World Applied Sciences Journal 32 (11): 2229-2233, 2014
ISSN 1818-4952

DOI: 10.5829/idosi.wasj.2014.32.11.1325

A Combined Compatible Block Coding and Run Length Coding Techniques for Test Data Compression

[1]C. Kalamani and [2]K. Paramasivam

[1]Department of ECE, Dr. Mahalingam College of Engineering and Technology Coimbatore, Tamil Nadu, India
[2]Department of ECE, KSR College of Engineering, Erode, Tamil Nadu, India

Abstract: Higher Circuit Densities in system-on-chip (SOC) designs and increase in design complexity have led to drastic increase in test data volume. This results in long test application time and high tester memory requirement. Test Data Compression/Decompression addresses this problem by reducing the test data volume without affecting the overall system performance. This paper proposes a test data compression scheme that combines the advantages of compatible block coding followed by simple run lengthcoding techniques to address the large test data volume of Automatic Test Equipment. This test data compression technique significantly reduces memory requirements. The algorithm is applied on various benchmark circuits and compared results with existing test compression/Decompression techniques. The experimental evaluation revealed that the proposed method achieves on an average, a compression ratio of 70%. The proposed approach, improves the compression efficiency without introducing any additional decompression penalty. Experimental results demonstrate that this approach produces up to 30% better compression compared to existing methods.

Key words: SoCs (System on chip) · VLSI (Very large scale integration) · Finite State Machine (FSM)

INTRODUCTION

The complexity of VLSI continues to grow, more number of transistors are integrated on a single chip and test data volume has drastically increased. The testing cost and testing power are the two major issues in the current generation integrated chip testing. Testing cost is related to test data volume and test data transfer time. Larger data volume demands not only higher memory requirements but also increase in testing time. Test data compression addresses this problem by reducing the test data volume without affecting the overall system performance. Test data compression involves adding some additional on-chip hardware before and after the scan chains. This additional hardware decompresses the test stimulus coming from the tester; it is also compacting the response after the scan chains and before it goes to the tester. This permits storing the test data in a compressed form on the tester. It is also easier to adopt in industry because it is compatible with the conventional design rules and test generation flows for scan testing. Test data compression provides two benefits. First, it reduces the amount of data stored on the tester, which can extend the life of older testers that have limited memory. Second is the most important benefit, which applies even for testers with plenty of memory.It can reduce the test time for a given test data bandwidth. Test vectors are highly compressible because only 1% to 5% of their bits are specified (care) bits. Test data compression fall broadly into three categories. They are coding scheme, linear decompression scheme and broadcast- scan scheme [1]. Code compression techniques are popular because they offer both good compression ratio and fast decompression scheme. Code based test data Compression has an advantage in generating difference, reordering patterns and it achieves high compression. Many coding scheme has been proposed for code based scheme [2]. The Huffman code is proved to be an optical statistical code. However, it requires an exponentially large decoder. A few variant of Huffman coding exist, such as selective Huffman coding (SHC), which splits test vector into fixed pattern and

Corresponding Author: C. Kalamani, Department of ECE, Dr. Mahalingam College of Engineering and Technology Coimbatore, Tamil Nadu, India. E-mail: kalamec@yahoo.co.in.

applied a huffman coding for the selected number of patterns but occupies a large area overhead.Optimal selective Huffman coding (OSHC) further reduces the data.Variable length input Huffman coding (VIHC) comprises a mapping, reordering and Huffman encoded [3]. The run-length based compression method encodes a repeated runs. Different types of run length coding techniques are frequency directed run length code (FDR) consists of a prefix and a tail with same size and requires complicated decoder and inefficient for long run's of 1's [4], Extended frequency directed run length code (EFDR) takes advantage of both runs of 0's and runs of 1's and outperformed the other coding techniques that are based on only runs of 0's [5] took advantage of both runs of 0's and runs of 1's and outperformed the other coding techniques that are based only on runs of 0's. A combined run length and Huffman coding for scan testing is to reduce the test data volume [6]. Multidimensional pattern run length (MD-PRC) considers multiple pattern information for compressing variable length pattern runs [7]. Block merging technique records merged blocks and number of merged block to achieve higher test compression ratio and less test application time with higher area overhead [8]. Block merging and eight coding is encoding based number of merged blocks form eight code word [9]. The rest of the paper is organized as follows. Section 2 describes proposed methods of test data compression and decompression mechanism. Section 3 presents the experimental results. Finally, section 4 concludes the paper.

Proposed Methods: The proposed method combines compatible block coding technique and run length coding for reducing the test data volume during test operation. In the compression stage, the first step is blocks coding and the second step is run length coding. These techniques take advantage of existing large number of don't care bit in test data which assigned values of 0 or 1 according to the need. After assigned values test data blocks have relationship of compatible and inversely compatible. Based on the compatibility or inversely compatible, form a group and replace consecutive repeating occurrence of a symbol by one occurrence of a symbol followed by the number of occurrence. The resultant is a compressed data.

Step 1: Block Coding Techniques: There exists a large number of don't care bit in the test data which are assigned with values of 0's or 1's according to the need. After the assigned values the test data blocks have a relationship of compatible or inversely compatible. The basic idea of block coding method is compatibility or inversely compatibility and the data blocks are combined into one group, binary code is used to express the test data which are compatible or inversely compatible with reference data, by this way test data is compressed [9]. Two patterns are recognized as compatible if every bit pair at the same position has the same value or any of them are don't care bit or Two patterns are recognized as inversely compatible if every bit pair at the same position has the inverse value or any of them are don't care bit. The compatible block encoding is explained with an example below.

For Example: Test data 01xx0x11x1xx can be divided into three sub-segments such as 01xx, x1xx, 0x11.The sub segment 01xx is taken as reference test data and last 2-sub segments will be found compatible with the first sub segment. In this case, the last 2 sub segments can be coded 001, the first bit "0" means last 2 sub segments and reference data is compatible ("1" for inverse compatible), "01" represent the quantity of compatible data block. Table 1 shows the relationship of uncompressed data and compressed data [10]. In Table 1, sign "0" represents that data blocks are compatible, sign "1" represents that data blocks are inversely compatible.

Step 2. Run Length Coding: Simple Run length coding is applied to get better compression ratio in the second step. RL coding is simple and more efficient, if the data uses only two symbols (for example 0 and 1) in its bit pattern and one symbol is more frequent than the other. Run length codingreplaces consecutive repeating occurrence of a symbol, by one occurrence of a symbol followed by the number of occurrence [6]. The Run Length based coding schemes have been very effective for the test data compression in case of current generation SOCs (System on chip) with a large number of IP(Intellectual property) cores. In the proposed method, the original data is block coded followed by Run Length encoded as below.

The steps used in encoding are as follows.

- Given parameter k(block size) and m(binary code), the original data is divided into blocks of length of the data block.
- From the initial position in order to choose the length of k data blocks as reference data block, then $s=2^m$ data block and check the reference data block is compatible(inversely compatible)then add 1 behind the reference data block and according to the Table 1 jump to step 4(6) or else jump to the step 3.

Table 1: Coding Scheme

Data block no	Signed bit	Code word	Signed bit	Code word
1	0	001	1	000
2	0	010	1	010
3	0	011	1	011
4	0	100	1	100
5	0	101	1	101
6	0	110	1	110
7	0	111	1	111
8	0	000	1	000

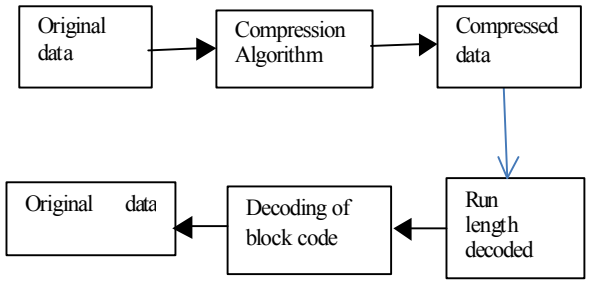

Fig. 1: Test data compression/decompression methods

- If s>0, then s=s-1 and jump to step 2 or else jump to step 4.
- If the followed j=2^m data block and reference data block is inversely compatible, jump to step6 or else jump to step5.
- If j>0, then j=j-1 and jump to step 4 or else jump to step 6.
- Add the sign at the end "0", a reference data block coding end.
- Find the runs of 0's and 1's, in the block coded data to calculate the length of block size and frequency of occurrence.
- Replace consecutive repeating occurrence of a symbol by one occurrence of a symbol followed by the number of occurrence
- Finally find the compressed ratio.

For example:

- Consider input data after preprocessing

11111111
11111111
10100000
00011000
11111111
00000000
10100000
01011111

- Eliminate Unique Data

0 0 0 0 0 0 0 0
0 0 0 1 1 0 0 0
0 1 0 1 1 1 1 1
1 0 1 0 0 0 0 0
1 1 1 1 1 1 1 1

- Find Compatible and incompatible

0 0 0 0 0 0 0 0
0 0 0 1 1 0 0 0
0 1 0 1 1 1 1 1

- Compressed Data

1 0 1 1 1 1 1 1 0 0 0 0

In the second stage of decompression the compressed data is run length decoded followed by block decoded. Block decoder is Finite state machine decoder.

The steps used in decoding are as follows.

- Compressed data is an input to run length decoder.
- Initialize first bit as 1 and then add first bit to the previous bit. This is used to find the bit position change.
- Initialize the variable as 0. Find the length of the variable
- Decompression is performed based on the bit position change in the initialized variable.
- Run length decompressed data are divided into blocks.
- Blocks of data are divided into two rows.
- The first bit of the data is analyzed for compatible or inversely compatible.
- If the bit is inversely compatible, the reference data will be inverted else the bit is remains same.
- Finally obtain the original data.

RESULTS AND DISCUSSION

In order to demonstrate the proposed scheme, compression and decompression modules are implemented using MATLAB-13 and experiments are carried out using large ISCAS 89 and ISCAS 85 benchmark Circuits. The compression ratio is defined in "Eq.1" is used to evaluate the performance of the proposed scheme.

Table 2: Compression Ratios Obtained By Proposed Method Compared With Various Existing Methods

	Compression Ratio (%)				
Circuits	Golomb Codes [1]	FDR Codes [2]	VIHC [5]	Compatible block Coding [6]	Proposed Method
C432	-	-	-	-	68.96
C499	-	-	-	-	69.51
C880	-	-	-	-	69.06
C4315	-	-	-	-	68.84
S5378	40.7	48.02	51.78	54.16	72.615
S9234	43.34	43.59	47.25	53.47	73.086
S38417	44.12	43.26	53.36	56.74	72.72
Average	42.62	44.95	50.79	54.79	72.80

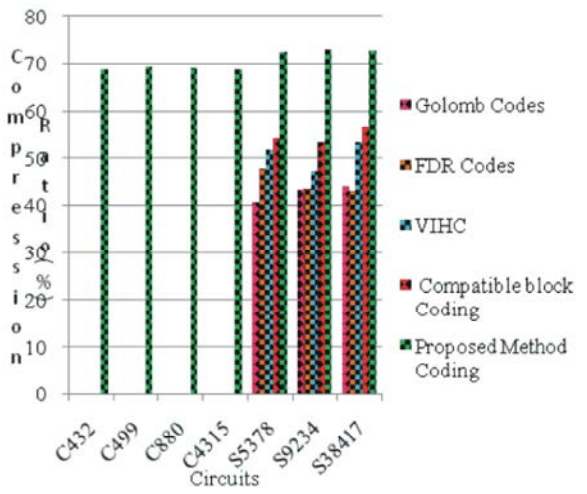

Fig. 2: Comparison Of Compression Ratio Of Proposed Method With Various Existing Method

Compression Ratio = 1 – [*compressed data* ÷ *original daa*] * 100

(1)

The compression ratio of proposed methods and various existing methods circuits are shown in the Table 2. The proposed methods shows better compression than other methods because of combining block coding and run length coding techniques. The compatible block coded data consists the consecutive occurrence of 1's or 0's is reduced by applying simple run length encoding.This presents increase compression ratio than compatible block code. Table 2 and Fig. 2. shows the comparison of compression ratio of proposed method with existing methods. The average compression ratio is increased by 30.2,27.85,22.01,18.01% compared with existing methods in table2.The combined compatible block coding and run length coding shows better compression ratio. The decompression circuits are simple and easy to implement and reproduce the original data without loss.

CONCLUSION

Large data volume is the most important issues in testing a VLSI circuit. Code compression technique offers an efficient solution for this issue. The compatible block coding can encode the compatible and inversely compatible test data blocks followed by run length technique reduce the large data volume, produces a better compression ratio. The proposed algorithm is applied on four ISCAS 85 and three ISCAS89 benchmarks and compared with the results of the existing test compression technique. The result demonstrates that the proposed technique obtained an average compression ratio of 70%. The original data is reconstructed without any loss of data.

REFERENCES

1. Touba Nur A., 2006. "Survey of test vector compression techniques.", IEEE Des Test Comput., 23(4): 294-304.
2 Ruan, X. and R. Katti, 2006. "An efficient data-independent technique for compressing test vectors in systems-on-a-chip", in Proc. ISVLSI, pp: 153-158.
3. Gonciari, P.T., 2003. "Variable length input Huffman coding for system-on-a-chip test", IEEE Trans. Comput. Aided Des. Integr. Circuits Syst., 22(6): 783-796.
4. 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 Trans. Comp., 52: 1076-1088.
5. El-malch, A.H., *et al*., 2008. "Test data compression for system-on-a-chip using extended frequency-directed run-length (EFDR) code", IET Comput. Digit. Tech., 2(3): 155-163.

6. Nourani, M. and M. Tehranipour, 2005. "RL-Huffman encoding for test compression and power reduction in scan application", ACM Trans. Des. Automat. Electron. Syst., 10(1): 91-115.
7. Lee, L.J., *et al*., 2009. "A multi-dimensional pattern run-length method for test data compression.", in Proc. AsianTest Symp., pp: 111-116.
8. El-malch, A.H., *et al*., 2008. "Efficient test compression technique based on block merging", IET Comput. Digital Tech., 2: 327-335.
9. Tie-bin Wu, *et al*., 2013. "Eficient Test compression technique for soc based on block merging and eight coding", J. Electron Test, 29: 849-859.
10. Jin, Shang and Liyong Zhang, 2013. "Test Data Compression Scheme Based On Compatible Data Block Coding.", Information Technology Journal, 12(1): 204-208, ISSN 1812-5638.
00. Chandra, A. and K. Chakrabarty, 2001. "System-on-a-chip data compression and decompression architecture based on a Golomb codes.", IEEE Trans. Comput Aided Des. Integr. Circuits Syst., 20(3): 355-368.