

Test Data Compression and Power Reduction Using Similarity Based Reordering Technique for Wireless Systems

R. HariKumar¹ · B. Manjurathi²

Published online: 29 January 2016
© Springer Science+Business Media New York 2016

Abstract System-on-Chip is the major challenge for both design and testing engineers due to its increase in power consumption. The system consumes more power in test mode than the normal mode due to the switching activity that takes place between test data. Similarly, in test mode the volume of test data used is very high when compared to normal mode. This paper presents a novel test data compression approach which simultaneously reduces test volume, test power consumption with less decoder complexity. A test data compression when combined with efficient preprocessing such as filling and reordering of test patterns leads to efficient compression and power reduction. These are some preprocessing steps prior to the compression coding, affects the power. In this paper, the preprocessing steps proposed are column wise bit filling; Similarity based test vector reordering, difference vector techniques. This approach with Modified Golomb coding is proposed to improve the compression as well as to reduce the scan in peak power and average power. This approach efficiently decreases the switching activity and also very simple method to combine with run length based coding. The Power reduction is very efficient by this proposed method compared to other preprocessing methods. A careful analysis of bit filling is proposed that leads to significant reduction in peak and average power. Similarity based test vector reordering technique is also proposed to reorder the test data for reducing the switching activity that takes place in test data and reduces the power consumption. The difference vector is applied which leads to increase in the run length of 0s. The modified Golomb algorithm proposed next can efficiently compress the test set that composed of both 0s and 1s. The decompression architecture is also presented in the paper. Experimental results on ISCAS'89 benchmark circuits show that the proposed approach efficiently reduces the volume of test data and the peak, average power consumption.

✉ B. Manjurathi
manjurathibece@gmail.com; manjurathibose@gmail.com

¹ Department of Electronics and Communication Engineering, Bannari Amman Institute of Technology, Sathyamangalam, India

² Department of Electronics and Communication Engineering, Karunya University, Coimbatore 641114, India

Keywords Test data compression · Similarity based test vector reordering · Don't care fill · Difference vector · Modified Golomb Code · Test power reduction

1 Introduction

Advances in VLSI technology [1] have been proposed the design paradigm where complete systems containing millions of transistors are integrated on a single chip. As the complexity of system on a chip continues to increase, the difficulty and cost of testing such chips is increasing rapidly. Testing system on a chip is dealing with the large size of test volume that is stored in the tester memory and transferred between tester and circuit under test. The test power plays a major role because the switching activity during test has a large contribution in overall test power. Test-data compression [2] offers a promising solution to the problem of reducing the test data volume for SOCs and to reduce the test time, peak power and average power. In this approach set [3], a pre-computed test set for an IP core is compressed to a much smaller test, which is stored in ATE memory. Depending upon the test data that is used, data compression is divided into three/categories stated by [4]. They are code based schemes, linear-decompression based schemes and Broadcast scan based schemes. Considering the suitability of core based system on chip, code based scheme is more appropriate. Code-based scheme is widely used [4] and it transfers the specified bit-strings in test data into a segment of codeword. This performs dividing the original data into groups and each group is replaced with a codeword to produce the compression data. The compressed data is then used for testing to avoid the difficulties. Based on the schemes and variants, the code based compression methods are divided into four different categories: Statistical codes, Run length codes, dictionary codes, and constructive codes. The drawback of using dictionary code is that if the dictionary size is large, it will result in too much overhead of the decoder. The constructive coding and statistical coding also result in complex and data dependent decoder. Hence Run length based coding is selected for this research work which deals with less decoder complexity compared to other code based methods. This also leads to increase the compression ratio and to reduce the test power consumption by reducing the switching activity. The run length based test data compression method is used for the pre computed test sets that are provided by the core vendors, which should be applied to the cores during testing. The code based test data compression technique satisfies the requirements of testing and applied directly to the test patterns.

2 Motivation and Background

2.1 Run Length Based Test Data Compression Techniques

The difference vector technique was applied to the pre computed test sets, followed by Golomb coding. Since Golomb gives better results when the number or zero's in the test set increases. Hence difference vector technique is used to increase the run length of 0's. This paper aims for only runs of 0s. This work [5] produces a compression ratio of 66.24 %. A variable to variable length compression codes presented by [6] that are designed using the distributions of the runs of 0's in a typical test sequences. The result shows that, the FDR

codes outperform the Golomb codes for test data compression. The difference vector technique is done before compression. Similar to the Golomb code, FDR code also forms a table based on run length of 0's and each code word is formed by combining group prefix and tail. The FDR code is applied to the test set after difference vector method. The compression ratio obtained is 68 %. The FDR is efficient for compressing data that has few 1's and long runs of 0's but inefficient for data streams that are composed of both runs of 0s and 1s. An extension to the FDR code called Extended Frequency Directed Code presented by [1] has been proposed. Better compression is achieved when both the runs of 0's and 1's are considered. The grouping in EFDR is also based on the FDR code except adding an extra bit based on the run length. There is an improvement in the compression ratio, as large as, from 19.36 to 80.31 % is seen to be obtained for one of the benchmark circuits.

A double encoding technique was presented by [8]. It is done using run-length encoding followed by Huffman encoding. This method works well only if don't cares in a test set is high. The Huffman coding technique [9] is explained. However, it requires complexity in decoder structure for distinct blocks. Several Huffman based compression techniques such as variable-input Huffman coding [10], variable- to-variable Huffman coding [11], selective Huffman coding [12], optimal selective Huffman coding [13], and complementary Huffman coding [14] and modified selective Huffman coding [15] are proposed to improve compression efficiency, area overhead and test application time. The decoder complexity of larger blocks in the Huffman coding makes it unsuitable for on chip decompression. Compression of the test data by geometric shapes is described in [16]. The cost of encoding block using geometric shapes is higher than the original cost of the blocks. A new method introduced for test data compression which is similar to Alternating variable run length code technique presented by [17]. The Alternating variable run length code (AVR) technique was applied for min test patterns where, don't care is filled with a technique which reduces the switching activity. It includes weighted transition reordering (WTM) method as a preprocessing technique which enhances increase in similarity between input bits and reduces the switching activity. The average and the peak power reduction is 30.43, 82.3 % respectively. Though AVR results in higher compression, the preprocessing techniques for filling and reordering needed to be chosen suitable to make it still more efficient in compression and power reduction.

2.2 Test Vector Reordering Techniques

The test vector reordering technique is used to rearrange the test vectors in a specific order so that the switching activity that takes place between the test patterns gets reduced. The graph theory algorithm based reordering process using the minimum hamming distance between the test vectors to reduce the scan power [18]. The Hamming distance between two test vectors is equal to the number of incompatible bits. 2D hamming distance based reordering [19] helps to reorder the test vectors vertically as well as horizontally. In this paper the compression and power reduction are not so efficient. hamming distance based reordering and column wise bit stuffing with difference vector (HDR-CBS-DV) is presented by [18], which can be used with any run length code technique for higher compression ratio. Four techniques have been applied in prior instead of directly applying code like Golomb, frequency-directed run length (FDR), extended FDR (EFDR), modified FDR (MFDR) or shifted alternate FDR (SAFDR). The compression obtained is improved drastically. This work produces a result of 57.26 % compression ratio. The power reduction was not carried out.

2.3 Don't Care Bit Filling Techniques

The column wise bit filling [18] are used to fill the X bits for reducing the number of transitions. This paper aims to increase the runs of 0s only. This approach not compresses efficiently with compression coding that composed of both runs of 0s and 1s. Hence to improve the compression efficiency and power reduction an analysis of bit filling techniques was performed.

In our research we analyzed the MT filling and also column wise bit filling techniques for finding its efficiency with our remaining flow. We performed reordering method next to get similar patterns nearby. The reordering method of ours will bring similar patterns nearby. That is the (run length of 0s as well as 1s is increased. The switching activity is greatly reduced here itself. Then difference vector method is applied to increase the run length of 0s further. The remaining run length of 1s is also compressed efficiently by our Modified Golomb compression algorithm. This compression algorithm aims best compression for both the run length of 0s and 1s with less decoder complexity. Since pre-processing techniques affect the power as well as compression we analyzed these in our research. The proposed scheme pre-processes the test data using techniques like bit filling; Similarity based reordering, and Difference vector. These techniques are helpful in increasing the run lengths of 0's and 1s thereby reducing switching activity and Power consumption. This flow is combined with Modified Golomb code which gives shorter codeword for both run length of 0's and 1's. If the run length is same runs for 0s and 1s this technique will share the same codeword. Our approach gives higher test data compression and also leads to higher test power reduction. The decoder complexity is very less by using these techniques.

The proposed preprocessing scheme of bit filling and reordering is also analyzed with AVR compression coding [17]. Our proposed flow of preprocessing techniques will result in higher run length of both 0s and 1s. This flow is analyzed with AVR coding since AVR compresses well for both runs of 0s and 1s. In this paper [17] it includes weighted transition reordering (WTM) method as a preprocessing technique. We analyzed our proposed approach of bit filling and similarity based reordering with AVR. This will result in better compression efficiency and power reduction than WTM reordering given in [17]. For bit filling and reordering with any run length based compression techniques our proposed flow of preprocessing can be chosen for improving compression and power reduction with less decoder complexity.

3 Proposed Method

The flow of the proposed technique is the combination of Bit filling; Similarity based reordering, difference vector and compression algorithm for the original test patterns. The proposed flow is given in Fig. 1.

3.1 Don't Care Bit Filling

The test patterns of benchmark circuits are rich in don't cares. They are filled using different bit filling technique. Adjacent fill technique or minimum transition (MT) fill technique is filling the don't care with the same value in the previous bit position. It helps to decrease the transition that takes place between the input bits. The don't care's are filled

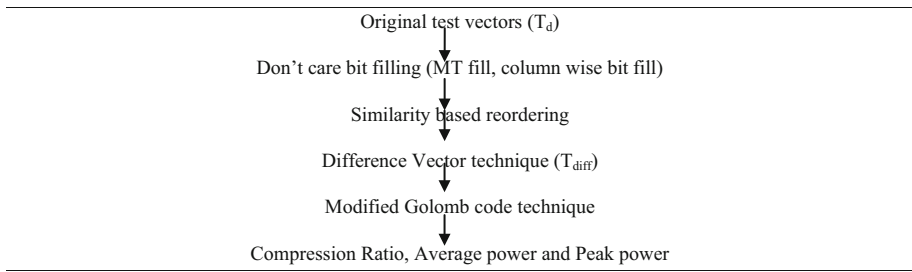


Fig. 1 Design flow for the proposed technique

with zero's are called zero fill and with one's are called one fill technique and these two methods does not reduce the switching activity. Hence, the patterns filled with zero are considered as original test vectors. Column wise bit filling technique (CBF) [15] is done by filling 'X' with the same bit position of the previous vector. Hence the switching activities between the input test vectors are less when compared to original input patterns.

3.2 Similarity Based Reordering Technique

The similar test patterns were brought nearer by the similarity coefficient formula given by Sokal and sneath [20] in the proposed technique. This reduces the incompatible bits in the test patterns. This is the basis for the Similarity based reordering in the proposed method later the difference vector technique, i.e. the xor operation between the adjacent vectors was performed. This difference vector technique increases the run length of 0's since similar patterns were obtained more from the above reordering method. The vector with maximum similarity coefficient value is chosen as the next reference vector. If two or more vectors hold the same similarity coefficient value, then the best value will chosen based on the weighted transition method. The algorithm for similarity based is clearly explained in Sect. 3.6.

3.3 Difference Vector Technique

According to [15] difference vector (DV) of two consecutive vectors will further increase the number of zeroes and hence compression will increase. The difference vector (T_{diff}) is obtained from $T_{diff} = \{t_1; t_1 \text{ xor } t_2, \dots, t_{n-1} \text{ xor } t_n\}$ where a bit-wise exclusive OR operation is carried out between patterns. The successive test patterns in a test sequence often differ in only a minimum number of bits. Hence, the patterns are reordered using Similarity based on maximum similarity between the test vectors and the difference vector technique is carried out. The test patterns will have maximum runs of 0s. There runs of 1s. A compression coding that compresses efficiently for both 0s and 1s is taken which leads to maximum compression efficiency and power reduction with less decoder complexity.

3.4 Modified Golomb Code Compression Technique

The Modified Golomb coding compresses efficiently for both runs of 0s and 1s with less decoder complexity. The encoding procedure of Golomb code in [3] is based on the group

size, where parameter (m) referred to as the group size. The parameter (a) used to choose the type of the run length. The modified Golomb code is similar to the normal Golomb code and the modification is done by writing the code word for run length of 1's. The best value of 'm' can only be determined through actual experimentation. Once 'm' is determined, the runs of 0s and 1s in the test data stream are mapped to groups of size m. the set of run lengths {0, 1, 2, m - 1} forms a group A₁. The set {m, m + 1, m + 2, 2m - 1} forms group A₂.

1. In general, the set of run lengths {(k - 1)m, (k - 1)m + 1, (k - 1)m + 2, km - 1} comprises group A_k.
2. Group prefix of (k - 1) 1s followed by a 0 which is denoted as 1^(k-1)0.
3. If m is chosen to be power of 2 i.e., m = 2^N members and a tail uniquely identifies each member within the group.

The code word for modified Golomb code is formed by combining the prefix and suffix. The table for modified Golomb code is shown in Table 1.

3.5 Alternating Variable Run Length Code

Our proposed preprocessing can be combined with any run length coding technique for efficient compression and power reduction. Since AVR coding shows that the technique compresses efficiently for 0s and 1s, careful bit filling and reordering makes AVR coding still improves compression. Our proposed preprocessing method of filling and reordering is analyzed with this AVR technique for proving its efficiency. The alternating variable run-length code [3] is also a variable-to-variable-length code and consists of two parts—the group prefix, tail. Here, the prefix identifies the group in which the run length lies and the tail identifies the member within the group. There were two group prefixes in each group and second prefix was the inverted data of 1st prefix in the same group. Each prefix was associated with half of the data from the group. According to the test data type, the original test data could be classified into runs of 1s ending with 0 and runs of 0s ending with 1. The code word for alternating variable run length code (AVR) is shown in Table 2.

Table 1 Modified Golomb code technique

Group	Run length		Group prefix	Tail	Code word
	a = 0	a = 1			
A ₁	1	1	0	01	001
	2	2		10	010
	3	3		11	011
A ₂	4	4	10	00	1000
	5	5		01	1001
	6	6		10	1010
	7	7		11	1011
A ₃	8	8	110	00	11000
	9	9		01	11001
	10	10		10	11010
	11	11		11	11011

Table 2 Alternating variable run length code technique (Avr)

Group	Run length	Group prefix	Tail	Code word
A1	1	01	0	010
	2		1	011
	3	10	0	100
	4		1	101
A2	5	001	00	00100
	6		01	00101
	7		10	00110
	8		11	00111
	9	110	00	11000
	10		01	11001
	11		10	11010
	12		11	11011
A3	13	0001	000	0001000
	14	1110	001	0001001
	–	–	–	–
	28	–	111	1110111

3.6 Algorithm

- The Min test patterns were taken and the bit filling technique is carried out. The proposed technique is done with different bit filling techniques such as zero fill, Minimum transition or Adjacent bit filling technique and column wise bit filling technique.
- The first pattern was taken as the reference pattern and the next reference pattern was chosen by the sokal and sneath similarity coefficient method. It is given by,

Sokal and sneath similarity coefficient =

$$a / (a + 2(b + c)) \tag{1}$$

where a = 11, b = 10, c = 01, d = 00.

The highest value sokal and sneath similarity coefficient vector is the next reference vector when compared to other vectors. Hence, the position of the vector with highest sokal and sneath similarity coefficient value is swapped from its position to the next reference position by using swap operator.

- If more than one pattern has the same maximum similarity coefficient value from the first pattern, then the weighted switching transition metric [17] is applied to find the next reference vector.
- After finding the second reference vector, the position is updated by using the swap operator. It swaps the position of the second reference vector and continues the next iteration.
- The iteration continues till the optimum solution is attained and the test vector are reordered using similarity based reordering technique.
- The next step is to find the difference vector technique for the reordered test pattern. This helps to increase the run length of zero and minimize the run length of one to

increase the compression ratio. The difference vector technique is done by EXOR operation between the successive vectors.

- The code words are applied for different run length using Modified Golomb code table shown in Table 3 the compression ratio is calculated using the formula,

$$\text{Compression ratio (\%)} = \frac{(\text{Original test data} - \text{Compressed test data})}{(\text{Original data})} \times 100. \quad (2)$$

3.7 Scan in Power Analysis

In the CMOS circuit, the pre-dominant fraction of the power was dissipated when that element switches. During testing, the circuit was entirely controlled by the test patterns applied to it. The power dissipated when two applying vectors could be compared by the weighted transitions in the vector. Scan vectors with higher weighted transition gives high power dissipation. The weighted transition method is calculated by using the equation given in [17]. The peak power and the average power is found using the formula given in [17]. The vector which has highest weighted transition gives the peak power and the average power is determined by dividing the total transition power with 'n' number of test patterns.

4 Decompression Architecture

Decompression architecture for testing soc designs when modified Golomb coding is used for test data compression shown as in Fig. 2. The decoder decompresses the encoded test set (T_E) and outputs the difference vector patterns (T_{diff}). Compared to run length coding, the synchronization mechanism for modified Golomb coding is more involved since both the code words and the decompressed data can be of variable run length. The Golomb decoder can be efficiently implemented by a $\log_2 m$ -bit counter and a finite-state machine (FSM). The block diagram of the decoder is shown in Fig. 3.

The bit_in is the input to the FSM and an enable (en) signal is used for input the bit when the decoder is ready. The signal (inc) is used to increment the counter and (rs) denotes that the counter finished counting. Signal out is the decoder output and v indicates when the output is valid. Hence, the decoder gives the output of the test patterns of the difference vector technique. The decoder of modified Golomb code technique can be implemented by making a small modification to the Golomb code technique. A toggle flip flop, exclusive-OR gate are required to switch between $a = 0$ and $a = 1$. Further the reordered test sets can be obtained by including exclusive-or gate and the cyclic scan register (CSR). These are used to generate the test patterns from the difference vectors. The steps in the design methodology are done for the Min test patterns of s5378, s9234, s13207, s15850, s38417 ISCAS'89 benchmark circuits. The proposed test vector reordering technique is very efficient when combined with modified Golomb code technique. To check the performance of reordering technique, the similarity based reordering technique and difference vector technique is combined with Alternating variable run length code technique and shows improvement in the compression ratio which is discussed in chapter 4.

Table 3 Compression ratio of proposed technique (MT fill + similarity based reordering + DV + modified Golomb) over original data

ISCAS 89 circuits	No of test vectors	No of bits per vector	Total bits (original data)	Group size of the proposed technique			CR of MT fill + similarity based reorder + DV + modified Golomb code (%)	
				M = 4	M = 8	M = 16		
				M = 32				
S5378	111	214	23,754	11,904	11,881	13,258	15,266	49.98
S9234	159	247	39,273	19,069	19,184	21,356	24,671	51.44
S13207	236	700	165,200	31,141	31,676	31,937	32,336	81.14
S15850	126	611	76,986	27,466	26,985	30,072	33,778	64.94
S38417	99	1664	164,736	69,177	66,605	71,104	80,413	59.56
Average	-	-	-	-	-	-	-	61.42

The bold values inside the table reflect that the corresponding group size compresses more bits than other group sizes for the circuit. The compression ratio is also calculated for the compressed bits of this same group size reflected in bold

Fig. 2 Decompression architecture

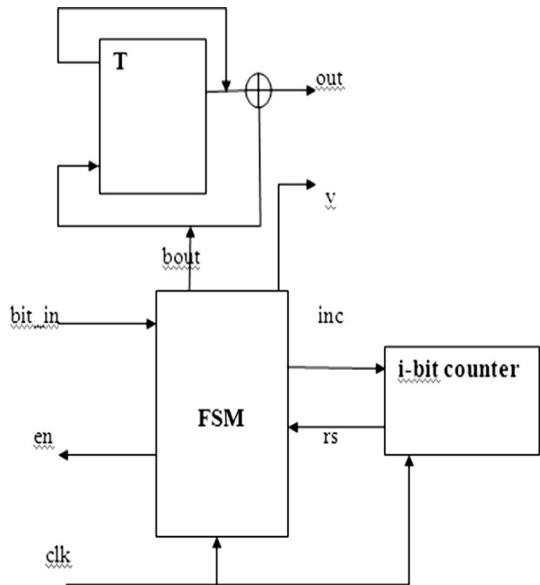
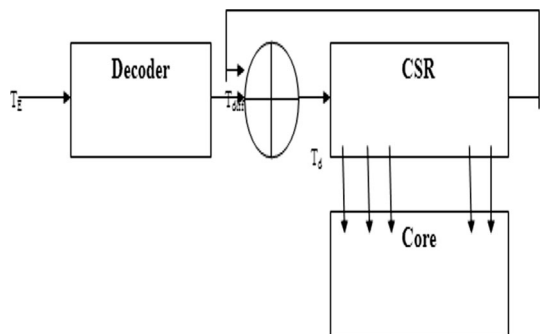


Fig. 3 Block diagram of decoder used for decompression



5 Results and Discussion

The proposed algorithm and the functionality are verified using Modelsim 6.5b. The analysis is carried out for the Min test patterns of ISCAS'89 benchmark circuits such as S5378, S9234, S13207, S15850 and S38417. The proposed techniques are applied for the predefined Min test vectors of ISCAS'89 benchmark circuits such as s5378, s9234, s13207, s15850, s38417 and the result is compared with the original number of bits used in the predefined test vectors. The group size 'm' of modified Golomb code is varied to obtain the best compression ratio and each circuit has given a better compression ratio in different group size. The best compression ratio can be obtained by changing the group size and verified experimentally. The circuit s5378, s15850 and s38417 gives higher compression ratio when the group size "m = 8". Similarly, the circuit s9234 and s13207 gives higher compression ratio when group size is "m = 4". First the minimum transition fill is done for the test patterns and analyzed with the similarity based reorder, difference vector and modified Golomb coding. The Table 3 shows the compression ratio (CR) of the proposed

Table 4 Compression ratio of proposed technique (CBF + similarity based reordering + DV + modified Golomb) over original data

ISCAS'89 Benchmark circuit	Total bits (original data)	Group size of the proposed technique				CR of CBF + similarity based + DV + Modified Golomb code (%)
		M = 4	M = 8	M = 16	M = 32	
S5378	23,754	10,635	9991	10,742	12,146	57.93
S9234	39,273	16,533	15,536	16,440	18,598	60.44
S13207	165,200	25,776	27,641	25,316	24,131	85.39
S15850	76,986	23,680	22,324	23,443	25,472	71.50
S38417	164,736	60,576	58,541	60,607	67,447	64.46
Average	–	–	–	–	–	67.94

The bold values inside the table reflect that the corresponding group size compresses more bits than other group sizes for the circuit. The compression ratio is also calculated for the compressed bits of this same group size reflected in bold

technique which includes Minimum transition fill technique, Similarity based test vector reordering, Difference vector (DV) technique and modified Golomb code technique. The average compression ratio (CR) is obtained for (MT fill + similarity based reorder + DV + Modified Golomb) is 61.42 % and shown in Table 3.

The analysis of column wise bit filling is also done. The Table 4 shows that the compression ratio (CR) of the proposed technique which includes Column wise bit filling technique; Similarity based test vector reordering, Difference vector technique (DV) and modified Golomb code technique. These techniques are applied for the predefined Min test vectors of ISCAS'89 benchmark circuits such as s5378, s9234, s13207, s15850, s38417 and the results are compared with the original number of bits used in the predefined test vectors. The group size 'm' of the modified Golomb code was varied to obtain the best compression ratio and each circuit has given a better compression ratio in different group size. The best compression ratio can be obtained by changing the group size Golomb) is 67.94 % and shown in Table 4 and verified experimentally. The circuit s5378, s9234, s15850 and s38417 gives higher compression ratio when the group size "m = 8". Similarly, the circuit s13207 gives higher compression ratio when group size is "m = 32". Hence, the average compression ratio (CR) obtained for (CBF + SIMILARITY BASED reorder + DV + Modified Golomb) is 67.94 %, shown in Table 4. The column wise bit fill gives higher compression for the proposed technique.

The compression ratio (CR) of proposed technique which includes column wise bit filling (CBF), Similarity based reordering, difference vector technique (DV) and modified Golomb code is compared with other existing techniques along with their compression ratio such as Golomb code technique [5], Frequency directed run length (FDR) [6], Alternating FDR (AFDR) [7], Extended FDR (EFDR) [2], nine coded (9C) [21], technique and block merging (BM) [22], AVR [17], with respect to their original number of bits. The proposed technique is applied to different ISCAS'89 benchmark circuits. The average compression ratio obtained for proposed techniques 67.94 % and it is higher than other methods used previously and the comparison is shown in Table 5. The comparison graph of compression ratio of proposed technique with other techniques over original data is shown in Fig. 4. It shows that the compression ratio for proposed technique (CBF + similarity based reorder + DV + modified Golomb code) is more when compared to other existing techniques such as Golomb code technique (Feng and G. Li.,2008), Frequency directed run length code (FDR) [4]), Alternating FDR (AFDR), Extended FDR

Table 5 Compression ratio of proposed technique (CBF + similarity based reordering + DV + modified Golomb) with other techniques over original data

ISCAS'89 benchmark circuit	Total no of bits (original data)	Golomb (%)	FDR (%)	AFDR (%)	EFDR (%)	9C (%)	BM (%)	Proposed technique (%)
S5378	23,754	37.11	48.02	50.77	53.67	51.64	51.19	57.93
S9234	39,273	45.25	43.59	44.96	48.66	50.91	56.86	60.44
S13207	165,200	79.74	81.30	80.23	82.49	82.31	84.89	85.39
S15850	76,986	62.82	66.22	65.83	68.66	66.38	69.49	71.50
S38417	164,736	28.37	43.26	60.55	62.02	60.63	59.39	64.46
Average	–	50.65	56.47	60.46	63.10	62.37	64.36	67.94

(EFDR) [5], nine coded technique (9C) [21] and Block merging (BM) [22]. The proposed preprocessing techniques of column wise bit fill, Similarity based reordering and difference vector gives better result when compared to other preprocessing techniques. The proposed reordering (CBF + similarity based reorder +DV) when preprocessed with other compression techniques such as FDR, Golomb, AVR improves the compression ratio as well as power reduction with lesser decoder area.

The analysis of our proposed preprocessing flow is carried out with Alternate variable run length code (AVR). Since the AVR coding compresses well for both runs of 0s and 1s, the suitable preprocessing techniques of bit filling, reordering with AVR still improves the compression and power reduction. Thus AVR coding [17] is chosen and our proposed preprocessing is analyzed with AVR code. The compression ratio of our proposed preprocessing with AVR is improved efficiently when compared to that of the given results of AVR [17]. To compare the proposed preprocessing with other preprocessing technique, like WTR [15], a simulation of weighted transition reordering flow [15] preprocessed to AVR [17] is performed. The Table 6, shows the comparison of proposed preprocessing flow combined to Alternating variable run length code (AVR) with weighted transition reordering (WTR) flow [15] combined to AVR. The result shows that the similarity based reordering flow, i.e. proposed preprocessing method gives better compression when combined with AVR. The compression ratio increases up to 2–3 % for our proposed method when compared to WTR [15] flow combined with AVR. These techniques are applied for Min test vectors of ISCAS'89 benchmark circuits. The scan in peak power for the compressed bits using proposed technique is calculated using equation given in [17], and the original peak power is obtained by mapping the don't care with zero. The Table 7 shows the results of scan in peak power of the proposed technique which includes column wise bit filling, Similarity based reordering, difference vector technique and the modified Golomb code technique. The results shows decrease in scan in peak power of about 83.36 % when compared to original scan in peak power. The scan in Average power for the compressed bits using proposed technique is calculated using equation given in [17], and the original Average power is obtained by mapping the don't care with zero. The Table 8 shows the scan in average power of the proposed technique which includes column wise bit filling, Similarity based reordering, difference vector technique and the modified Golomb code technique. The results shows decrease in scan in average power of about 63.35 % when compared to original scan in average power. This proposed method gives higher peak and average power reduction than other existing techniques. The decoder complexity also less in our proposed approach.

Fig. 4 Compression ratio comparison of proposed technique with other existing techniques

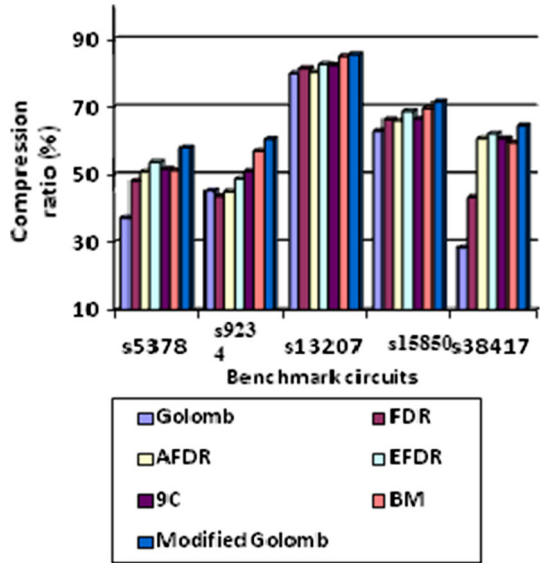


Table 6 Compression ratio (CR) (%) of proposed technique with AVR technique

ISCAS'89 benchmark circuit	Total no of bits (original data)	Weighted transition reorder (WTR) + CBF + DV with AVR No of bits CR %	Proposed technique (CBF + similarity based reorder + DV + with AVR) No of bits CR %
S5378	23,754	9233 61.13	8826 62.80
S9234	39,273	16,129 58.93	14,512 63.04
S13207	165,200	23,896 85.54	19,928 87.93
S15850	76,986	19,388 73.82	19,698 74.41
S38417	164,736	51,679 68.63	50,164 69.54
Average	–	69.61	71.54

6 Conclusions

Test data compression technique with suitable bit filling and reordering methods used to reduce the power consumption while testing the circuits. Since the volume of test data used for testing is higher, these preprocessing methods are helpful in reducing the test volume and power consumption. Hence, the proposed technique which involves column wise bit filling, Similarity based test vector reordering and modified Golomb code method of test data compression is found efficient to increase the compression ratio of about 67.94 % over other existing test data compression techniques. The proposed preprocessing technique is very efficient when combined with other test data compression methods. The decoder complexity is also less in our proposed approach. The proposed reordering technique when combined with Alternating Variable run length technique gives 2–3 % increase in compression ratio when compared with weighted transition based Reordering (WTR) combined with AVR. Similarly the proposed technique shows improvement in reduction of scan in average power and scan in peak power of about 63.35 and 83.36 % respectively. In future

Table 7 Reduction in peak power of proposed technique

ISCAS'89 benchmark circuit	Scan in peak power of Test vectors with 'X' mapped to '0' (original data)	Scan in peak power of (CBF + similarity based + DV + modified Golomb)	Reduction in peak power (CBF + similarity based reorder + DV + modified Golomb) over original data (%)
S5378	10,127	2446	75.84
S9234	12,994	3176	75.55
S13207	101,127	3314	96.72
S15850	81,832	9780	88.04
S38417	505,295	97,605	80.68
Average	–	–	83.36

Table 8 Scan in average power of proposed technique

ISCAS'89 benchmark circuit	Scan in average power of test vector with 'x' mapped with '0' (original data)	Scan in average power of (CBF + similarity based reorder + DV + Modified Golomb)	Reduction of average power (CBF + similarity based reorder + DV + Modified Golomb) over original data (%)
S5378	3336	1761.23	47.20
S9234	5692	2141.26	62.38
S13207	12,416	2071.95	83.31
S15850	20,742	6685.80	67.76
S38417	172,665	75,767.83	56.11
Average	–	–	63.35

works, the proposed technique can be used for preprocessing technique for efficient results which does not require any on chip overhead for the process.

Acknowledgments The authors would like to thank PROF. Nur. A. Touba for his valuable support by providing the scan in Mintest vectors of ISCAS'89 benchmark circuits.

References

1. El-Maleh, A., & Al-Abaji, R. (2002). Extended frequency-directed run-length code with Improved application to system-on-a-chip test data compression. *Proceedings of the International Conference on Electronics, Circuits and Systems*, 2, 449–452.
2. Chandra, A., & Chakrabarty, K. (2003). Test Data Compression and test resource partitioning for system-on-a-chip using frequency-directed run-length (FDR) codes. *IEEE Transactions on Computers*, 52(8), 1076–1088.
3. Feng, J. & Li, G. (2008). A test data compression method for system-on-a- chip. In *Proceedings of the 4th IEEE International Symposium on Electronic Design, Test and Applications*.
4. Touba, N. A. (2006). Survey of test vector compression techniques. *IEEE Design & Test Magazine*, 23(4), 294–303.
5. Chandra, A., & Chakrabarty, K. (2001). Efficient test data compression and decompression for system-on-a-chip using internal scan chains and Golomb coding. In *DATE '01: Conference on design, automation and test in Europe*.

6. Chandra, A., & Chakrabarty, K. (2001b) Frequency—Directed run-length (FDR) codes with application to system-on-a-chip Test data compression. In *the Proceedings of the 19th IEEE VLSI Test Symposium*.
7. Chandra, A., & Chakrabarty, K. (2002d) Reduction of SOC test data volume, scan power and testing time using alternating run-length codes. In *DAC '02: Proceedings of the 39th conference on Design automation*.
8. Nourani, Mehrdad, & Tehranipour, Mohammad. (2005). RL-Huffman encoding for test compression and power reduction in scan application. *ACM Transactions on Design Automation of Electronic Systems*, 10(1), 91–115.
9. Jas, A., Jayabrata, G., & Touba, N. A. (1999). Scan vector compression/decompression using statistical coding. *Proceedings of IEEE VLSI Test Symposium, IEEE, LosAlamitos, CA, UnitedStates, 1999*, 114–120.
10. Gonciari, P. T., AlHashimi, B. M., & Nicolici, N. (2003). Va-riable-length input Huffman coding for system-on-a-chip test. *IEEE Transaction Computer-Aided Design Integrtration Circuits Systems*, 22(6), 783–796.
11. Kavousianos, X., Kalligeros, E., & Nikolos, D. (2008). Test data compression based on variable-to-variable Huffman encoding with code word reusability. *IEEE Transaction on Computer-Aided Design of Integrated Circuits and Systems*, 27(7), 1333–1338.
12. Jas, A., Ghosh-Dastidar, J., & Touba, M. N. (2003). An efficient test vector compression scheme using selective Huffman coding. *IEEE Transaction on Computer-Aided Design of Integrated Circuits and Systems*, 22(6), 797–806.
13. Kavousianos, X., Kalligeros, E., & Nikolos, D. (2007). Optimal selective Huffman coding for test-data compression. *IEEE Transactions on Computer*, 56(8), 1146–1152.
14. Lu, S. -K., Chuang, H. M., Lai G. Y., Lai, B. -T., & Huang Y. -C. (2009). Efficient test pattern compression techniques based on complementary Huffman coding. In: *IEEE Circuitsand Systems International Conferenceon Testing and Diagnosis, ICTD009*.
15. Mehta, U. S., Dasgupta, K. S., & Devashrayee, N. M. (2010). Weighted transition based reordering, columnwise bit filling, and difference vector: a power-aware test data compression method. *VLSI Design, 2011*, 9.
16. ElMaleh, A., AlZahir, S., & Khan, E. (2011). Test data compression based on geometric shapes. *Computers and Electrical Engineering*, 37(3), 376–391.
17. Ye, B., Zhao, Q., & Duo, Z. (2011). Test data compression using Alternating variable run length code. *Integration, the VLSI Journal*, 44(2), 103–110.
18. Mehta U. S., Dasgupta K. S., & Devashrayee N. M. (2010a) Hamming distance based reordering and column wise bit stuffing with difference vector: a better scheme for test data compression with run length based codes. In *International Conference on VLSI Design*.
19. Mehta U. S., Dasgupta K. S., & Devashrayee N. M. (2010b) Hamming distance based 2-D reordering with power efficient dont care bit filling optimizing the test data compression method. In *9th International Symposium on System-on-Chip. Tampare, Finland*.
20. Sokal, R. R., & Sneath, P. H. A. (1963). *Principles of numerical taxonomy* (p. 359). San Francisco: Freeman.
21. Tehranipoor, M., Nourani, M., & Chakrabarty, K. (2005). Nine coded compression technique for testing embedded cores in SoCs. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 13(6), 719–731.
22. El-Maleh, A. H. (2008). An efficient test vector compression technique based on block merging. *IET Computers and Digital Techniques*, 2(5), 327–335.



Dr. R. HariKumar received his B.E (ECE) Degree from REC (NIT) Trichy 1988. He obtained his M.E (Applied Electronics) degree from College of Engineering, Guindy, and Anna University Chennai in 1990. He was awarded Ph.D. in I and C Engg from Anna university Chennai in 2009. Currently he is working as Professor ECE at Bannari Amman Institute of Technology, Sathyamangalam. He has published forty-five papers in International and National Journals and also published around Eight two papers in International and National Conferences conducted both in India and abroad. His area of interest is Bio signal Processing, Soft computing, VLSI Design and Communication Engineering. He is life member of IETE, IEEE and ISTE.



B. Manjurathi received her B.E (ECE) Degree from SRM Engineering college, Chennai 2004. She obtained her M.E (VLSI Design) degree from PSG College of Technology, Coimbatore in 2006. Currently she is working as Assistant Professor ECE at Karunya University, Coimbatore. She is a Ph.D candidate in I and C Engg from Anna university, Chennai. Her area of interest is Low power VLSI Design and testing. She is life member of ISTE.