ON IMPROVING THE EFFECTIVENESS OF SYSTEM-ON-A-CHIP TEST DATA COMPRESSION BASED ON EXTENDED FREQUENCY DIRECTED RUN-LENGTH CODES

Aiman H. El-Maleh and Raslan H. Al-Abaji

King Fahd University of Petroleum and Minerals, Dhahran 31261, Saudi Arabia

Email: {aimane, raslan}@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 test data compression techniques have been proposed. Frequency-directed run-length (FDR) code is a variable-to-variable run length code based on encoding runs of 0's. In this work, we demonstrate that higher test data compression can be achieved based on encoding both runs of 0's and 1's. We propose an extension to the FDR code and demonstrate by experimental results its effectiveness in achieving higher compression ratio.

Keywords: *System-on-a-Chip, . Frequency-directed run-length, run length.*

الملخص

تحدي هام من تحديات فحص الأنظمة على الدارة (SOC) هو التعامل مع كميات هائلة من المعلومات. لتقليل حجم معلومات الفحص، تم إقتراح عدة تقنيات لضغط المعلومات. من هذة التقنيات طريقة الضغط التي تعتمد على تردد أطوال السلاسل (FDR) وهي طرقة من متغير الى متغير، تعتمد على تشفير سلاسل الأصفار. في هذا العمل، نبين على أننا نستطيع الحصول على قدرة ضغط أعلى عندما نأخذ بعين الإعتبار سلاسل الأحاد والأصفار معاً. تم إقتراح تعديل على (FDR) و تم الإيضاح عن طريق النتائج التجريبية، على أن التعديل أثبت جدارته وفعاليته في التوصل لنسبة ضغط أعلى.

1. 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 the complexity of systems-on-a-chip continues to increase, the difficulty and cost of testing such chips is increasing rapidly [Chandramouli, 1996] [Zorian et al., 1998]. One of the challenges in testing system-on-a-chip is dealing with the large size of test data that must be stored in the tester memory and transferred between the tester and the chip under test. The cost of

automatic test equipment (ATE) increases significantly with the increase in their speed, channel capacity, and memory. Thus, to reduce the testing time and cost, it is necessary to reduce the volume of test data.

Test data reduction can be achieved by both test compaction [Schulz et al., 1988] [Pomeranz et al., 1991] [Kajihara et al., 1993] [Chang et al., 1995] [Hamzaoglu et al., 1998], and test compression [T. Yamaguchi et al., 1997] [Jas and Touba, 1998] [Jas et al. 1999] [Jas and Touba, 1999] [Chandra et al., 2000] [Chandra et al., 2001] [El-Maleh et al., 2001]. Several test data compression techniques have been proposed in the literature. In [Jas et al., 1999], statistical coding is used for encoding test data based on a modified version of Huffman coding. In [El-Maleh et al., 2001], efficient test data compression is achieved based on partitioning the test data into two-dimensional blocks and encoding each block separately based on geometric shapes. Another technique proposed in [Jas and Touba, 1998] 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 [Chandra et al., 2000] to enhance the scheme described above. 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 [Jas and Touba, 1998] and [Chandra et al., 2000] was proposed in [Chandra et al., 2001]. It uses frequency-directed run-length (FDR) code, 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 [Jas and Touba, 1998], [Chandra et al., 2000] , and [Chandra et al., 2001] 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 vectors may reduce the number of 1's in the encoded data. However, it was demonstrated in [Chandra et al., 2001] that, in general, better test data compression results are achieved, based on both FDR and Golomb codes, by encoding the actual test vectors. Based on test data analysis, we have observed that the frequency of runs of 1's is as significant as runs of 0's, for many of the circuits. This suggests that encoding both runs of 0's and 1's could result in higher test data compression. In this work, we propose an extension to the FDR codes to encode the test data based on encoding both types of runs.

2. FREQUENCY-DIRECTED RUN-LENGTH (FDR) CODE

Many of the test data compression techniques are based on run-length coding. A run is a consecutive sequence of equal symbols. A sequence of symbols can be encoded using two elements for each run; the repeating symbol and the number of times it appears in the run.

Frequency-directed run-length (FDR) code 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 that group. When moving from group A_i to group A_{i+1}, the length of the code words increases by two bits, one for the prefix and one for the tail. Runs of length i are mapped to group A_j, where $j = \lceil \log_2(i+3) \rceil - 1$. The size of the i'th group is equal to 2^i, i.e., group A_i contains 2^i members. The FDR code for the first three groups is shown in Table 1.

Table 1. 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
	8		010	110010
	9		011	110011
	10		100	110100
	11		101	110101
	12		110	110110
	13		111	110111

3. TEST DATA ANALYSIS

Based on test data analysis, it has been observed that test sets contain a large number of runs of 1's in addition to runs of 0's. By considering both types of runs, the total number of runs will decrease, which could result in higher test data compression.

To support this observation, we have analyzed test data for the largest ISCAS 85 and full-scanned versions of ISCAS 89 circuits. We have used the test sets generated by MinTest [Hamzaoglu et al., 1998], using both static and dynamic compaction. Test sets generated by 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.

Given a relaxed test set, techniques based on encoding only runs of 0's fill all the X's by 0's to reduce the number of runs that need to be encoded. However, to encode both runs of 0's and 1's in a test set, X's are filled by 1's if they are bounded by 1's from both sides, otherwise

they are filled by 0's. This results in a reduction in the total number of runs that need to be encoded. Table 2 shows the analysis of the number of runs on the used test sets. The first column indicates the circuit name. The second column shows the number of runs of 0's in the test set assuming that only runs of 0's will be encoded. The third, fourth, and fifth columns indicate the number of runs of 0's, runs of 1's, and the total number of runs, respectively, assuming that both types of runs will be encoded. As can be seen from the table, for most of the circuits, the number of runs of 1's is as significant as the number of runs of 0's. For all the circuits, the total number of runs decreases and for some circuits the reduction is significant.

Table 2. Analysis of number of runs in test data.

		Encoding 0 Runs	**Encoding 0 and 1 Runs**		
Circuit	**Original Bits**	**0 Runs**	**0 Runs**	**1 Runs**	**Total Runs**
c2670	10252	1677	505	414	919
c5315	6586	1628	561	454	1015
c7552	15111	2695	652	1111	1763
s13207	163100	4804	2615	1157	3772
s15850	57434	4635	2514	1106	3620
s35932	21156	7554	1236	1071	2307
s38417	113152	20970	5331	3761	9092
s5378	20758	2915	1072	806	1878
s9234	25939	3843	1770	980	2750
s13207d	165200	5021	2581	1210	3791
s15850d	76986	5329	2644	1202	3846
s35932d	28208	10018	235	346	581
s38417d	164736	29473	5773	4834	10607
s38584d	199104	16814	7585	4074	11659
s5378d	23754	3537	1237	1001	2238
s9234d	39273	4816	2347	1212	3559

Figures 1, 2, and 3 show the frequency of both runs of 0's and runs of 1's for test sets of the circuits: s15850, s9234, and s35932d, respectively. As can be seen from the figures, the frequency of runs of 1's follow a similar shape to that of runs of 0's, although with a smaller magnitude. For the circuit in Figure 1, it can be observed that there are more runs of 1's than 0's for run length < 5, but for run length > 5 there are more runs of 0's. For the circuit in Figure 2, we can see that runs of 0's with any length are on the average more that the runs of

1's with the same length. For the circuit in Figure 3, it can be observed that runs of 1's of small and large run length are more than those of ·'s. But for middle run length ranges, the number of both 0 and 1 runs is comparable.

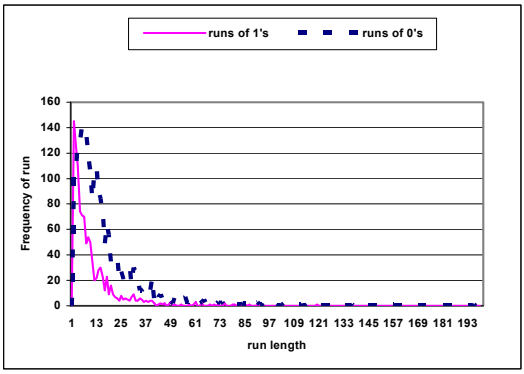

Figure 1. Distribution of runs of 0's and 1's for circuit s15850.

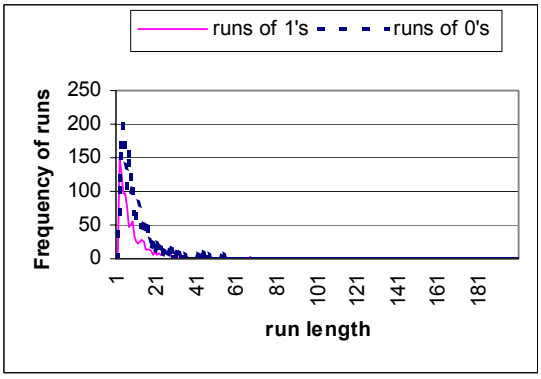

Figure 2. Distribution of runs of 0's and 1's for circuit s9234.

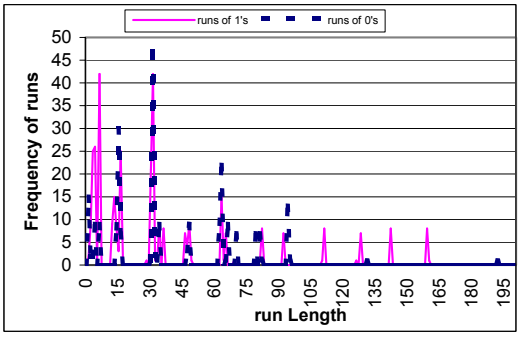

Figure 3. Distribution of runs of 0's and 1's for circuit s35932d.

Top

4. EXTENDED FDR (EFDR) CODE

To encode both runs of 0's and 1's, we extend the FDR code based on 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. This code, called Extended FDR (EFDR), is shown in Table 3. It should be observed that this code is a direct extention to the FDR code shown in Table 1. However, in this code we do not have run length of size 0. This is because we are encoding both runs of 0's and runs of 1's. Note that 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, i.e. runs of 1's of length i are the complement of runs of 0's of the same length, and vice versa. As with FDR code, in this code when moving from group A_i to group A_{i+1}, the length of code words increases by two bits, one for the prefix and one for the tail. Runs of length i are mapped to group A_j, where $j = \lceil \log_2 (i+2) \rceil - 1$. The size of the i'th group is equal to 2i+1, i.e., group Ai contains 2i+1 members.

Table 3. 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
	9		010	0110010	1110010
	10		011	0110011	1110011
	11		100	0110100	1110100
	12		101	0110101	1110101
	13		110	0110110	1110110
	14		111	0110111	1110111

To illustrate the use of this code, let us consider an example. Consider the test T={01100011111111000000001}, of size 22 bits. The number of 0 runs in this test is 10. However, the number of both 0 and 1 runs is 5. Encoding this test using FDR codes results in the encoded test TFDR={01 00 1001 00 00 00 00 00 00 110010} of size 26 bits. Thus, for this example the number of bits needed to encode the test data using FDR codes is more than the

actual size of the original test data. However, encoding this test using EFDR codes, we obtain the encoded test TEFDR={000 100 001 11011 0110000}, of size 21 bits. Obviously, for this example EFDR codes outperform FDR codes. Note that FDR codes suffer whenever we have runs of 1's, as each 1 bit will be encoded by a separate 0 run of length 0.

5. EXPERIMENTAL RESULTS

Table 4 compares the compression results using the FDR and EFDR codes. The first column shows the circuit name and the second column shows the size of the test set in bits. The third and fourth columns show the number of compressed bits using FDR and EFDR codes, respectively. The last two columns indicate the respective compression ratios. The *compression ratio* is computed as:

$$Comp.Ratio = \frac{\#Original\ Bits - \#Compressed\ Bits}{\#Original\ Bits} X100$$

Table 4. Compression results of FDR & EFDR codes.

Circuit	Original Bits	FDR Bits	EFDR Bits	FDR CR	EFDR CR
c2670	10252	5760	4807	43.82	**53.11**
c5315	6586	5238	4700	20.47	**28.64**
c7552	15111	9500	8843	37.13	**41.48**
s13207	163100	34608	33637	78.78	**79.38**
s15850	57434	24992	25105	**56.49**	56.29
s35932	21156	20312	11502	3.99	**45.63**
s38417	113152	70536	53914	37.66	**52.35**
s5378	20758	11032	10210	46.85	**50.81**
s9234	25939	16912	16127	34.80	**37.83**
s13207d	165200	30880	29992	81.31	**81.85**
s15850d	76986	26016	24643	66.21	**67.99**
s35932d	28208	22746	5554	19.36	**80.31**
s38417d	164736	93452	64962	43.27	**60.57**
s38584d	199104	77798	73853	60.93	**62.91**
s5378d	23754	12356	11419	47.98	**51.93**
s9234d	39273	22148	21250	43.61	**45.89**

As can be seen from the table, significant improvements in the compression ratio are obtained for some of the circuits. Consider for example the circuit s35932. For the first test set of this circuit, the compression ratio improves from 3.99% using FDR to 45.63% using EFDR code. For the second test set of the same circuit, the compression ratio increases from 19.36% using FDR to 80.31% using EFDR code. This result is not surprising as based on the statistics for this circuit given in Table 2, the total number of runs reduces significantly when both types of runs are used versus using only 0 runs. Similarly, significant increase in the compression ratio is obtained for the test sets c2670, c5315, s38417, and s38417d. For all the test sets except one, using EFDR codes achieve higher compression ratio. For test data decompression based on EFDR codes, the decoder design follows a direct extention of the FDR decoder proposed in [Chandra et al., 2001].

6. CONCLUSION

In this work, we have proposed an extension to the recently proposed FDR code, namely Extended FDR (EFDR) code. The proposed technique is based on encoding both runs of 0's and 1's as opposed to encoding only runs of 0's. Based on experimental results on ISCAS benchmark circuits, it has been demonstrated that the proposed EFDR code outperformed FDR code and resulted in significant increase in test data compression ratio for several circuits, reaching as large as 60% for one of the benchmark circuits.

ACKNOWLEDGMENT

This project is supported by King Fahd University of Petroleum & Minerals under project FT2000/07.

REFERENCES

1. R. Chandramouli, and S. Pateras, 1996, "Testing Systems on a Chip," IEEE Spectrum, pp. 42-47.
2. Chandra and K. Chakrabarty, 2000, "Test Data Compression for System-On-a-Chip using Golomb Codes," *Proc. of IEEE VLSI Test Symp., pp. 113-120.*
3. Chandra, A. and Chakrabarty, K., 2001, "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.
4. J. Chang and C. Lin, Nov. 1995, "Test Set Compaction for Combinational Circuits," *IEEE Trans. Computer Aided Design*, pp. 1370-1378.
5. El-Maleh, S. Al-Zahir, and E. Khan, 2001, "A Geometric-Primitives-Based Compression Scheme for Testing Systems-on-a-Chip," *Proc. of IEEE VLSI Test Symp.*, pp. 54-59.

Top

6. Hamzaoglu and J. H. Patel, 1998, “Test Set Compaction Algorithms for Combinational Circuits’’*, Proc. Int. Conf. Computer-Aided Design*, pp. 283-289.
7. Jas and N.A. Touba, 1998, “Test Vector Decompression via Cyclical Scan Chains and its Application to Testing Core-Based Designs,” *Proc. of Int. Test Conf.,* pp. 458-464.
8. Jas and N.A. Touba, 1999, “Using an Embedded Processor for Efficient Deterministic Testing of System-on-a-Chip,” *Proc. of IEEE Int. Conf. on Computer Design (ICCD).*
9. Jas, J.G. Dastidar and N.A. Touba, 1999, “Scan Vector Compression/ Decompression Using Statistical Coding,” *Proc. of IEEE VLSI Test Symp., pp. 114-120.*
10. Kajihara, S., Pomeranz, I., Kinoshita, K. and Reddy, S.M., 1993, "Cost-Effective Generation of Minimal Test Sets for Stuck-at Faults in Combinational Logic Circuits," *30th Design Automation Conf.*, pp. 102-106.
11. Pomeranz, L. Reddy, and S. Reddy, 1991, “COMPACTEST: A Method to Generate Compact Test Sets for Combinational Circuits,” *Proc. of Int. Test Conference*, pp. 194-203.
12. M. Schulz, E. Trischhler, and T. Sarfert, Jan. 1988, “SOCRATES: A Highly Efficient Automatic Test Pattern Generation System,” *IEEE Trans. Computer-Aided Design*, pp. 126-137.
13. T. Yamaguchi, M. Tilgner, M. Ishida, and D.S. Ha, 1997, “An Efficient Method for Compressing Test Data,” *Proc. of Int. Test Conf.,* pp. 191-199.
14. Y. Zorian, E.J. Marinissen, and S. Dey, 1998, “Testing Embedded-Core Based System Chips,” *Proc. of Int. Test Conf.,* pp. 130-143.

Top