International Journal of Computer Applications Technology and Research
Volume 2– Issue 2, 152 - 154, 2013, ISSN: 2319–8656

A Novel Method for Encoding Data Firmness in VLSI Circuits

V. Karthikeyan
Department of ECE
SVS College of Engineering
Coimbatore, India

V. J. Vijayalakshmi
Department of EEE
Sri Krishna College Of Engineering & Technology
Coimbatore, India

P. Jeyakumar
Department of ECE
Karpagam University
Coimbatore, India

Abstract: The number of tests, corresponding test data volume and test time increase with each new fabrication process technology. Higher circuit densities in system-on-chip (SOC) designs have led to drastic increase in test data volume. Larger test data size demands not only higher memory requirements, but also an increase in testing power and time. Test data compression method can be used to solve this problem by reducing the test data volume without affecting the overall system performance. The original test data is compressed and stored in the memory. Thus, the memory size is significantly reduced. The proposed approach combines the selective encoding method and dictionary based encoding method that reduces test data volume and test application time for testing. The experiment is done on combinational benchmark circuit that designed using Tanner tool and the encoding algorithm is implemented using Model -Sim

Key words: Test data volume, Test data compression, Selective encoding, Dictionary based encoding

1. INTRODUCTION

Testing is a process of checking the fabricated IC's for any incorrect behaviour due to faults like logical fault, delay fault, fabrication faults [1]. Test data volume is now recognized as a major contributor to the cost of manufacturing testing of integrated circuits (ICs). Test vector compression has been an active area of research, yielding a wide variety of techniques. There are various methods for test data compression but code based data compression scheme is more useful for intellectual property (IP) cores, since their structure is often hidden from the system integrator [2]. A test pattern compression scheme is proposed in order to reduce test data volume and generation time. Test application time and test data volume is affected by a number of parameters, such as the number of patterns ,the number of scan chains, and number of scan cells and the scan frequency. Scan design methods reduce test data volume and test application time by partitioning scan chains to shorter segments and broadcast the same test data to multiple scan chains. While compaction schemes try to reduce the number of patterns generated without compromising fault coverage levels, compression schemes in turn target reduction of the storage requirements of the compacted test patterns. While compression of the original test vectors provides some level of test volume reduction, the compression schemes also try to compress the difference between the successive vectors to improve compression efficiency. Data compression techniques are used to alleviate the ATE test data volume problem.Dictionary-based test data compression is a promising approach for test data volume reduction. they provide a dual advantage of good compression efficiency as well as fast decompression mechanism. We present a selective encoding method that reduces test data volume and test application time for the scan testing of IP cores. This method encodes the slices of test data that are fed to the scan chains in every clock cycle. Unlike many prior methods, the proposed method does not encode all the specified (0's and 1's) and unspecified (don't care) bits in a slice. They encode only the target symbols.

2. EXISTING METHODS

Many research works have tried to solve the data volume problem in high performance VLSI circuits. An efficient test data compression technique reduces the test data volume considerably. Test data compression offers a promising solution to the problem of increasing test data volume. A test set for the circuit under test (CUT) is compressed to a much smaller data set, which is stored in ATE memory. Huffman codes are the most effective ones, since it probably result in the shortest average codeword length, Then the main problem is the high hardware overhead of the required de-compressors. Huffman decoding leads to both large de-compressors and very low compression ratios. Since the vast majority of the cores have multiple scan chains, a serial-in, parallel-out register must be used for spreading the decoded data in them and thus, no test-time savingsLei Li, Krishnendu Chakrabarty and Nur A. Touba present dictionary-based test data compression approach for reducing test data volume in SOCs. The proposed method is based on the use of a small number of ATE channels to deliver compressed test patterns from the tester to the chip and to drive a large number of internal scan chains in the circuit under test. Jun Liu1, Yinhe Han1, Xiaowei Li1, propose an extended selective encoding which presents Flexible grouping strategy is able to decrease the number of encoded groups to improve compression ratio. It can exploit a large number of don't care bits to reduce testing power with no compression ratio loss.In the selective Huffman coding method of compression, the area overhead is high compared to other techniques. In the dictionary based approach the occurrence of mismatches will be high. The bitmask based compression method creates more matching patterns but the compression efficiency is less compared to the dictionary based approach. The bit encoding method encodes all the bits in the data. An efficient compression method can be achieved by encoding the selected symbols in the test vector set. That is only the targeted symbols will be encoded in this method. And hence the compression efficiency will be more compared to other techniques. But the efficiency is only for those circuits which are having test vectors with higher length of runs of zeroes or ones.

3.BACKGROUND AND RELATED WORK

Data compression technique is eliminating coarse-grained redundant data, typically to improve storage utilization. It helps reduce the consumption of expensive resources, such as hard disk space or transmission bandwidth. Compression is used just about everywhere. The task of compression consists of two components, an encoding algorithm that takes a data and generates a "compressed" representation (hopefully with fewer bits), and a decoding algorithm that reconstructs the original data or some approximation of it from the compressed representation. These two components are typically intricately tied together since both have to understand the shared compressed representation. The amount of data required to test ICs is growing rapidly in each new generation of technology. Increasing integration density results in larger designs with more scan cells and more faults. Moreover, achieving high test quality in ever smaller geometries requires more test patterns targeting delay faults and other fault models beyond stuck at faults. Conventional external testing involves storing all test vectors and test response on an external tester, which is ATE. Test data compression consists of test vector compression on the input side and response compaction on the output side. Test vector compression has been an active area of research, yielding a wide variety of techniques. The proposed approach combines the selective encoding method and dictionary based encoding method that reduces test data volume and test application time for testing.

4. PROPOSED METHOD

4.1 Dictionary Based Encoding

This method combines the dictionary base encoding method and the selective encoding method Dictionary-based methods are quite common in the data compression domain. While statistical methods use a statistical model of the data and encode the symbols using variable-size codewords according to their frequencies of occurrence, dictionary-based methods select strings of the symbols to establish a dictionary, and then encode them into equal-size tokens using the dictionary. The dictionary stores the strings, and it may be either static or dynamic (adaptive). The former is permanent, sometimes allowing for the addition of strings but no deletions, whereas the latter holds strings previously found in the input stream, allowing for additions and deletions of strings as new input is processed. A simple example of a static dictionary is an English dictionary used to encode English text that consists of words. A word in the input text is encoded as an index to the dictionary if it appears in the dictionary. Otherwise it is encoded as the size of the word followed by the word itself. In order to distinguish between the index and the raw word, a flag bit needs to be added to each codeword.

In the dictionary-based test data compression method, each codeword is composed of a prefix and a stem. The prefix is a 1-bit identifier that indicates whether the stem is a dictionary index or a word of uncompressed test data. If it equals 1, the stem is viewed as a dictionary index. On the other hand, if the prefix equals 0, the stem is an uncompressed word and it is *m* bits long. The length of the dictionary index depends on the size of the dictionary. If *D* is the set of the entries in the dictionary, the length of the index l_{index}=ceil [log$_2$ |*D*|, where|D| is the size of the dictionary. Since *l*index is much smaller than *m*, the compression efficiency is greater if more test data words can be obtained from the dictionary. However, the dictionary must be reasonably small to keep the hardware overhead low. Fortunately, since there are many don't-care bits in scan test data for typical circuits, we can appropriately map these don't-care bits to binary values and carefully select the entries for the dictionary, so that as many words as possible are mapped to the entries in the dictionary.

4.2 Selective Encoding Method

This approach encodes the slices of test data. Each slice is encoded as a series of C-bit slice-codes, where C=K+2, K=ceil (log2 (N+1)), and N is the number of internal scan chains in the CUT. The number of slice codes needed to encode a given slice depends on the distribution of 1's, 0's, and don't cares in the slice. The proposed technique does not require dedicated test pins for each core in SoC. If cores are tested sequentially, only one common test interface is needed. If some cores are tested in parallel, then they can together be viewed as a larger core with more scan chains. However, if the test sets for the cores are delivered with the don't care bits to the system integrator, an appropriate compression method can be used at the system level to reduce test data volume and testing time. This imposes no additional burden on the core vendor. Un-modelled faults can still be detected if the compression method does not arbitrarily map all don't cares to either 1's or 0's. The proposed approach only encodes a subset of the specified bits in a slice. First, the encoding procedure examines the slice and determines the number of 0s and 1svalued bits. If there are more 1's (0's) than 0's (1's), then all X's in this slice are mapped to 1(0), and only 0's (1's) are encoded. The 0's (1's) are referred to as target-symbol*s* and are encoded into two data codes in two modes. i) Single bit mode and ii) Group copy mode. In the single-bit-mode, each bit in a slice is indexed from 0 to N-1. A target-symbol is represented by a data-code that takes the value of its index. In the group-copy mode, a bit slice is divided into groups, and each group is bits wide. If a given group contains more than one target symbol, then the group-copy mode is used. Two data codes are needed to encode a group. The first data code specifies the index of the first bit of the group, and the second data code contains the actual data. In the group-copy mode, don't cares can be randomly filled instead of being mapped to 0 or 1 by compression scheme. The encoding procedure is as follows:

Step1: Format the given test vectors into slices

Step2: For each slice, determine the number of 0s (k0) and 1s (k1) in the slice.

Step3: If k0 > k1, then target-Symbol: = 1, 1st control-code: = 00;   else target-Symbol: = 0; 1st control-code: = 01;

```
Step4: For each group of the slice, calculate the number of target-symbols;

Step5: If number-of-target-symbols > 1 then encode the group using the group-copy-mode;

    else encode the group using the single-bit-mode;

Step6: End for (group);

Step7: Generate slice-codes for the current slice;

Step8: End for (slice);
```

In Step 1, each test vector is divided into a series of slices. Encode each slice as a series of slice codes. In Steps 2–3, the numbers of 0's and 1's are calculated, and the target symbol as well as the control code of the first slice code is set. The first slice code of each slice must contain an initial-control code (00 or 01). Steps 4-6 encode all the groups of a slice. For each group of a slice, if it contains more than one target symbol, it is encoded using the group-copy mode, otherwise it is encoded using the single-bit mode. Once all groups have been encoded, the slice code generation step (Step 7) becomes straightforward.

5. CONCLUSION

In VLSI design process, data volume minimization and power optimization is major concern. In this project a novel method is proposed for test data volume minimization. The proposed algorithm is to reduce the data volume by compressing the test vectors. The proposed method delivers compressed patterns from the tester to the chip.

6. REFERENCES

[1] M. Abramovici, M.A. Breuer and A.D. Friedman, "Digital Systems Testing And Testable Design", Computer Science Press, 1990

[2]N. A. Tauba, "Survey of Test Vector Compression Techniques", IEEE Transaction Design & Test of Computers, 2006.

[3] Zhanglei Wang, Krishnendu Chakrabarty, "Test Data Compression Using Selective Encoding of Scan Slices" IEEE transactions on Very Large Scale Integration (VLSI) systems, Vol. 16, No. 11, November 2008.

[4] Usha S. Mehta, Niranjan M Devashrayee, Kanker S. Dasgupta, "Hamming Distance Based 2-D Reordering With Power Efficient Don't Care Bit Filling Optimizing the Test Data Compression Method", IEEE, 2010.

[5] Patrick Girard, Laboratory of Informatics, Robotics and Microelectronics of Montpellier "Survey of Low Power Testing of VLSI Circuits", IEEE Design & Test of Computers, 2002.

[6] Witold A. Pleskacz, Tomasz Borejko, Tomasz Gugala, Pawel Pizon and Viera Stopjakova,Def Sim –The Educational Integrated Circuit for Defect Simulation, MSE'05 Anaheim, California, USA– June 12-13, 2005.

[7]K.Paramasivam, Dr.K.Gunavathi, Reordering Algorithm for Minimizing Test Power in VLSI Circuits, Engineering Letters Vol. 14, No. 1, February 2007, pp: 78-83.

[8] Lei Li , Krishnendu Chakrabarty, Nur A. Touba, "Test Data Compression Using Dictionaries with Selective Entries and Fixed- Length Indices", ACM Transactions on Design Automation of Electronic Systems, Vol. 8, No. 4, October 2003, Pages 470–490.

[9] http://www.ece.uic.edu/~masud/resources.html