A Hybrid Test Compression Technique for Efficient Testing of Systems-on-a-Chip

Aiman H. El-Maleh

King Fahd University of Petroleum & Minerals, P.O. Box 1063, Dhahran, 31261, Saudi Arabia
Email: aimane@ccse.kfupm.edu.sa

ABSTRACT

One of the major challenges in testing a System-on-a-Chip (SOC) is dealing with the large test data size. To reduce the volume of test data, several efficient test data compression techniques have been recently proposed. In this paper, we propose hybrid test compression techniques that combine the Geometric-Primitives-Based compression technique with the frequency-directed run-length (FDR) and extended frequency-directed run-length (EFDR) coding techniques. Based on experimental results, we demonstrate the effectiveness of the proposed hybrid compression techniques in increasing the test data compression ratios over those obtained by the Geometric-Primitives-Based compression technique.

1. INTRODUCTION

With today's technology, it is possible to build complete systems containing millions of transistors on a single chip. One of the major challenges in testing SOC is dealing with the large size of test data that must be stored in the tester and transferred between the tester and the chip [1]. The cost of automatic test equipment (ATE) increases significantly with the increase in their speed, channel capacity, and memory. Thus, to reduce the testing cost, the need for test data reduction becomes imperative. To achieve such reduction, several test compaction and lossless test compression schemes are used.

In test data compression, the objective is to reduce the number of bits needed to represent the test data. Several test data compression techniques have been proposed in the literature [2-6]. One of these techniques is proposed in [2] and uses what is called variable-to-block run-length coding. In this technique, a code word is used to encode a block of data based on the number of zeros followed by a one in that block. This technique is used for compressing fully specified test data that feeds a cyclical scan chain. A cyclical scan chain is used to decompress this data and transfer it to the "test scan chain". Golomb code is a variable-to-variable run-length code that is used in [3] to enhance the scheme described in [2]. It divides the runs into groups, each is of size *m*. The number of groups is determined by the length of the longest run, and the group size *m* is dependent on the distribution of test data. Another enhancement to the work done in [2] and [3] was proposed in [4]. It uses frequency-directed run-length (FDR) codes, which is another variable-to-variable coding technique. It is designed based on the observation that the frequency of runs decreases with the increase in their lengths. Hence, assigning smaller code words to runs with small length and larger code words to those with larger length could result in higher test data compression.

The techniques in [2-4] are all based on encoding only runs of 0's. This was motivated based on the idea that encoding the difference vectors instead of the actual test data may reduce the number of 1's in the encoded data. However, it was demonstrated in [4] that, in general, better test data compression results are achieved, based on both FDR and Golomb codes, by encoding the actual test data. Based on test data analysis in [6], it was observed that, in the actual test data, the frequency of runs of 1's is as significant as runs of 0's, for many of the circuits. Hence, to encode the test data based on both types of runs, extended frequency directed run-length (EFDR) codes were proposed in [6], that result in higher test data compression.

Recently, an efficient test data compression technique that gives high compression ratios, the Geometric-Primitives-Based compression technique [5], has been proposed. It is based on partitioning the test data into two-dimensional blocks and encoding each block separately based on geometric shapes. In the Geometric-Primitives-Based Compression technique, some of the blocks are encoded by storing the real test data because the encoded block size is larger than the actual test data block size. So, reducing the number of these blocks could result in higher test data compression.

In this paper, we propose hybrid test data compression techniques that exploit the use of either FDR or EFDR codes to reduce the number of blocks that are not encoded by the geometric shapes and encoded by storing the real test data. We demonstrate based on experimental results the effectiveness of the proposed hybrid compression technique in increasing the test data compression ratios over those obtained by the Geometric-Primitives-Based compression technique.

2. GEOMETRIC-BASED COMPRESSION

The Geometric-Primitives-Based Compression technique is based on encoding the 0's or the 1's in a test set by geometric shapes. In this technique, the number of shapes are limited to the basic four shaps, namely: point, line, triangle, and rectangle as shown in Table 1. For the rectangles, two points are needed to encode the shape and each point costs $2*\log_2 N$ bits, where N is the block dimension. However, lines and triangles are represented by a point and a distance *d*. Two bits are used to determine the type of line or the type of triangle encoded. In this technique, the test vectors are first sorted to generate clusters of either 0's or 1's in such a way that it may partially or totally be fitted in one or more of the geometric shapes shown in Table1. Then, the test set is partitioned into blocks each of which is NxN bits. For test vectors whose columns and/or rows are not divisible by the predetermined block dimension N, a partial block will be produced at the right end columns and/or the bottom rows of the test data. Since the size of such partial blocks can be deduced based on the number of vectors, the vector length, and the block dimension, the number of bits used to encode the coordinates of the geometric shapes can be less than $\log_2 N$. The decoder recognizes those special cases and decodes them properly.

Table 1. The used primitive geometric shapes.

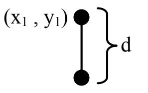
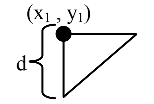
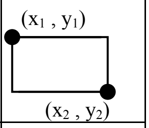
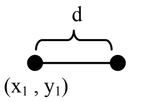
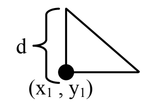
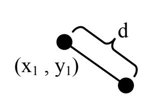
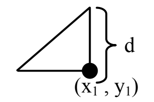
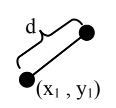
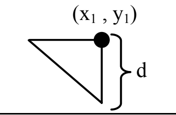
	Lines	Triangles	Rectangle
Type 1			
Type 2			**X**
Type 3			**X**
Type 4			**X**

Table 2. Geometric compression block encoding format.

Header Code	Encode Block
00	with real test data
010	as filled with 0’s
011	as filled with 1’s
10	with geometric shapes covering 0’s
11	with geometric shapes covering 1’s

The encoding process is then applied on each block independently. There are three cases that may occur:

(i) The block contains only 0’s and X’s, or 1’s and X’s. In this case, the block is encoded by the code 01 followed by the bit that fills the block.

(ii) The block needs to be encoded by a number of shapes. In this case, two bits are needed to indicate the existence of shapes and the type of bit encoded. If the encoded bit is 0, then the code is 10, otherwise it is 11.

(iii) The number of bits needed to encode the shapes is greater than the total number of bits in the block. In this case, the block is encoded by storing the real data. The real data is stored after a two-bit code (00).

The block encoding format is summarized in Table 2.

3. FDR COMPRESSION

Frequency-directed run-length (FDR) code [4] is a variable-to-variable coding technique based on encoding runs of 0’s. In FDR code, the prefix and the tail of any codeword are of equal size. In any group A_i, the prefix is of size i bits. The prefix of a group is the binary representation of the run length of the first member of the group. When moving from group A_i to group A_{i+1}, the length of the code words increase by two bits, one for the prefix and one for the tail. The FDR code for the first three groups is shown in Table 3. Since the FDR technique is based on encoding only runs of 0’s, all the X’s in a test set will be filled by 0’s to reduce the number of runs that need to be encoded.

Table 3. FDR code.

Group	Run Length	Group Prefix	Tail	Code Word
A1	0	0	0	00
	1		1	01
A2	2	10	00	1000
	3		01	1001
	4		10	1010
	5		11	1011
A3	6	110	000	110000
	7		001	110001
	……		……	……
	13		111	110111

Table 4. Extended FDR (EFDR) code.

Group	Run Length	Group Prefix	Tail	Code Word Runs of 0’s	Code Word Runs of 1’s
A1	1	0	0	000	100
	2		1	001	101
A2	3	10	00	01000	11000
	4		01	01001	11001
	5		10	01010	11010
	6		11	01011	11011
A3	7	110	000	0110000	1110000
	8		001	0110001	1110001
	……		……	……	……
	14		111	0110111	1110111

4. EXTENDED FDR (EFDR) COMPRESSION

Based on test data analysis in [6], it has been observed that test sets contain a large number of runs of 1’s as well as runs of 0’s. By encoding both types of runs, the total number of runs will decrease, which could result in higher test data compression. To encode both runs of 0’s and 1’s, the extended FDR technique [6] is used by adding an extra bit to the beginning of a code word to indicate the type of run. If the bit is 0, this indicates that the code word is encoding a run of type 0, otherwise it encodes a run of type 1. The EFDR code for the first three groups is shown in Table 4. Unlike the FDR code, the EFDR code does not have run length of size 0. This is because both runs of 0’s and 1’s are encoded. Runs of 0’s are strings of 0’s followed by a 1, while runs of 1’s are strings of 1’s followed by a 0. Since the EFDR technique encodes both types of runs, the X’s are filled with 0 except when the X’s are bounded by 1 from both sides, they are filled with 1.

5. HYBRID TEST COMPRESSION SCHEME

As it was mentioned before, in the Geometric-Primitives-Based compression technique there are some blocks which are encoded by storing the real test data. This is because the size of these blocks when they are encoded is larger than their original size. So, no compression is achieved for such blocks. In order to reduce the number of these blocks, we propose to combine the Geometric-Primitives-Based compression technique with either the FDR or the EFDR compression techniques. In this case, the

Table 5. Geometric-FDR (GFDR) & Geometric-EFDR (GEFDR) compression block encoding format.

Header Code	Encode Block
000	with real test data
001	with FDR (EFDR) codes
010	as filled with 0's
011	as filled with 1's
10	with geometric shapes covering 0's
11	with geometric shapes covering 1's

FDR or EFDR techniques are applied to encode a block. The block is encoded with these techniques if its encoding size is less than the encoding size with geometric shapes. The block encoding format for the hybrid technique combining the geometric and FDR compression techniques, called GFDR, is shown in Table 5. Note that the difference between this encoding scheme and the Geometric encoding scheme is in the header code starting with 00. So, blocks that will still be encoded with real test data will have an extra bit in the header. The other blocks have exactly the same format. The block encoding format for the hybrid technique combining the geometric and EFDR compression techniques, called GEFDR, is similar to GFDR with the difference of using EFDR instead of FDR.

Test data decompression will be done on chip and the decoded test will then be applied to the chip under test. The decoders for the proposed hybrid techniques are a direct combination of the decoders for the Geometric [5], FDR[4], and EFDR [6] techniques.

6. EXPERIMENTAL RESULTS

In order to demonstrate the effectiveness of the proposed hybrid compression schemes, we have performed experiments on a number of the largest ISCAS85 and full-scanned versions of ISCAS89 benchmark circuits. We have used the test sets generated by MinTest [7], using both static and dynamic compaction. Test sets generated by the dynamic compaction option have the letter *d* appended in their name. All the test sets used achieve 100% fault coverage of the detectable faults in each circuit. Test sets generated based on static compaction were relaxed, as this has the advantage of keeping unnecessary assignments as X's, which enables higher compression.

The test sets were partitioned into 8x8, 16x16, and 32x32 blocks, respectivly. Then, the hybrid compression schemes, GFDR and GEFDR, are applied for each case separetly. Table 6 shows the compression ratios obtained for five compression schemes namely, geometric, FDR, EFDR, GFDR, and GEFDR, respectively. The best result from the three block sizes is reported for each case.

The *compression ratio* is computed as:

$$Comp.\,Ratio = \frac{\#\,Original\ Bits - \#\,Compressed\ Bits}{\#\,Original\ Bits} X100 \quad (1)$$

As can be seen from the table, the two hybrid compression techniques, GFDR and GEFDR, both improved the compression ratio over the Geometric compression technique for all the circuits. However, the GEFDR compression scheme achieved better results and improved the compression ratio on average from 59.06% to 62.13%. Among the five compared compression schemes, the GEFDR compression scheme achieved the best results in 9 out of 14 test sets. However, the GFDR compression scheme achieved the best results in 3 out of the 14 test sets. The best compression ratio for the remaining test sets is achived by the EFDR compression technique.

Table 7 shows a detailed analysis of the number blocks encoded by the different encoding formats for the Geometric, GFDR, and GEFDR compression schems. This analysis is shown for an 8x8 block size. The second column shows the total number of encoded blocks. The third colums shows the number of blocks encoded as a block filled with either 0 or 1. The fourth and fifth columns show the number of blocks encoded by geometric shapes and those encoded by the real test data, respectively for the Geometric compression scheme. The sixth, seventh and eightth columns show the number of blocks encoded by geometric shapes, those encoded by FDR codes, and those encoded by the real test data, respectively for the GFDR compression scheme. Similarly, the last three columns show the number of blocks encoded by geometric shapes, those encoded by EFDR codes, and those encoded by the real test data, respectively for the GEFDR compression scheme. As can be seen from the table, both the GFDR and GEFDR compression schemes reduce the number of blocks encoded by the real test data and hence improve the compression ratio. For the circuits considered, the average number of real blocks is 15.16% for the Geometric compression scheme, 10.02% for the GFDR compresion scheme, and 7.37% for the GEFDR technique. Thus, the GEFDR compression technique reduces the number of real blocks by more than 50%. As indicated by the results, there is still a percentage of blocks that achieve no compression and are encoded by storing the real test data.

The average number of blocks encoded by FDR codes in the GFDR technique is 12.23% and the average number of blocks encoded by EFDR codes in the GEFDR technique is 17.5%. This indicates that these blocks achieve better compression if encoded by these codes rather than by geometric shapes, which adds to the benefit of the proposed hybrid compression schemes.

7. CONCLUSION

In this work, we have proposed two hybrid compression schemes that combine the Geometric and FDR compression schemes (GFDR), and the Geometric and EFDR compression schemes (GEFDR). The objective of these schemes is to reduce the number of blocks in the Geometric compression scheme that are encoded with the actual test data. Based on experimental results on ISCAS benchmark circuits, it has been demonstrated that the proposed hybrid compression schemes improved the test data compression ratio for all the circuits over those obtained by the Geometric compression scheme. The GEFDR technique achieved the best results and improved the compression ratio on average from 59.06% to 62.13% over the Geometric compression scheme.

ACKNOWLEDGMENT

This work is supported by King Fahd University of Petroleum & Minerals under project FT2000-07. The author would like to thank Mr. Faisal Ba Haiderah for his help in the implmentation of this work.

Table 6. Compression results of Geometric, FDR, EFDR, GFDR, and GEFDR techiques.

Circuit	Test Set Size	Geometric CR	FDR CR	EFDR CR	GFDR CR	GEFDR CR
c2670	10252	51.85	43.82	53.11	54.14	**54.56**
c5315	6586	27.88	20.47	28.64	29.03	**29.21**
s13207	163100	85.01	78.78	79.38	**85.48**	85.40
s15850	57434	60.32	56.49	56.29	**61.70**	61.43
s35932	21156	25.78	3.99	**45.63**	26.27	44.93
s38417	113152	46.50	37.66	**52.35**	48.37	51.45
s5378	20758	51.55	46.85	50.81	53.12	**53.18**
s13207d	165200	86.63	81.31	81.85	87.60	**87.74**
s15850d	76986	70.19	66.21	67.99	71.21	**71.42**
s35932d	28208	78.12	19.36	80.31	78.12	**81.71**
s38417d	164736	62.23	43.27	60.57	63.09	**65.23**
s38584d	199104	65.59	60.93	62.91	66.30	**67.03**
s5378d	23754	57.94	47.98	51.93	58.32	**58.62**
s9234d	39273	57.22	43.61	45.89	**58.39**	57.87
AVG		59.06	46.48	58.40	60.08	62.13

Table 7. Detailed analysis of block encodings for Geometric, GFDR, and GEFDR compression schemes.

Circuit	Geometric				GFDR			GEFDR		
	#Blocks	#Filled	#Shapes Encoded	#Real	#Shapes Encoded	#FDR Encoded	#Real	#Shapes Encoded	#EFDR Encoded	#Real
c2670	180	56	99	25	70	33	21	68	40	16
c5315	115	8	61	46	52	22	33	51	23	33
s13207	2640	1671	963	6	895	72	0	906	62	1
s15850	924	127	787	10	677	120	0	698	97	2
s35932	442	76	191	175	182	54	130	129	196	41
s38417	1872	252	1484	136	1214	348	58	1059	534	27
s5378	351	73	220	58	193	46	39	185	63	30
s13207d	2640	2041	531	68	487	76	36	487	87	25
s15850d	1232	614	536	82	470	110	38	481	112	25
s35932d	442	56	384	2	384	0	2	334	50	2
s38417d	2704	1068	1447	189	1290	220	126	1129	453	54
s38584d	3111	1180	1584	347	1413	284	234	1442	332	157
s5378d	378	143	157	78	142	24	69	134	43	58
s9234d	620	150	410	60	367	71	32	375	60	35
AVG(%)		28.97	55.87	15.16	48.78	12.23	10.02	46.16	17.50	7.37

REFERENCES

[1] Y. Zorian, E.J. Marinissen, and S. Dey, "Testing Embedded-Core Based System Chips," *Proc. of Int. Test Conference,* pp. 130-143, 1998.

[2] A. Jas and N.A. Touba, "Test Vector Decompression via Cyclical Scan Chains and its Application to Testing Core-Based Designs," *Proc. of Int. Test Conf.,* pp. 458-464, 1998.

[3] A. Chandra and K. Chakrabarty, "Test Data Compression for System-On-a-Chip using Golomb Codes," *Proc. of IEEE VLSI Test Symp., pp. 113-120,* 2000.

[4] A. Chandra and K. Chakrabarty, "Frequency-Directed Run-Length (FDR) Codes with Application to Systems-on-a-Chip Test Data Compression," *Proc. of IEEE VLSI Test Symp.,* pp. 42-47, 2001.

[5] A. El-Maleh, S. Al-Zahir, and E. Khan, "A Geometric-Primitives-Based Compression Scheme for Testing Systems-on-a-Chip," *Proc. of IEEE VLSI Test Symp.*, pp. 54-59, 2001.

[6] A. El-Maleh and R. Al-Abaji, "Extended Frequency-Directed Run-Length Codes with Improved Application to System-on-a-Chip Test Data Compression", *Proc. of IEEE Int. Conf. Electronics, Circuits and System, pp. 449-452, 2002.*

[7] I. Hamzaoglu and J. H. Patel, "Test Set Compaction Algorithms for Combinational Circuits'', *Proc. Int. Conf. Computer-Aided Design*, pp. 283-289, Nov. 1998.