Analysis of test data compression and power reduction using multiple encoding for Opto Electronic circuits

B. MANJURATHI[a], R. HARIKUMAR RAJAGURU[b]
[a] *Assistant Professor, Department of Electronics and Communication Engineering, Karunya University, Coimbatore, India*
[b] *Professor, Department of Electronics and Communication Engineering, Bannari Amman Institute of Technology, Sathyamangalam, India*

System on chip is the major challenging issue for both design engineers and testing engineers due to its high power consumption. During testing, the volume of the test data is extremely high when compared to normal mode and because of that the switching activity which takes place between the test vectors leads to increase in power consumption. The Proposed technique is the combination of hamming distance based reordering, Bit stuffing and encoding for the original test patterns. It improves the compression ratio and also gives reduction in Average, peak power of MINTEST test sets of ISCAS'89 benchmark circuits.

(Received October 10, 2015; accepted February 10, 2016)

Keywords: Column wise bit filling technique, Hamming distance, Reordering, Multiple run length code technique, Selective Huffman encoding, Power reduction, VLSI

1. Introduction

Advancement in VLSI technology resulted in a change in the design method where millions of transistors are integrated on a single chip. As the complexity increases, the difficulty and the cost of testing the chip also increases rapidly [1, 2]. The important challenge in testing system on a chip is dealing with the large volume of test data that should be stored in the test memory and transferred between the tester and the circuit under test (CUT). Thus to reduce the testing time and cost, it is necessary to reduce the volume of test data [3]. Similarly, reducing the test data volume also enhances to reduce the peak and the average power. Moreover, the advantage of test data compression is that it generates the complete set of patterns applied to the circuit under test (CUT) with Automatic Test Pattern Generator (ATPG), and this set of test patterns are optimized with respect to desired fault coverage [4]. Compression is mainly classified into lossless and lossy compression methods. Lossless method of compression can reconstruct the original data but lossy can reconstruct only the approximation of the original data.

Test data compression is divided in to three categories based on the test vector which we use [4]. They are code based schemes, linear decompression based scheme and broadcast scan based scheme. Considering the suitability of IP core based system on chip, code based test data compression scheme is more suitable. From various code based test data compression scheme like constructive codes, dictionary codes, statistical codes and the run length based codes is more suitable because of its simple on chip decoder and better compression capacity [5].To improve the compression ratio and to reduce the switching activity in the run length based data compression, hamming distance based reordering, column wise bit filling, difference vector technique is proposed. These techniques are helpful in increasing the run length as well as decrease the switching activity in the test data. Further process of multiple run length code technique is used to improve the compression ratio. Multiple run length code technique involves different run length code technique which is used for test data compression.

The paper is organized as follows: the motivation and background for test data compression methods and test vector reordering for test power reduction is covered in section 2. The section 3 presented with the proposed method and the section 4 includes the design methodology of different modes of operation and the section 5 contains a brief explanation of decompression architecture and section 6 holds the experimental results performance comparison is discussed. The conclusion part is presented in section 7 with references.

2. Motivation and background

2.1 Test data compression techniques

Since the run length based coding scheme is more suitable for IP core based system on chip, test data compression based on Golomb coding [6, 7] was proposed earlier. The encoding is based on runs of 0s with variable length code words. Another technique which shows better improvement than Golomb code is frequency directed run length (FDR) code [8, 9] which is also based on variable length code words. The main difference between Golomb and FDR is their group size. Maleh and Abaji extended the frequency directed run length code as extended frequency directed run length code (EFDR) [3] which encodes both the run length of 0s and 1s. Alternating run length code (AR) based on FDR code, which encodes both runs of 0s and 1s is proposed in [10]. Then Modified FDR code used in [11] shows improved compression ratio than the normal FDR. The explanation for each run length code is shown in [12]. Test data compression using Huffman encoding [13] is proposed which improves the compression ratio compared to older technique.

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 [14]. 2D hamming distance based reordering [15] helps to reorder the test vectors vertically as well as horizontally. Similarly the Artificial intelligence (AI) based reordering technique [16] which helps to reduce the switching activity that takes place. Particle Swarm Optimization based reordering also shows improvement in both scan power and the leakage power.

2.3 Don't care bit filling

Don't care bit filling is used to fill the don't care bits by any values in the test patterns. There are various bit filling techniques used to fill the X bit. 0-fill, 1-fill, random fill are the techniques used previously. Later MT-fill and column wise bit filling [14] is used to fill the X bits that reduce the transition. The genetic algorithm based heuristic to fill the don't cares is proposed [17].

3. Proposed method

The Proposed technique is the combination of hamming distance based reordering, Bit stuffing and encoding for the original test patterns. The proposed technique flow is shown in fig 1.

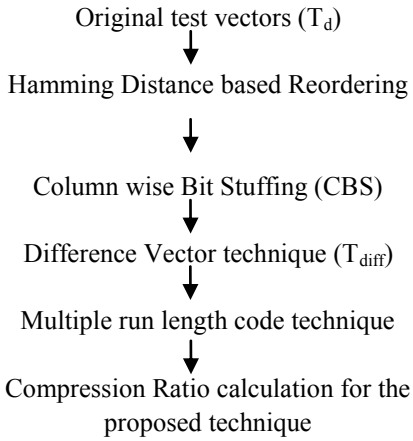

Fig.1. Design flow for the proposed technique

3.1 Selection of first reference pattern

The reference pattern is chosen according to [14], where the original test patterns (T_d) with minimum number of don't care is selected. If more than one test pattern have minimum don't care values, then any one can be selected. Hence the first test pattern is chosen as a reference vector for further steps.

3.2 Hamming distance reordering

Once the reference vector is chosen, the patterns with minimum Hamming distance from the reference vector should be placed as a second reference vector [14]. The hamming distance between two test patterns is equal to the number of incompatible bits. For example, consider two test patterns (F_1 = 1 0 1 X 0 1) and (F_2 = 0 0 1 X 1 1). The distance d (F_1, F_2) is 2 because the first and the fifth bits in the patterns are incompatible. Each test pattern is considered as a vertex in an undirected graph G and the distance between two patterns will be the weight present between them. The vector which has the less hamming distance is placed as a second reference vector. This process is repeated to find the consecutive reference vector.

3.3 Column wise bit filling

The don't care bit of the first reference pattern is simply filled with 0s and the remaining vectors are column wise bit filled to make similar combinations of test patterns. Column wise bit filling (CBF) used in [14] is used to fill don't care bits for the remaining test vectors. CBF is the method of filling the don't care with that of the same value in the previous column.

3.4 Difference vector technique

Difference vector technique is the method of doing EXOR operation between two consecutive vectors, will further increases the number of zeroes and hence compression ratio will increase [14]. The difference vector (T_{diff}) is obtained from $T_{diff} = \{t_1; t_1 \text{ xor } t_{2;\, tn-1} \text{ xor } t_n\}$ where a bit-wise exclusive OR operation is carried out between patterns.

4. Multiple run length code

4.1 Different mode of operation

Multiple run length code is based on multi code compression (MCC) technique [18] which is the combination of different run length codes like Golomb, FDR, AR and IFDR. In the proposed method, only Golomb and AR are used for both run length of 0s and 1s. The following assumptions are very useful to improve the compression ratio in multiple run length code.

Table 1. Multiple run length code.

Run length type		Compression mode
a = 0	a = 1	
1	1	Bypass mode
2	2	Bypass mode
3	3	AR
4	4	AR
5	5	AR
6 or higher	6 or Higher	Golomb

1. If the run length of 0s and 1s is 1 and 2, no compression is required; hence bypass mode is used [18].
2. If the run length of 0s and 1s are 3 to 5, Alternating run length encoding technique is used.
3. If the run length of 0s and 1s exceeds 6 and above, then Golomb encoding technique is used with same code words for both 0s and 1s run length by including a parameter 'a' which denotes the 0 and 1 run length. Table I shows the multiple run length code technique.

4.2 Modified Golomb code

The actual encoding procedure of Golomb code in [6, 7] is based on the group size parameter 'm'. The best value of m for the test data compression is determined through actual experimentation. The same technique is employed here also by an extra parameter 'a' which denotes the runs of 0s and 1s in the test data stream that are mapped to groups of size m. According to [6, 7] the set of run lengths {0, 1, 2…m-1} forms a group A_1. The set {m, m+1, m+2…2m-1} forms group A_2.

1. Initially the run lengths {(k-1) m, (k-1) m+1, (k-1) m+2... km-1} comprises group A_k is determined.
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 and the modified Golomb code method is shown in table 2.

Table 2. Modified golomb code for m=4.

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

4.3 Alternating run length code

Table 3. Alternating run length code

Group	Run length type		Group Prefix	Tail	Code word
	a=0	a=1			
A_1	0	0	0	0	00
	1	1		1	01
A_2	2	2	10	00	1000
	3	3		01	1001
	4	4		10	1010
	5	5		11	1011
A_3	6	6	110	000	110000
	7	7		001	110001
	..	..		..	…
	13	13		111	110111

The alternating run length code is also a variable to variable run length code and consists of two divisions, group prefix and tail. The group prefix identifies the group in which the run length lies and the tail identifies the member within the group [10]. An additional parameter associated with this code is the alternating binary variable

'a'. If a=0, it is 0 run length type and if a=1, the run length is of type 1. Hence the Alternating run length or (AR) technique codeword is shown in table 3. It is similar to the FDR code except with an additional parameter. For any codeword the prefix and the tail are of equal length. The prefix length of the group A_i equals 'i'. Hence the codeword size increases by two bits as we move from group A_i to A_{i+1} [10].

4.4 Selective huffman encoding

This technique is used to perform the double compression after the data is compressed by the multiple run length Code Compression technique, for analyzing compression effect only. From the compressed data, another compression by Huffman code is done by [19]. Huffman code is constructed by the frequency of number of repeated data that occurs from the first compressed data. Hence the next level of compressed data is obtained by Huffman code. To get the best compression ratio we apply selective Huffman encoding to the encoded block length values. This idea is to assign minimum number of bits to the codeword that occur more frequently and larger number of bits to those that occur less frequently. The amount of compression that can be achieved with Huffman coding depends on the skewed frequency.

4.5 Algorithm of the proposed technique

- The first step is to find the reference pattern from T_d which has minimum number of don't cares. If more than one pattern has same number of don't cares, then any one of the pattern should be selected.
- The test vectors are reordered by finding the minimum Hamming distance between each test pattern. Similarly, if more than one pattern has same distance then any one can be selected.
- Once the test patterns are reordered, the normal 0 filling is done for the first reference vector.
- Then the column wise bit filling (CBF) should be done to fill don't care with the same bit present in the successive vectors. Column wise bit filling is the useful technique which increases the run length of 0s after combining with difference vector technique method.
- Next step is to find the difference vector (EXOR) operation between the successive vectors. Let V_i be the test vectors where {i = 1, 2, 3…n}. The EXOR operation is done by {V_1 xor V_2, V_2 xor V_3…V_{n-1} xor V_n}.
- Next step is to apply the code word for both run length using multiple run length code technique.
- The selective Huffman encoding is applied for the code words obtained from multiple run length code.
- The compression ratio (CR) for the proposed technique is calculated by the following formula.
-

$$CR\% = (T_d - T_e) * 100 \quad (1)$$

Where,

CR% = Compression ratio in percentage

T_d = Original test data

T_e = Compressed test data

Hence the compressed test vector is obtained by applying the multiple run length code technique. CBF and difference vector technique are more suitable to increase the run length of 0s by performing the EXOR operation between the successive test vectors.

5. Experiment results

In this section, we experimentally evaluate the proposed compression technique for the ISCAS'89 benchmark circuits. The set of test vectors are reordered using Hamming distance and the unspecified bits are filled using column wise bit filling method and the difference vector technique is applied. Test data is compressed using multiple run length code and the group size of Golomb (m=16) is chosen for the compression process. The table 1 presents the experiment result for the ISCAS'89 benchmark circuits with test sets T_d obtained from Mintest ATPG.

The compressed data T_e obtained after multiple run length code technique is used to calculate the compression ratio using equation 1. Table IV shows the compression result for five ISCAS'89 benchmark circuits and the peak compression ratio of about 91.6% is obtained for S13207 circuit. Table V shows the comparison of compression ratio of the proposed technique with different run length technique. The results for FDR, EFDR, MFDR, Golomb and AFDR are obtained from [14]. The peak power and the average power can be calculated using the power formula given in the reference [20].

The reduction of average power for each benchmark circuit is shown in table VI. Similarly the reduction of peak power of benchmark circuit by the proposed technique is shown in table VII. The improvement of compression ratio over different technique seems to be drastically increased and the figure 2 shows the compression ratio of various run length code technique with the proposed technique.

Table 4. Compression ratio of multiple run length code over original data.

ISCAS'89 Benchmark circuit	Test patterns	Number of bits per pattern	Total no of bits (original data)	Compressed bits after multiple run length code	CR of multiple run length code (%)	CR of multiple run length with Huffman encoding (%)
S5378	111	214	23754	7053	**70.3**	**74.8**
S9234	159	247	39273	11950	**69.5**	**74.3**
S13207	236	700	165200	13817	**91.6**	**92.7**
S15850	126	611	76986	15796	**79.4**	**82.6**
S38417	99	1664	164736	41398	**74.8**	**78.4**
Average					**77.12**	**80.56**

Table 5. Comparison of compression ratio of the proposed technique with different techniques [14] over original data T_d.

ISCAS'89 benchmark circuits	Test patterns	Number of bits per pattern	Total no of bits (original data) T_d	CR (%) FDR	CR (%) Golomb	CR (%) MFDR	CR (%) EFDR	CR (%) ASFDR	CR (%) Proposed method
S5378	111	214	23754	62.33	52.97	57.26	60.03	57.36	**74.8**
S9234	159	247	39279	61.06	56.05	60.58	57.56	53.23	**74.3**
S13207	236	700	165200	87.47	70.03	87.80	86.40	85.16	**92.7**
S15850	126	611	76986	72.84	62.55	72.82	70.43	67.53	**82.6**
S38417	99	1664	164736	66.18	56.09	62.46	65.67	62.18	**78.4**
Average				69.97	59.53	68.18	68.01	65.09	**80.56**

Table 6. Average power analysis of multiple run length code technique over original data.

ISCAS'89 circuit	Original power of T_d with don't cares mapped to '0'	Compressed data of the proposed technique	Reduction power of proposed from original (%)
S5378	3336	1103	**67.31**
S9234	5692	1593	**72.01**
S13207	12416	898	**92.76**
S15850	20742	4353	**79.01**
S38417	172665	48950	**71.65**
Average			**76.54**

Table 7. Peak power analysis of multiple run length code technique over original data.

ISCAS'89 circuit	Original power of T_d with don't cares mapped to '0'	Compressed data of the proposed technique	Reduction power of proposed from original (%)
S5378	10127	1487	**85.31**
S9234	12994	2033	**84.35**
S13207	101127	1302	**98.71**
S15850	81832	5607	**93.14**
S38417	505295	58840	**88.35**
Average			**89.97**

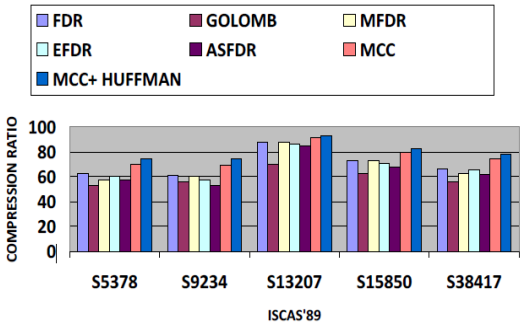

Fig.2. Comparison of compression ratio for ISCAS 89 benchmark circuit.

6. Conclusion

Test data compression technique which includes Hamming Distance Reordering, Column wise Bit Stuffing (CBS), Difference vector and multiple run length Code technique with selective Huffman encoding is proposed. The original test sets for ISCAS'89 benchmark circuits is obtained from Mintest test vectors. MINTEST test vector is a predefined test sets used for testing the circuit under test. The proposed techniques is applied to the MINTEST test vectors and the compression ratio is obtained. The compression ratio for the proposed technique is 80.56% over original data when compared to different run length codes such as GOLOMB, Frequency Directed Run length (FDR), Extended Frequency Directed run length (EFDR), Modified Frequency Directed run length (MFDR) and Alternating Run length codes (AR). Similarly, the average power and the peak power reduced to the average of 76.54% and 89.97% respectively. Hence the result shows that the compression ratio has been drastically improved and the power gets reduced to the maximum amount.

Acknowledgements

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] R. Chandramouli, S. Pateras, Testing Systems on a Chip, IEEE Spectrum, p. 42, Nov.1996.

[2] Y. Zorian, E. J. Marinissen, S. Dey, Testing Embedded- Core Based system chips, Proc. Of Int. Test Conf., p. 130, 1998.

[3] Aiman El-Maleh, R. Al-Abaji, Proc. Int. Conf: on Electronics, Circuits and Systems, **2,** 449 (2002).

[4] A. Nur Tauba, Survey of Test Vector Compression Techniques, IEEE transaction Design & Test of Computers, July 2006.

[5] U. Mehta, K. S Dasgupta, N. M. Devashrayee, Weighted Transition based Reordering, Column wise Bit filling and difference vector- A power aware test data compression method, VLSI Design journal.

[6] Anshuman Chandra, K. Chakrabarty., Efficient test data compression and decompression for system-on-a-chip using internal scan chains and Golomb coding, DATE '2001**.**

[7] Anshuman Chandra, Krishnendu Chakrabarty. IEEE transaction on Computer-Aided design of Integrated circuits and systems, **20**, March 2001.

[8] A. Chandra, K. Chakrabarty, Frequency- Directed Run-Length (FDR) Codes with Application to System-on-a-Chip Test Data Compression, 2001 in the Proceedings of the 19th IEEE VLSI Test Symposium.

[9] A. Chandra, K. Chakrabarty, IEEE Transactions on Computers, **52(**8), 2003.

[10] A Chandra, K. Chakrabarty, Reduction of SOC test data volume, scan power and testing time using alternating run-length codes, DAC '02: Proceedings of the 39th conference on Design automation.

[11] J. Feng, G. Li, A Test Data Compression Method for System-on-a- Chip 2008, 4th IEEE International Symposium on Electronic Design, Test and Applications.

[12] U. Mehta, K. Dasgupta, N. Devashrayee, An Open Access International Journal "VLSI Design", **2010**(670476), 1 January (2010).

[13] M. Nourani, M. Tehranipour, ACM Trans. Design Automat. Electron. Syst., **10**(1), 91(2005).

[14] U. Mehta, K. Dasgupta, N. Devashrayee Hamming distance based reordering and column wise bit stuffing with difference vector: A better scheme for test data compression with run length based codes, Proceedings of 23rd International Conference on VLSI Design (VLSID 10). Bangalore: IEEE, p. 33, 2010.

[15] U. Mehta, K. Dasgupta, N. Devashrayee, Hamming distance based 2-D reordering with power efficient don't care bit filling optimizing the test data compression method. Tampare, Finland: 9th International Symposium on System-on-Chip, , p. 1, 2010.

[16] U. Mehta, K. Dasgupta, N. Devashrayee, Artificial intelligence based scan vector reordering for capture power minimization. Proceedings of (IEEE) International Symposium on (VLSI) ISVLSI-11, June 2011, p. communicated.

[17] S. Kundu, S. Chattopadhyay, Efficient don't care filling for powerreduction during testing, Proceedings of IEEE International Conference on Advances in Recent Technologies in Communication and Computing, pp. 319–323, 2009.

[18] Deepika Sharma and et al., Test data volume minimization using Double hamming Distance Reordering with Mixed RL- Huffman based compression scheme for system on chip, Nirma University International Conference on Engineering 2012.

[19] Abhijit Jas, J. Gosh-Dastidar, M. Ng, Nur A.Touba., IEEE Trans. Computer-Aided Design Integr. Circuits Syst.,. **22**(6), 797 (2003).

[20] Bo Ye, Qian zhao, Duo Z., The Vlsi journal, p. 103, 2011.

*Corresponding author: bmanjurathi@gmail.com