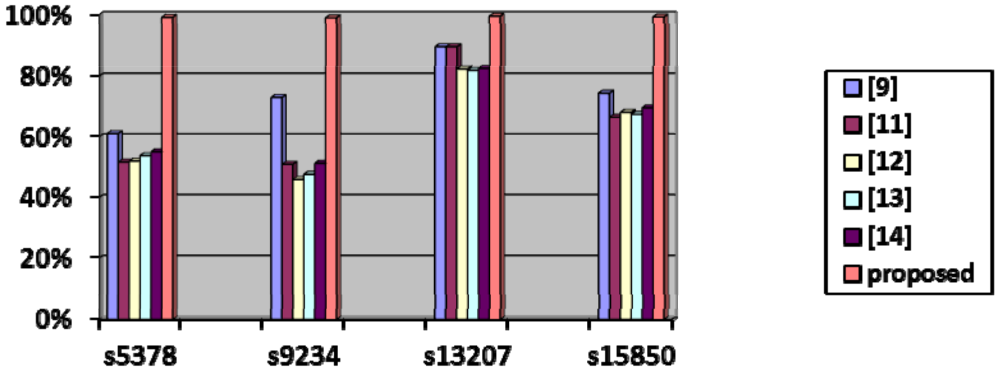

Figure 1: Comparison of different methods

V. CONCLUSION

In this paper, with the efficient adaptive compression algorithm proposed, the data's are effectively compressed and encoded. This proposed method encodes three chains of 1, 0 and C concurrently. This method achieves high compression ratio and results are compared with ISCAS89 bench mark circuit.

VI. REFERENCES

[1] Min-yong Wan, Yong Ding, Yun Pan and Xiao-lang Yan, An efficient compatibility based test data compression and its decoder architecture, Journal of Electron Test, 2011.

[2] Balakrishnan KJ and Touba A, Relationship between entropy and test data compression. IEEE Transactions on Computer-Aided Design Integrated Circuits Systems, 2006.

[3] El-Maleh AH, Efficient test compression technique based on block merging. IET Computer Digital Tech, 2008.

[4] Han Y, Yu Hu, Li X, Li H and Chandra A, Embedded test decompressor to reduce the required channels and pattern memory of tester for complex processor circuit, IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 2007.

[5] Kavousianos X, Kalligeros E, Nikolos D, Multilevel Huffman coding: an efficient test-data compression method for IP cores, IEEE Transactions on Computer-Aided Design Integrated Circuits Systems, 2007.

[6] Kavousianos X, Kalligeros E, Nikolos D, Optimal selective Huffman coding for test-data compression. IEEE Transaction on Computer-Aided Design Integrated Circuits Systems, 2007.

[7] Kavousianos X, Kalligeros E, Nikolos D, Test data compression based on variable-to-variable Huffman encoding with codeword reusability, IEEE Transactions on Computer-Aided Design Integrated Circuits Systems, 2008.

[8] Mrugalski G, Mukherjee N, Rajski J, Czysz D, Tyszer J, Compression based on deterministic pattern clustering of conflict test cubes. Proceedings International Test Conference, 2009.

[9] WanMY, DingY, PanY, Zhou S,YanXL, Test data compression using extended frequency-directed runs length code based on compatibility, Electron Letter, 2010.

[10] Tehranipoor, M., Nourani, M., and Chakrabarty, K, Nine-coded compression technique for testing embedded cores in SoCs', IEEE Transaction. Very Large Scale Integration (VLSI) System, 2005.

[11] El-maleh, A and Al-Abaji, R, Extended frequency directed run length codes with improved application to system-on-a-chip test data compression, Electronic Circuits and Systems, 2002.

[12] Nourani, M., and Tehranipour, M, RL, Huffman encoding for test compression and power reduction in scan application, ACM Transaction Electron. System,2005.

[13] El-maleh, A.H, Efficient test compression technique based on block merging, IET Computing Digital Technology, 2008