

# RUN-LENGTH BASED COMPRESSION FOR SELECTIVE UNSPECIFIED TEST PATTERN

<sup>1</sup>S. SARAVANAN, <sup>2</sup>A. BALASUBRAMANIYAN

<sup>1</sup>Assistant Professor, School of Computing, SASTRA UNIVERSITY, Thanjavur, India -613402

<sup>2</sup>M.Tech- VLSI Design, School of Computing, SASTRA UNIVERSITY, Thanjavur.

E-mail: [saran@core.sastra.edu](mailto:saran@core.sastra.edu) , [bala.balu123@gmail.com](mailto:bala.balu123@gmail.com)

## ABSTRACT

In all System-on-a-Chip (SoC) designs, there is a necessity to reduce the large test data volume and this is achieved by test data compression. One of the methods is the variable-to-variable length compression method. A selective run-length based compression which comes under variable-to-variable method is presented in this paper. The proposed work is based on threshold calculation on don't cares (X). So compression is not targeted for all the test patterns. Depending upon threshold value compression is identified. The test patterns having large number of don't-cares (X's) are selected for compression. Also the test vectors having less compression ratio is not considered for compression by calculating its threshold value. The selected test patterns can be divided into number of blocks containing equal number of test vectors. Each block is compared with the adjacent blocks bit-by-bit and it has to be merged. The number of blocks merged can be given in the control code. The experiments are conducted on ISCAS'89 benchmark circuits to know the effectiveness of the compression technique. The results show that the technique has a reasonable effect on compression.

**Keywords:** *Test Data Compression, Selective Run-Length Compression, Don't-Cares (X's), Threshold Calculation, Block Merging, Control Code.*

## 1. INTRODUCTION

Recent advances in the process technology make more and more functions which are crammed into a single device. In modern devices Intellectual Property (IP) cores and several modules are integrated on a single chip. Billions of transistors are fabricated on a single wafer. Although increasing integration of transistors on a single chip produces robust design, more defects are produced accordingly. In this situation there is a need to test those designs. As the technology advances, huge volume of test data is needed to be tested.

Although the modern System-on-a-Chip (SoC) incorporates many functions, they were affected by several test challenges. Huge test data volume is one of the major problems for the SoC vendors. Because the huge test data volume not only exceeds the commercial Automatic Test Equipment (ATE) memory and I/O channel capacity, it also increases the testing time and test power. Increase in test time, memory and power cause a direct impact on test cost and time

to market. So the vendors are in a situation to reduce the volume of test data.

Reduction of test data gathers the attention of all the SoC designers for the past several years. The methods which are used to reduce the amount of test data that is stored in the ATE memory are test vector compaction and test data compression. Many test data compression techniques have been proposed so far to reduce the test data volume and improve the transmission efficiency between the ATE and SoC. The compression technique is used to condense the test data and is stored in ATE memory. Through the test channels the compressed data are transferred to SoC. An on-chip decoder is used to retrieve the original test data without any loss and is scanned serially. Objective of this paper is to reduce test volume by identifying unspecified test patterns. This is achieved with help of calculating threshold value in run-length based compression.



## 2. PREVIOUS WORKS

All the test data compression techniques fall into any one of this category: Code-based schemes, Linear-decompression-based schemes and Broadcast-scan-based schemes. Code-based scheme compression involves partitioning the original test data into symbols and replaces it by a codeword according to its specific property to encode the data. In the decompression area the decoder simply replace the codeword by the specific symbols [1]. A variable-to-variable length GOLOMB coding is proposed [2] which give a low cost, very high compression with a scalable on-chip decoder. Another variable-to-variable length compression technique called frequency-directed run-length (FDR) codes [3] distributes the runs of 0's in the test sequence. Testing time and area overhead can be reduced by using a method called (VIHC) Variable-length Input Huffman coding [4].

The maximum run-length can be limited to tradeoff compression ratio by combining both run-length and Huffman coding called RL-HC [5]. A block merging technique is used to merge many consecutive test blocks to reduce the test data volume [6]. Both test data volume and dictionary volume are reduced by having smaller number of codeword for larger block size [7]. A new horizontal compression technique is used [8] for multiple cores. A new encoding technique with more flexible control code to attain high data compression is called scan slice encoding [9]. A compression technique is proposed which combines hamming Distance Based Reordering (HDR), Column wise Bit Stuffing (CBS) and Difference Vector (DV). The scheme pre-processes the test data before applying any other compression technique for giving better compression [10]. Reducing test data volume and test power by various techniques is discussed in [11 - 14].

## 3. PROPOSED SCHEME

Selective pattern compression is a run-length-based compression technique. In this method the encoder encodes the test set by runs of compatible pattern. The test patterns can be selected for compression from the original test data. The selected pattern must have large number of don't-care bits (X's). In this technique the selected pattern can be divided into number of bit-patterns. Each bit-pattern is compared with

the adjacent one and the number of runs can be denoted by a control code. The final codeword has control code followed by encoded pattern. The control code (C) denotes the number of pattern runs of the encoded pattern (E) and the merge of both is denoted by codeword.

### Selective Compression

In this method the test pattern has to be selected for doing compression. This is due to the fact that the compression is easy and possible if the don't-cares (X's) are higher in the test data. If the rate of known values (0's and 1's) is higher, it can't able to achieve good compression. So the compression technique can be applied only for those test vectors having larger number of X's. By calculating a threshold value (Th) it is possible to know the test vectors which are giving good compression. The threshold value is proportional to the rate of compression. The threshold value is calculated by using the equation (1) as,

$$\text{Threshold (Th)} = ((\text{total number of bits}) - (\text{known bits})) / (\text{total number of bits}) \dots (1)$$

For an example in Table 1, the input test patterns P1, P2 and P3 have the following threshold values (Th) as 0.43, 0.5 and 0.31. In the above example the threshold value is kept as Th = (0.4). So, there is no need to compress the test vectors having threshold value lesser than (0.4). So P3 can be neglected for achieving good compression. The patterns having the threshold value greater than (>0.4) is taken and it is partitioned into number of bit-patterns according to the control code. In Figure 1, the number of pattern runs is fixed to 4 (i.e., a 2-bit control code (C) is used to denote the number of pattern runs as "00", "01", "10", "11"). Here only 4-bit patterns are used; where the length of the bit-pattern is varied according to the length of the test data.

For example, in Figure 1, the number of patterns is fixed to 4 by using a 2-bit control code(C). As shown in figure, the 16-bit test pattern (P1) "10XXXX011001X0XX" can be divided into 4 blocks each with 4-bit bit-patterns. Now take two bit-patterns from left and compare it bit-by-bit. The comparison result shows the possibility of merging. If possible, merge those bit-patterns and the merged data is compared with third bit-pattern. It continues until the last

pattern. For each and every merge the C is incremented. Initially the C is “00”. While merging most of the “X” can be replaced by either “0” or “1”.

After merging all the bit-patterns, the final encoded pattern for above example is “111001”. Where the first 2 bits “11”from left is the C, which denotes the number of blocks merged (pattern runs) and the remaining 4 bits are the

encoded pattern of the original test data. If the possibility for merging two bit-pattern is false, then a codeword has to be generated before that pattern and again a new process of merging is done from the next pattern. Finally both the codeword has to be placed is shown in Table 1 at P2.

Table 1- Proposed Selective Compression

C = Control Code, E= Encoded Pattern

Codeword = C + E.

Patterns No	Input test patterns	Threshold (Th)	C E	C E	C E	C E
P1	10XXXX011001X0XX	0.43				11 1001
P2	1XXX110X10XXXX01	0.50			01 110X	01 1001
P3	10XX0XX1XX10XXX1	0.31	00 10XX	00 0XX1	00 XX10	00 XXX1

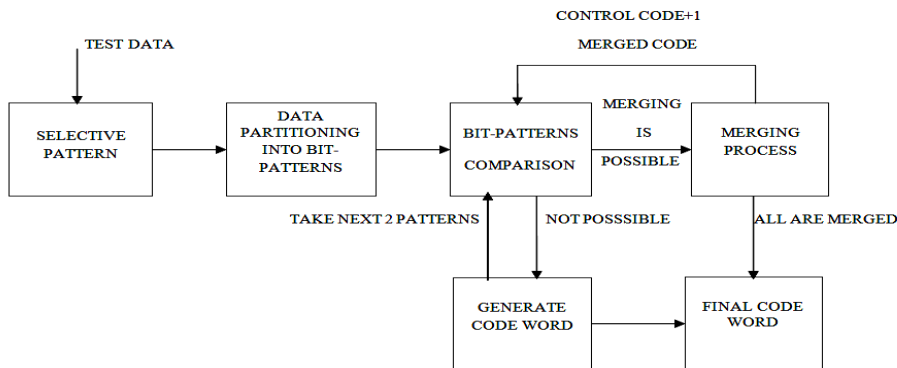


Figure 1- Proposed Block Diagram

**Block Diagram**

Figure 1, shows the block diagram of selective pattern compression. The block diagram shows the flow of the compression process. First the test pattern is selected from test data for doing compression by using the threshold value (Th). Then the test pattern is partitioned into number of bit-patterns of specified length. First two bit-patterns has to be taken and have to compare bit-by-bit. Depending upon the possibility of merging, successive bit-patterns has to be merged with the previous outputs and final codeword has to be generated. If there is no

possibility of merging two bit strings, a codeword has to be generated for the previous outputs and new process of merging has to be started from the next string.

**State Diagram**

Figure 2, shows the state diagram of selective pattern compression. When RESET=‘1’, the system will be in state 0, which initialize the system. When RESET= ‘0’, the test data is partitioned into bit-patterns in state 1. Again the system goes to state 0, when condition (COND) = ‘1’. If COND = ‘0’, it goes to state 2 where comparison of bit-patterns are done to check the

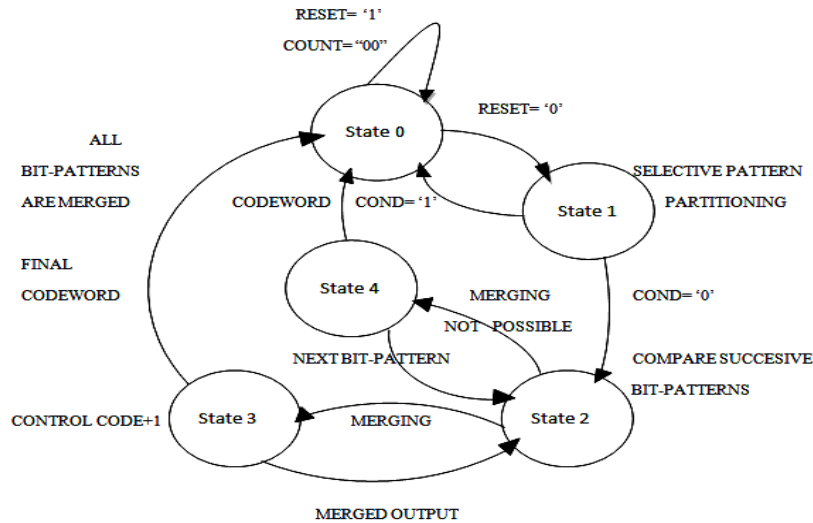
possibility of merging two bit-patterns. If possible goes to state 3 and if not goes to state 4.

At state 3, the two bit-patterns are merged into one with the increment of control code and the merged output going to state 2 for comparison with the next bit-pattern for merging. The process continues till the last bit-pattern was merged. When all the bit-pattern was merged, the present control code (C) with the encoded data (merged pattern) gives the codeword. The final codeword is the compressed data. At state 4, a codeword for the present pattern continues its process from the state 2 by taking next two bit-patterns for merging. Finally the process resets when all strings were merged.

#### 4. EXPERIMENTAL RESULTS

The experiments are done on 4 large ISCAS'89 benchmark circuits generated by Mintest ATPG (Automatic Test Pattern Generator). Here the test patterns are divided into 8-Blocks. The compression ratio (CR) for each and every method can be calculated by the equation (2) as,  

$$CR\% = \frac{((\text{Total number of test data}) - (\text{Compressed data}))}{(\text{Total number of test data})} \times 100\% \dots (2)$$



Compression Ratio (CR)								
	Existing Systems					Proposed System		
Circuits	GOLOMB	FDR	VIHC	RL-HC	BM	6-BLOCK	5-BLOCK	4-BLOCK
s5378	37.11	47.98	51.52	53.75	54.98	45.28	48.43	53.13
s9273	42.25	43.61	54.84	47.59	51.19	44.39	53.27	59.16
s13207	79.74	81.3	83.21	82.51	84.89	71.88	72.85	73.09
s15850	62.82	66.21	60.68	67.34	69.49	68.48	72.41	75.75

Figure 2- State Diagram

Table 2- Compression Ratio of the Proposed Method

In Table 2, the compression ratio of the proposed method is compared with various techniques. In 6-Block technique the system will have at most 6-Blocks out of 8-Blocks after compression. This means that the system can be able to compress only 2-blocks at minimum. If it comes under one or none, it can be eliminated

from compression by calculating its Th value. This is called selective compression. In the same way 5-Blocks and 4-Blocks are calculated.

Figure 3 shows the sample simulation waveform of the selective compression. The selected region in the waveform gives the

compressed output. The input taken has 16-bit length with Th value of (0.43) and its compressed output has 6-bit length. Figure 4 shows the comparison of 6, 5, 4-Block techniques. It shows that the 4-Block technique has higher compression ratio than others. Also Figure 5 analyze compression ratio of various techniques. Mostly the proposed technique achieved better compression than many techniques. If less number of unspecified test pattern is used then this proposed technique will limit the rate of compression.

### 5. CONCLUSION

A selective pattern run-length code is proposed in this paper. Test pattern with larger number of

unknown values can be able to compress efficiently. To identify more compression, proper threshold value is observed. This will limit the proposed method for achieving higher compression ratio. Thus more number of known values is not considered for compression due to its larger amount of code words. So these test patterns can be eliminated for compression. These leads to allow only selected test patterns for doing compression. This method can be experimented on larger ISCAS'89 benchmark circuits. Proposed results show more compression was achieved by 4-block technique about 6% to 9% than the existing method. It can be further extended with proper new technique to reduce the length of code words for achieving more compression.

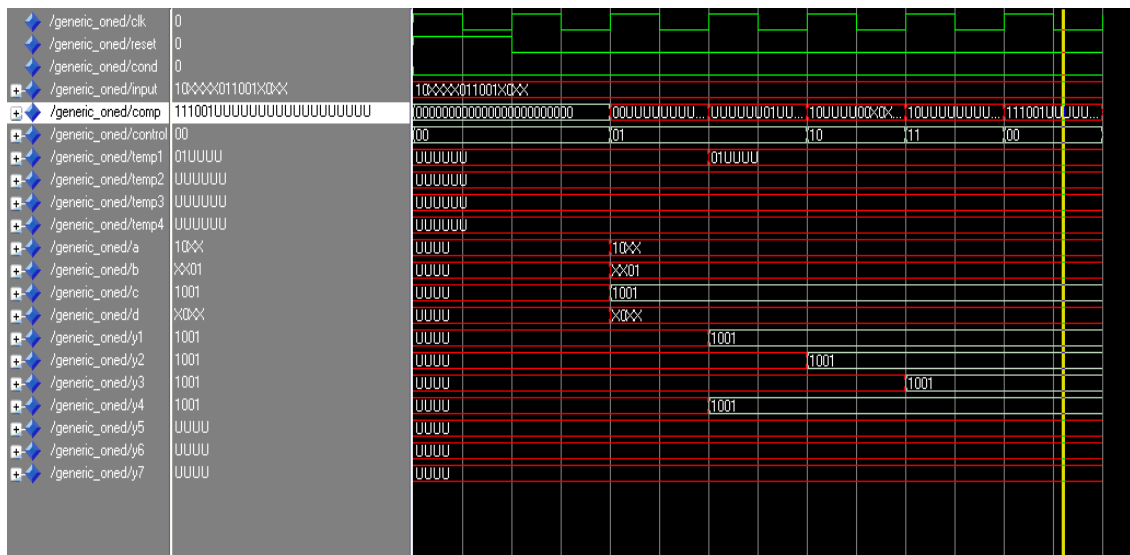


Figure 3- Simulation Waveform of Proposed Method

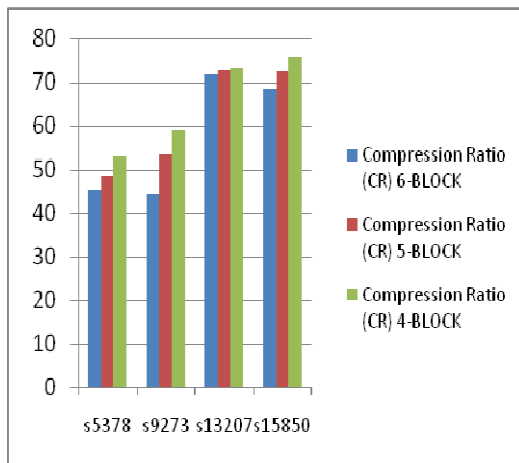


Figure 4- Comparison of 6, 5, 4-Blocks

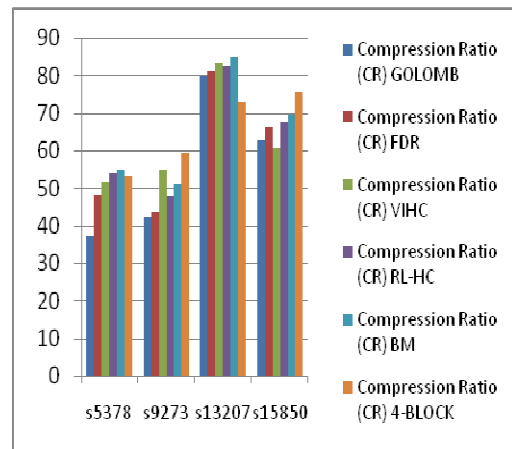


Figure 5- Comparison of various compression techniques



## REFERENCES

- Conference, PRIME 2007, IEEE Transaction.*
- [1] Toubna NA, "Survey of Test Vector Compression Techniques", *IEEE Design and Test of Computers.*, 2006, 23(4):294–303.
- [2] Chandra A, Chakrabarty K, "System-on-a-chip Data Compression and Decompression Architecture Based on Golomb Codes", *IEEE Transactions Computer-Aided Design of Integrated Circuits & System.*, 2001, 20(3):355–368.
- [3] Chandra A, Chakrabarty K, "Test Data Compression and Test Resource Partitioning for System-on-a-chip using Frequency-directed run-length (FDR) codes", *IEEE Transactions on Computer.*, 2003, Vol-52 (8), pp:1076–1088.
- [4] Gonciari PT, Al-Hashimi B and Nicolici N, "Improving Compression Ratio, Area Overhead, and Test Application Time for System-on-a-chip Test Data Compression/Decompression", *In Proc design automation test in Europe, Paris.*, 2002, pp:604–611.
- [5] Nourani M and Tehranipour M, "RL-Huffman Encoding for Test Compression and Power Reduction in Scan Application", *ACM Transactions on Design Automation of Electronic Systems (TODAES).*, 2005, Vol-10(1), pp:91–115.
- [6] El-Maleh AH, "An Efficient Test Vector Compression Technique Based on Block Merging", *IET Computer Digit Tech.*, 2008, 2(5):327–335.
- [7] Terumine Hayashi, Naohiro Hiraiwa, Tsuyoshi Shinogi, Haruhiko Takase, and Hidehiko Kita, "Test Modification and Compression Technique for Reducing Total Test Volume with Dictionary Data", *ASICON 2005, 6<sup>th</sup> International Conference, IEEE Transaction.*
- [8] Julien Dalmosso, Marie-Lise Flotters and Bruno Rouzeyre, "Systems-on-Chip: Use of Test Data Compression Technique for Reducing Test Time"., *Research in Microelectronics and Electronics*
- [9] Keun-Soo Lee, Hyuntae Park, Hyeonuk Son and Sungho Kang, "A New Scan Slice Encoding Scheme with Flexible Code for Test Data Compression", *SoC Design Conference (ISOCC), 2010 International Conference, IEEE Transaction.*
- [10] Usha S. Mehta, Kankar S. Dasgupta and Niranjana M. Devashrayee, "Hamming Distance Based Reordering and Columnwise Bit Stuffing with Difference Vector: A Better Scheme for Test Data Compression with Run Length Based Codes", *VLSI Design, 2010. VLSID'10, 23<sup>rd</sup> International Conference.*
- [11] S. Saravanan and Har Narayan Upadhyay, "Transition Vector Reduction using Segmentation method based on Compression Technique", *Australian Journal of Basic and Applied Sciences*, 5(9): 2147-2151, 2011.
- [12] S. Saravanan, P. Selvakumar, A. Balasubramaniyan and R. Silambamuthan, "Achieving Higher Test Data Compression using Pattern ", *IEEE International Conference on Computational Intelligence and Computing Research, ICCIC-2011.*
- [13] Chia-Yi Lin and Hung-Ming Chen, "A Selective Pattern –Compression Scheme for Power and Test-data Reduction", *Computer-Aided Design (ICCAD), IEEE/ACM International Conference, 2007.*
- [14] Chia-Yi Lin, Hsiu-Chuan Lin and Hung-Ming Chen, "On Reducing Test Power and Test Volume by Selective Pattern Compression Schemes", *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 18(8) : 1220-1224, 2010.