IOSR Journal of VLSI and Signal Processing (IOSR-JVSP)
Volume 4, Issue 2, Ver. I (Mar-Apr. 2014), PP 91-95
e-ISSN: 2319 – 4200, p-ISSN No. : 2319 – 4197
www.iosrjournals.org

Test Data Compression Using Variable Prefix Run Length (VPRL) Code

K.S.Ganesh kumar[1], Karen Thangam Jacob[2], B.Manjurathi[3]
[1,2,3]Department of ECE, Karunya University, Coimbatore, Tamil nadu, India.

Abstract: *One of the major challenges in testing a system-on-a-chip (SoC) is dealing with the large test data volume and large scan power consumption. To reduce the volume of test data, several test data compression techniques have been proposed. This paper presents a new test data compression scheme, which reduces test data volume for a system-on-a-chip (SoC). The proposed approach is based on the use of MT (Minimum Transition)-fill technique and Variable Prefix Run Length (VPRL) codes for test data compression. These VPRL codes can efficiently compress the data streams, that are composed of both runs of 0s and 1s. Experimental results for ISCAS'89 benchmark circuits supports and proves the proposed approach, better to the other existing techniques, by reducing test data volume.*
Keywords: *System on a chip, MT-Fill, AVR code, Test data compression, Compression ratio.*

I. INTRODUCTION

Advances in VLSI technology have resulted in a change in the design paradigm where complete systems containing millions of transistors are integrated on a single chip. As systems grow in size and complexity, the volume of test data and test scan power also increases rapidly. Power dissipation during a test is also a significant problem, as the size and complexity of system-on-a-chip (SoC) continues to grow [1]. The power dissipation can increase by a factor of 2-3 compared with the functional mode of operation [2]. The Dynamic power dissipation plays a major role in overall test scan power. The Switching activity during test has a large contribution in dynamic power and thereby overall scan power. In stored pattern testing, the test patterns need to be transferred from the automatic test equipment (ATE) to the SOC and then applied to the cores that are being tested[3]. The responses of the cores are compacted and sent back to the ATE for signature analysis. Large test data not only increases the testing time but may also exceed the tester memory capacity [3-7]. The system integrator was also confronted with the challenges of test data volume, power consumption during the test. There are many techniques to control the volume of test data, test application time and power consumption in test mode. Test data volume and power reduction can be achieved by utilizing Built-in-Self-test (BIST) [2,3], test data compaction or test data compression techniques [2-18, 20-22]. However, BIST requires a longer test application time. It is extensively used for memory testing but is not common for logic testing [3]. Although test compaction techniques reduce the test application time, the compacted test sets might achieve less detection of non-modeled physical defects. Hence to reduce the test data volume test data compression is considered to be the best alternative. The objective of test data compression is to reduce the number of bits needed to represent the test data. There exists many test data compression techniques like Broadcast scan-based, Linear decompression based, and Code-based techniques [8,9,19]. Code-based test data compression scheme is more appropriate for larger devices. Some of the code-based test data compression schemes are Dictionary codes, Statistical codes, Constructive codes, and Run length-based codes are used for test data compression [8,9,19]. Among these, run length- based codes are used to encode the repeatedly occurring values and is an efficient method for test data compression. The don't care bit filling methods [10] and test vector reordering [11] further enhances the test data compression. Thus, combining run length- based codes with the Bit filling [10] & reordering techniques [11], not only reduces the switching activity, but becomes efficient in increasing the test data compression.

The rest of the paper is organized as follows: Section 2 presents the different types of test data compression techniques and its effect. Section 3 presents the proposed design algorithm steps. Section 4 presents the decompression architecture. Section 5 presents the power consumption analysis. Experimental results are shown in Section 6 and Conclusions are presented in Section 7.

II. RELATED WORKS

We can Classify the Existing works as Data dependent and Data independent. In the data independent type the decoder or decompression program is suitable for all type of test sets, i.,e it is reusable for any kind of test sets. The Golomb [3,7], FDR [4], EFDR [5,6], ALT-FDR [14], 9C [13], BM [12,15], CEBM [16], ERLC [17] methods belongs to data independent. In the data dependent type the decoder or decompression program is only suitable for specific test sets. The bit filling methods are effective for reducing switching activity and thereby the power. The different types of bit filling methods and their effective results are discussed in [3]. The Golomb [3,7], FDR [4], EFDR [5,6], ALT-FDR [14], CEBM [16], ERLC [17] methods compress the test vectors based on the run-length. The methods Golomb [3,7] and FDR [4] are best suitable for the test vectors having more number of 0 runs, because these methods can compress only the 0 run-lengths in the test set. The EFDR [5,6] code compresses both runs of 0s and 1s, the first bit in the codeword specifying the type of run-lengths. The ALT-FDR [14] code was proposed for both runs of 0s and 1s, wherein a binary parameter, α was used to indicate type of run-length. Here, test vectors starting with both zero run-length and one run-length are suitable for this method. The Block merging (BM) [12,15] technique was proposed to merge the compatible blocks of fixed length, in the test set. The disadvantage being that it is only suitable for the test vectors having more number of don’t care bits. Using this technique, high compression was achieved by encoding runs of fixed length blocks, in which only the merged block and number of blocks merged, are recorded. The Nine coded (9C) [13] compression technique uses exactly nine code words for compression process and is flexible in using both fixed and variable length blocks. The ERLC [17] technique was proposed to encode the test set having both run of 0’s and 1’s and in this method the repeated codeword is used to denote the other three bit codeword. Its decoder architecture adds one more component, the bit register, to store the repeated run length. The same type of run length in the test sets is difficult to achieve and hence not found to be efficient. The pre-processing techniques hamming distance based reordering (HDR), column wise bit stuffing (CBS), and difference vector (DV) are shown in [11].

Don’t care bit filling is used to fill the X bit in the test patterns. There are different bit filling techniques used to fill the X bit. 0-fill, 1-fill and random fill are the techniques used previously. Later MT- fill and column wise bit filling [10, 11] is used to fill the X bit which reduces the transition. The genetic algorithm based heuristic to fill the don’t cares is proposed in [21]. This approach produces an average percentage improvement in leakage power as well as dynamic power over 0-fill, 1-fill and MT- fill.

III. PROPOSED METHOD

The Proposed technique is the combination of Bit filling and encoding for the original test patterns. The proposed technique flow is shown in figure 1.

Original test vectors

↓

Minimum Transition (MT) bit filling

↓

Variable Prefix Run-length (VPRL) codes technique

↓

Compression Ratio calculation for the proposed method

Figure 1: Design Flow for the Proposed Technique

3.1 Minimum Transition (MT)-Fill

A series of X entries in the test vector are filled with the same value as the first non-X entry on the right side of this series [10]. This minimizes the number of transition in the test vector when it is scanned in. For example consider the test vector: 100**XX**011**X**11. This vector, after MT-fill [10] , would become 100**00**011**1**11. If the test vector has a string of X bits that is not terminated by a non-X bit on the right side, then it should be filled by the bit value to the left of the sequence. For example: XX00011**XX** should be 0000011**11** after MT-fill.

3.2 Variable Prefix Run-length (VPRL) Code

The Variable Prefix Run-length (VPRL) code is a variable-to-variable-length code and it consists of two parts – the group prefix and tail. The group prefix suggests the group to which the run length of either 0’s or 1’s, belong, while the tail further points out to the specific member of the group. The highlight of this technique holds in the simplicity of the coding table, wherein two group prefix are included in each group, the second prefix being the inverted data of 1^{st} prefix in each group [2]. The first prefix has m 0s ending with 1. Here, the run length is specified by runs of 0s terminating with a 1, or runs of 1s terminating with a 0. For example, 111110 and 110 are the runs of 1s and 000001 and 001 are the runs of 0s. The runs of 1s and 0s are put into groups A_1, A_2, A_3,……….., A_m, where k is the longest run length L_{max} in the test set. The m^{th} group includes 2^{m+1} kinds of run length. In other words, group A_1 consists of 4 kinds of run length “1, 2, 3, 4”. Group A_m is calculated with the following equation [2]:

$$m = [\log_2(L+4)-2] \qquad (1)$$

Table 1: Variable Prefix Run length (VPRL) code

Group	Run length	Group prefix	Tail	Code Word
A_1	1	01	0	01 0
	2		1	01 1
	3	10	0	10 0
	4		1	10 1
A_2	5	001	00	001 00
	6		01	001 01
	7		10	001 10
	8		11	001 11
	9	110	00	110 00
	10		01	110 01
	11		10	110 10
	12		11	110 11
A_3	13	0001	000	0001 000
	14		001	0001 001
	……..		……..	……….
	18		101	0001 101
	19		110	0001 110
	20		111	0001 111
	21	1110	000	1110 000
	22		001	1110 001
	………..		………..	……………
	26		101	1110 101
	27		110	1110 110
	28		111	1110 111

Original data (T_D)= 01 0001 110 11110
Run length 1 3 2 4
Encoded data(T_E)= 010 100 011 101
α=0 α=0 α=1 α=1

Figure 2: Example of encoding method of VPRL code.

Unlike FDR, which needs a two-bit codeword "00" for each '1' in the run-length of 1s, in VPRL both runs of 0s and 1s are encoded using the same codeword, with an additional binary parameter *α,* used to indicate runs of 0's and 1's, as was used in the ALT-FDR[14]. Still, a higher compression ratio is obtained for variable prefix run-length code, compared to Alternating FDR (ALT-FDR) [14] code and other improved codes, as the length of many code words are comparatively smaller in these codes, helping to achieve an improved compression ratio. For example, for runs of 0s "00000001", the VPRL codeword is "00110", while the codeword is "110001" for the FDR code [4] and the ALT-FDR code [14], and "0110000" is the codeword for the EFDR code [5,6].Hence, the data volume will decrease much, using VPRL coding , an alternative to the other codes.

IV. EXPERIMENTAL RESULTS

In this section, the proposed test data compression/decompression method for the ISCAS' 89 benchmark circuits was experimentally evaluated. The ISCAS'89 benchmark circuits test sets are obtained from the Mintest ATPG program.

Table 2 shows the test data volume of the benchmark circuits taken into consideration, and gives the analysis of the Total number of runs of 0s and 1s, in each of these test sets. Here, α indicates the total run-length.

The test set was generated by Mintest and its don't care bits are mapped according to the Minimum transition fill technique, aiding higher compression and lower power consumption, later.

The Compression Ratio (C.R) was computed as follows:

$$C.R = \frac{Original\ bits - Compressed\ bits}{Original\ bits} * 100\% \qquad (2)$$

Table 3 presents the experimental results of the Encoded bits (T_E) and Compression ratio using the VPRL code, over the original data.

Table 4 and Figure 3 projects the Comparison of Compression ratios of proposed method with other test independent Compression techniques, such as the Golomb [3], FDR [4], ALT-FDR [14], EFDR [5], AVR [2], MAVR [22]. It shows that proposed method yields the best Compression ratio compared to other codes.

Table 2: Analysis of Total number of runs in a test set

ISCAS'89 Circuit	Total no of Patterns	Bits per pattern	Size of Original data(T_D) (bits)	Total Runs (no of 0 runs + no of 1 runs) (α)
S5378	111	214	23,754	2289
S9234	159	247	39,273	4138
S13207	236	700	1,65,200	4722
S15850	126	611	76,986	4695
S38417	99	1664	1,64,736	11055
S38584	136	1464	1,99,104	13653

Table 3: Compression Ratio of Proposed Method

ISCAS'89 Circuit	Size of Original data (T_D) (bits)	Size of Encoded data (T_E) (bits)	Proposed Method
S5378	23,754	9879	**58.41**
S9234	39,273	14310	**63.56**
S13207	1,65,200	16477	**90.02**
S15850	76,986	16421	**78.67**
S38417	1,64,736	43327	**73.69**
S38584	1,99,104	46779	**76.50**

Table 4: Comparison of compression ratios with other techniques over original data

ISCAS'89 Circuit	Golomb (%)	FDR (%)	ALT-FDR (%)	EFDR (%)	AVR (%)	MAVR (%)	Propose method
S5378	37.11	48.02	50.77	53.67	61.13	62.31	**58.41**
S9234	45.25	43.59	44.96	48.66	58.93	59.93	**63.56**
S13207	79.74	81.30	80.23	82.49	85.54	86.44	**90.02**
S15850	62.82	66.22	65.83	68.66	74.82	77.43	**78.67**
S38417	28.37	43.26	60.55	62.02	68.63	69.15	**73.69**
S38584	57.17	60.91	61.13	64.28	71.30	71.90	**76.50**
Average	44.35	57.22	60.58	63.30	70.06	71.19	**73.47**

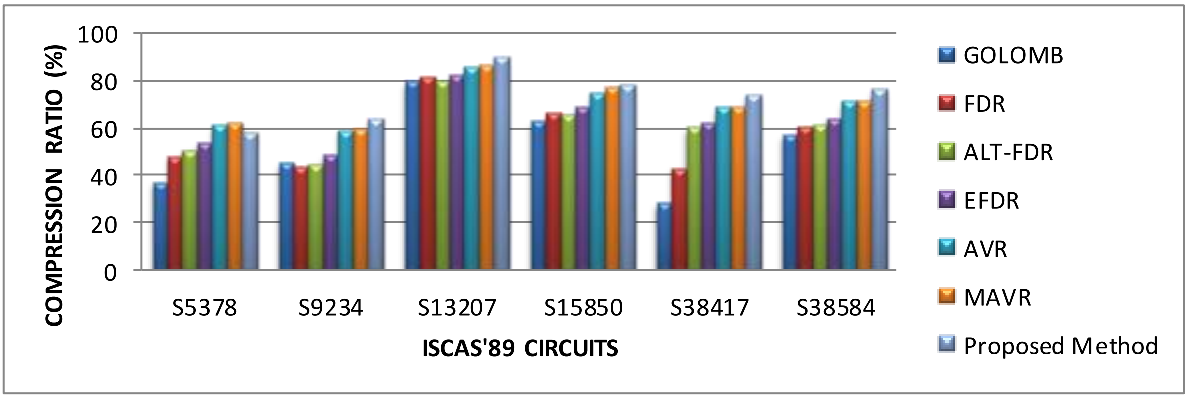

Figure 3: Comparison of compression ratios with other techniques over original data

V. CONCLUSION

Test data Compression is necessary due to the high cost of ATE memory, and the enormous amount of test volume and power consumption. The Proposed method combines Minimum Transition (MT)-fill and VPRL code, an efficient test data compression method, which reduces SoC test data volume. The MT-fill techniques, are used for increasing the run lengths of 0s and 1s,with also particularly reducing the switching activity which in turn reduces the average and peak power. Experimental results for the ISCAS'89 benchmark circuits shows that the proposed approach decreased test data volume with an average Compression ratio of 73.47% over the Original data.

REFERENCES

[1] J. Lee, N.A Touba, LFSR-reseeding scheme achieving low-power dissipation during test, IEEE Trans. On Computer-Aided Des. Integr. Circuits and System, Vol. 26, no. 2, 2007, pp. 396-401.

[2] Bo Ye, Qian Zhao, Duo Zhou, Xiaohua Wang, Min Luo, Test data compression using alternating variable run-length code, Elsevier, Integration, the VLSI journal, Vol.44, 2011, pp. 103-110.

[3] A. Chandra, K. Chakrabarty, System-on-a-chip Test Data Compression and Decompression Architecture based on Golomb Codes, IEEE Trans. On Computer-Aided Des. Integr. Circuits and System, Vol. 20, no. 3, 2001, pp. 355-368.

[4] A. Chandra, K.Chakrabarty, Test Data Compression and Test Resource Partitioning for System-oc-a-Chip using Frequency-Directed Run-length (FDR) Codes, IEEE Trans. Computers, Vol. 52, no. 8, Aug. 2003, pp. 1076-1088.

[5] A. H. El-Maleh, Test Data Compression for System-on-a-chip using Extended Frequency-Directed Run-Length Code, IET Computer digital Tech., Vol.2, no.3, 2008, pp. 155-163.

[6] M. Y. Wan, Y. Ding, Y. Pan, S. Zhou, X. L. Yan, Test Data Compression using Extended Frequency-Directed Run Length Code based on Compatibility, Electronics letters, Vol. 46, no. 6, Mar 2010, pp. 404-405.

[7] Anshuman Chandra, Krishnendu Chakrabarty, Low-Power Scan Testing and Test Data Compression for System-on-a-Chip, IEEE Trans. On Computer-Aided Des. Integr. Circuits and Systems, Vol. 21, no. 5, May 2002, pp. 597-604.

[8] Nur A. Touba, Survey of Test Vector Compression Techniques, IEEE Design & Test of computers, July 2006, pp.294-303.

[9] Usha S. Mehta, Kankar S.Dasgupta, and Niranjan M. Devashrayee, Run-Length-Based Test Data Compression Techniques: How Far from Entropy and Power Bounds?- A Survey, Hindawi Publishing Corporation, VLSI Design, Volume 2010.

[10] K. A. Bhavsar, and U.S. Mehta, Analysis of Don't Care Bit Filling Techniques for Optimization of Compression and Scan Power, International Journal of Computer Applications, Vol. 18,Mar 2011, pp. 30-34.

[11] Usha S. Mehta, Kankar S. Dasupta, Niranjan 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, In: 23rd International Conference on VLSI Design, 2010. pp. 33-38.

[12] A. H. El-Maleh, An Efficient Test Vector Compression Technique Based on Block Merging, IET Computer Digital Tech., Vol. 2, no. 5, 2008, pp. 327-335.

[13] M. Tehranipoor, M. Nourani, K. Chakrabarty, Nine-Coded Compression Technique foe Testing Embedded Cores in SoCs, IEEE Trans. On Very Large Scale Integration(VLSI) syst., Vol. 13, no. 6, 2005, pp. 719-731.

[14] A. Chandra, K. Chakrabarty, A Unified Approach to Reduce SoC Test Data Volume, Scan Power and Testing Time, IEEE Trans. On Computer-Aided Design of Integr. Circuits and systems, Vol. 22, no. 3, Mar 2003, pp. 352-362.

[15] Tie-Bin Wu, Heng-Zhu Liu, Peng-Xia Liu, Efficient Test Compression Technique for SoC Based on Block Merging and Eight Coding, J Electron Test, Vol. 29, no. 6, Dec 2013, pp. 849-859.

[16] Wenfa Zhan, Huaguo Liang, Cuiyan Jiang, Zhengfeng Huang, Aiman El-Maleh, A Scheme of test data compression based on coding of even bits marking and selective output inversion, Computers and Electrical Engineering, Vol. 36, no. 5, Sep.2010, pp. 969-977.

[17] Wenfa Zhan, and Aiman El-Maleh, A new scheme of test data compression based on equal-run-length coding (ERLC), Integration, the VLSI Jouranal, Vol. 45, no. 1, Jan. 2012 pp. 91-98.

[18] Jia Li, Xiao Liu, Yubin Zhang, Yu Hu, Xiaowei Li, and Qiang Xu, On Capture Power-Aware Test Data Compression for Scan-Based Testing, Proceedings of the International Conference on Comput.-Aided Designs, 2008, pp. 67-72.

[19] Zainalabedin Navabi. (2011). Digital System Test and Testable Design Using VHDL Models and Architectures [online]. Available: http://www.springer.com

[20] R. Sankaralingam, R.R Oruganti, N.A Touba, Static compaction techniques to control scan vector power dissipation, Proc. IEEE VLSI Test Sump., 2000, pp. 35-40.

[21] S. Kundu and S. Chattopadhyay, Efficient don't care filling for power reduction during testing, Proceedings of IEEE International Conference On Advances In Recent Technologies In Communication And Computing, 2009, pp. 319–323.

[22] P.R. Sruthi and M.Nirmala devi, A Modified Scheme for Simultaneous Reduction of Test Data Volume and Testing Power, International Symposium, VDAT 2012, 2012, pp.198-208.