Reduced Test Pattern Generation using Reconfigurable Compression Techniques for Testing Soc

V.Vinothini[1], S.Yamuna[2]
[1](M.E (VLSI Design), sri shakthi institute of engineering and technology, coimbatore, India)
[2](Assistant professor (ECE Dept), Sri shakthi institute of engineering and technology, coimbatore, India)

ABSTRACT : *This paper present a hybrid test vector compression method reducing the testing time and test data volume size of SOC. Higher circuit densities in system-on-chip (SOC) designs have led to drastic increase in test data volume size. Today the silicon densities increase day by day, so it become very difficultly to test SOC, because it need the large test pattern. The method consists of two steps: Burrow wheeler transform (BWT) and pattern overlapping method. BWT is a block sorting lossless and reversible data transform. It improves the efficiency of text compression algorithm. The hybrid based compression scheme provides the high compression ratio and fast testing time. The major contributions of this paper are as follows: 1) it develops an efficient pattern overlapping technique for test data in order to create maximum matching patterns. 2) it proposes a test compression technique using efficient pattern overlapping method to significantly reduce the testing time and memory requirements of SOC.*

Keywords *- Automatic test equipment (ATE), Burrows– Wheeler transformation (BWT), design-for-testability (DFT), intellectual property (IP) core, system-on-chip (SoC) test, pattern overlapping method.*

I. INTRODUCTION

In System-On-Chip (soc) designs, higher circuit densities have led to larger volume of test data, which demands larger memory requirement in addition to an increased testing time. Test data compression plays a crucial role, reducing the testing time and memory requirements An important objective to realize through elaborate testing of very large scale integration (VLSI) circuits and systems is to ensure that the manufactured products are free from defects and simultaneously guarantee that they meet deemed specifications.

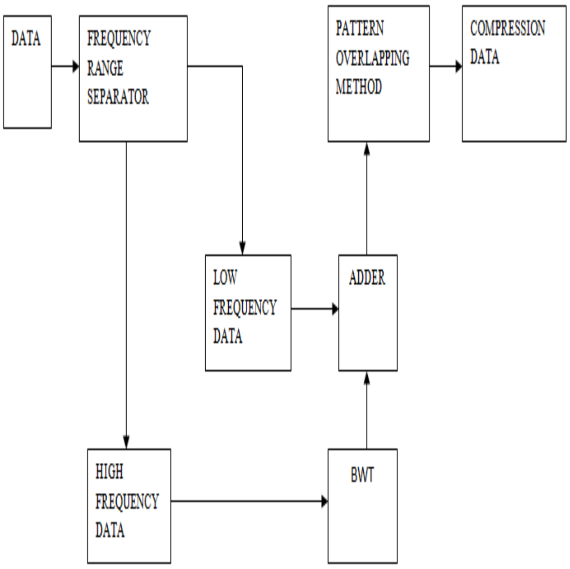


Fig. 1. Block diagram of the proposed method.

In addition, the information collected during the test process may help in an increase of the product yield by improving the process technology with consequent lowering of the production cost. The integrated circuit (IC) fabrication process involves various steps, viz., photolithography, printing, etching, and doping. Testing SOC is very complexity because soc contain many IP cores. An SOC often includes multiple types of circuitry, such as digital logic, memories, and analog circuitry. To reduce the complexity of testing use the hybrid test vector compression to minimize the test data volume and testing time by using the two method BWT(burrow wheeler transform) and Pattern overlapping method.

II. PROPOSED METHODOLOGY

The Fig. 1 shows the block diagram of the developed technique. All the test vectors required for testing an SoC are first compressed in software mode. The compressed test vectors and an efficient decompression program are then loaded into the embedded processor core of the SoC. The processor executes the decompression program and then applies all the uncompressed original test vectors to each and every core of the SoC for generating and analyzing the output responses. In the execution of the proposed technique, the undernoted four steps are involved.

2.1 . Division of Test Data Into Blocks

All the test vectors are divided into several blocks of equal size; the size of a block depends on the total number of bits in each vector.

Block number	1	2	3	4	5	6	7
Vector 1	0100	1100	0001	1000	0110	1000	0111
Vector 2	0110			1011	0011		0101
Vector 3	0100	1001			0010		0010

Fig. 2. Original test vector divided into several blocks.

As shown in Fig. 2, test vector-1 has seven blocks of size four bits each. The test vector-2 also has seven blocks of the same size but all the bits in its second, third, and sixth blocks are the same as that in the test vector-1.Similarly, the third, fourth, and sixth blocks in the testvector-3 are the same as in the test vector-2. If we have the information of the first, fourth, aft, and seventh blocks in the test vector- 2 along with the reference test vector-1, then, we can easily compute the entire set of test vector-2. One of the test vectors is then considered as a reference test vector and the next test vector is generated from the previous vector by storing only those blocks that differ from the previous one

2.2. Frequency Computation of Data Blocks

On completion of this block matching process, the high frequency and low frequency blocks are separated for further computation (viz., high frequency groups such as data sets in columns 1, 5, and 7 and low frequency groups such as data sets in columns 2, 3, 4, and 6 in Fig. 2).

There will be lots of unspecified data (represented by x for *don't care*) that appear in practical situations. Those data are redefined in such a way that we can maximize the frequency of data blocks.

Block number	1	2	3	4	5	6	7
Vector 1	01XX	0XX0	X0X0	10XX	X1X1	0XX1	X0X1
Vector 2	01X0	01XX	X01X	10X1	0X1X	1XX0	0X01
Vector 3	0XX0	10X1	10X0	XX01	00XX	10X0	01X1

Fig.3. Original test vector divided into several sub blocks

As shown in Fig. 3, there are 28 test data sets for three test vectors that are divided into seven blocks each. Since, there are only four data sets in block of data,we can have 16 possible combinations (viz., minterms) in each block. The frequency of a block is determined by analyzing the number of unspecified data in all the blocks. For example, the minterm 1001 is contained in block numbers 2, 4, and 7 in the test vector-1. The same minterm also appears in block number 4 of the test vector-2 and in block number 2 of the testvector-3, respectively. So, the frequency of 1001 in these sets of test vectors is determined as 5.

2.3. Preprocessing of High Frequency Data Blocks

The BWT algorithm is executed on all high frequency blocks of data. Burrows and Wheeler have released the details of a transformation function that opens the door to some revolutionary new data compression techniques.

	F						L
S4	B	B	S	D	R	D	O
S5	B	S	D	R	D	O	B
S2	D	O	B	B	S	D	R
S0	D	R	D	O	B	B	S
S3	O	B	B	S	D	R	D
S1	R	D	O	B	B	S	D
S6	S	D	R	D	O	B	B

Fig. 4. Original set of strings (***S* 0**) associated with the buffer

BWT converts a block of data into a format that is extremely well suited from the standpoint of data compression. The BWT is performed on an entire block of data, all at once. This transformation takes a block of data and rearranges them using a sorting algorithm known as lexicographical sorting. The detail of the sorting process is explained as follows using an example string DRDOBBS. The resulting output block contains exactly the same data elements that it started with, but differing only in their ordering.

	F						L
S4	B	B	S	D	R	D	O
S5	B	S	D	R	D	O	B
S2	D	O	B	B	S	D	R
S0	D	R	D	O	B	B	S
S3	O	B	B	S	D	R	D
S1	R	D	O	B	B	S	D
S6	S	D	R	D	O	B	B

Fig.5 Set of strings after sorting.

This transformation is a reversible process, meaning thereby that the original ordering of the data elements can be restored with no loss of fidelity. Also, a block of data transformed by BWT can be compressed using any or a combination of standard techniques.
the matrix is formed by rotating the original sequence of the string. Then, the matrix of Fig. 4is lexicographically sorted. After sorting, the set of strings is arranged, as shown in Fig. 5.

BWT OPERATION

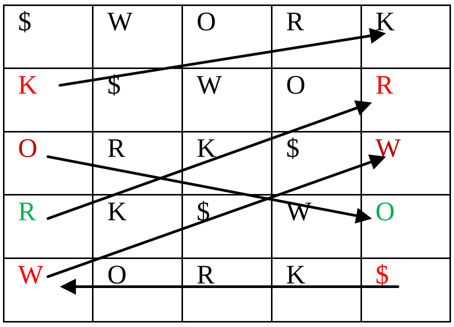
Basic operation of BWT .It is starting "WORK"

It is Block sorting and reversible operation method .It is rotation operation method.

$	W	O	R	K
K	$	W	O	R
R	K	$	W	O
O	R	K	$	W
W	O	R	K	$

Fig. 6. Original set of strings

It arrange the first column in ascending order.



$	W	O	R	K
K	$	W	O	R
O	R	K	$	W
R	K	$	W	O
W	O	R	K	$

Fig.7 Set of strings after sorting.

2.4.Patterns Overlapping Based Approaches

An illustrative example of an overlapping-based compression is shown in Figure 8. Here the non-compressed test length equals to the number of patterns multiplied by the number of CUT scan-chain cells, 10□5 = 50 bits in the example case. When properly overlapped, the compressed test length is only 16 bits. Note that by shifting the pattern only by one bit the overlap needs not be always achieved. Then two or more clock cycles (shifts) must be applied. Such a situation is in referred to as a presence of *link patterns*. They do not increase the fault coverage, but may increase the defect coverage.

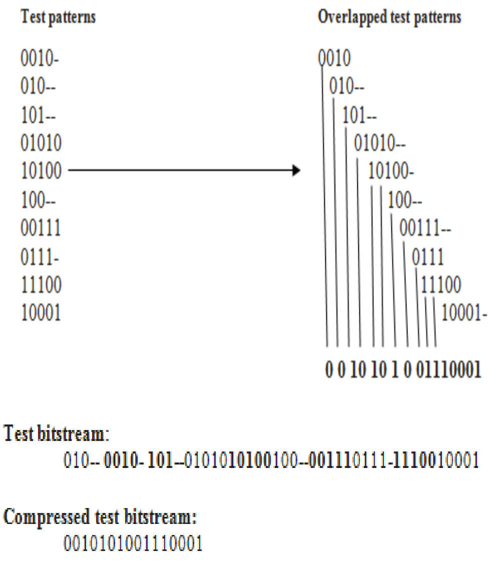


Fig 8. Patterns overlapping

Test don't cares are greatly beneficial for the compression, since they can be overlapped with any value. Thus, don'tcares bring more freedom into the overlapping process. The patterns may be *reordered*, in order to reach maximum compression (i.e.,maximum overlap).

III. SOFTWARE REQUIREMENTS

3.1. Modelsim se 6.3f

In this paper the Models SE 6.3f software is used for simulation and verification. Models is a verification and simulation tool for VHDL, Verilog, SystemVerilog, and mixed- language designs. ModelSim's architecture allows platform independent compile with the outstanding performance of native compiled code. The basic simulation flow is given by working libraries, Compile design files, Load and run simulation, Debug results. ModelSim offers numerous tools for debugging and analyzing the design. The project flow for Modelsim is given by Create a project, add files to the project, compile the project, run the simulation, Debug the results

3.2. Xilinx Ise Design Tools

Xilinx ISE is the design tool provided by Xilinx. Xilinx provides a free version of this tool called Web pack which would be virtually identical for our purposes. There are four fundamental steps in all digital logic design. These consist of:

1. Design – The schematic or code that describes the circuit.
2. Synthesis – The intermediate conversion of human readable circuit description to FPGA code (EDIF) format. It involves syntax checking and combining of all the separate design files into a single file.
3. Place & Route – Where the layout of the circuit is finalized. This is the translation of the EDIF into logic gates on the FPGA
4. Program – The FPGA is updated to reflect the design through the use of programming (.bit) files.

IV. RESULTS AND DISCUSSION

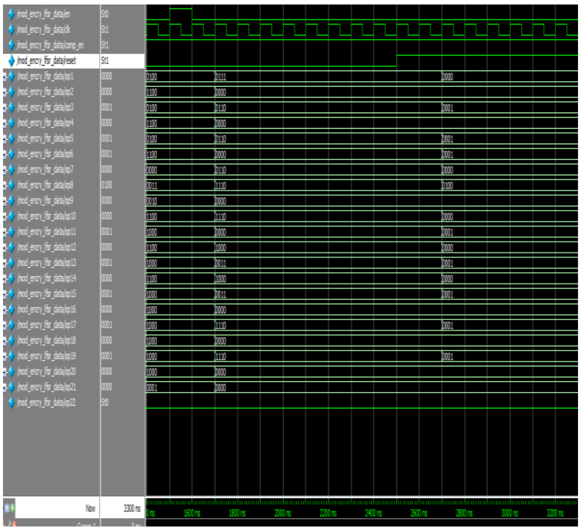
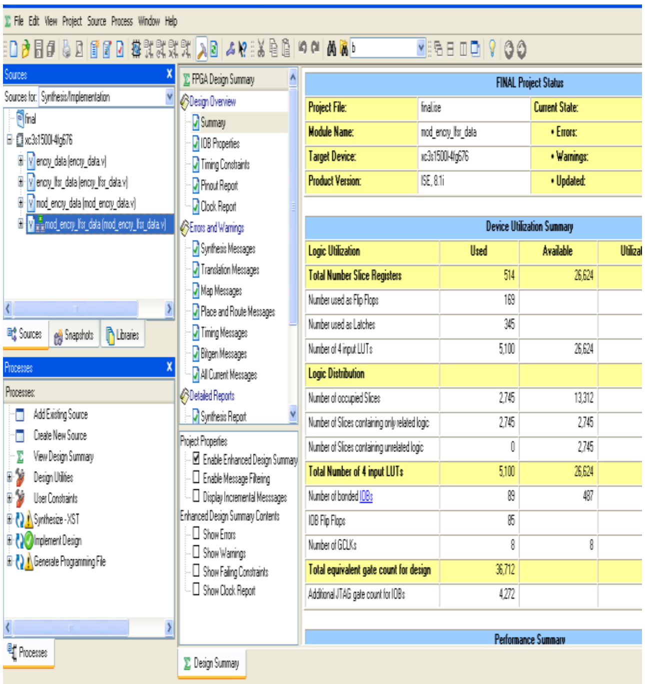


Fig.9 Result of Encode LFSR output



FINAL Project Status

Project File:	final.ise	Current State:
Module Name:	mod_encry_lfsr_data	• Errors:
Target Device:	xc3s1500l-4fg676	• Warnings:
Product Version:	ISE, 8.1i	• Updated:

Device Utilization Summary

Logic Utilization	Used	Available	Utilizat
Total Number Slice Registers	514	26,624	
Number used as Flip Flops	169		
Number used as Latches	345		
Number of 4 input LUTs	5,100	26,624	
Logic Distribution			
Number of occupied Slices	2,745	13,312	
Number of Slices containing only related logic	2,745	2,745	
Number of Slices containing unrelated logic	0	2,745	
Total Number of 4 input LUTs	5,100	26,624	
Number of bonded IOBs	89	487	
IOB Flip Flops	85		
Number of GCLKs	8	8	
Total equivalent gate count for design	36,712		
Additional JTAG gate count for IOBs	4,272		

Performance Summary

Fig .10 Design summary of test LFSR

V. CONCLUSIONS

This paper present a Modelsim simulated results of reconfigurable compression techniques .The hybrid Compression technique that targets the unique characteristics of the block matching test data compression, BWT along with several coding algorithms on test data sequences. The BWT reduces the number of transitions for the test vector sequences, reduce the delay and test data volume size. Although, BWT involves a very complex method of computation and takes a longer time for compression, the reverse process is very simple and quite faster. Pattern overlapping method create more matching pattern reducing time cost and size of data . It improves the efficiency of text compression algorithm. The hybrid based compression scheme that provides the high compression ratio and achieve the output with high accuracy. This technique achieved the high fault coverage. The activity of hybrid test data set is less than original test data.

REFERENCES

[1] M. Abramovici, C. Stroud, and M. Emmert, "Using embedded FPGAs for SoC yield improvement," in Proc. Des. Autom. Conf., 2002, pp. 713–724.

[2] K. Basu and P. Mishra, "Test data compression using efficient bitmask and dictionary selection methods," IEEE Trans. VLSI Syst., vol. 18, no. 9, pp. 1277–1286, Sep. 2010.

[3] S. Biswas and S. R. Das, "A software-based method for test vector compression in testing system-on-a-chip," in Proc. IEEE Instrumen. Meas. Technol. Conf., Apr. 2006, pp. 359–364.

[4] S. Biswas, S. R. Das, and E. M. Petriu, "Space compactor design in VLSI circuits based on graph theoretic concepts," IEEE Trans. Instrumen. Meas., vol. 55, no. 4, pp. 1106–1118, Aug. 2006.

[5] M. Burrows and D. J. Wheeler, "A block-sorting lossless data compression algorithm," Digit. Syst. Res. Center, Palo Alto, CA, USA, Tech. Rep. 124, 1994..

[6] A. El-Maleh, S. Al Zahir, and E. Khan, "Ageometric-primitives-based compression scheme for testing system-on-a-chip," in Proc. VLSI Test Symp., 2001, pp. 54–58.

[7] V. Groza, R. Abielmona, and M. H. Assaf, "A self-reconfigurable platform for built-in self-test applications," IEEE Trans. Instrumen. Meas., vol. 56, no. 4, pp. 1307–1315, Aug. 2007.

[8] M. Ishida, D. S. Ha, and T. Yamaguchi, "COMPACT: A hybrid method for compressing test data," in Proc. VLSI Test Symp., 1998, pp. 62–69.

[9] V. Iyengar, K. Chakraborty, and B. T. Murray, "Built-in self testing of sequential circuits using precomputed test sets," in Proc. VLSI Test Symp., 1998, pp. 418–423.

[10] A. Jas, J. G. Dastidar, and N. A. Touba, "Scan vector compres-sion/decompression using statistical coding," in Proc. VLSI Test Symp., 1999, pp. 114–120.

BIOGRAPHIES

First Author.

V.Vinothini received the B.E. degree from Latha mathavan Engineering College Madurai, India in 2012 and doing Masters of Engineering in VLSI Design at Sri Shakthi Institute of Engineering and Technology, Coimbatore, India .Her research interests include Body Genetic Algorithm, VLSI Design, and VLSItesting

Second Author

D **S.Yamuna** received the B.E. degree from Tamilnadu College of Engineering Coimbatore, India, in 2012, the M.E degree from Sri Shakthi Institute of Engineering and Technology, Coimbatore, India in 2014.She is currently an Assistant Professor with the Department of Electronics and Communication Engineering, Sri Shakthi Institute of Engineering and Technology, Coimbatore. Her research interests include functional verification, VLSI Design, VLSI testing, and Low Power of VLSI Circuits.