GECOM: Test Data Compression Combined with All Unknown Response Masking

Youhua Shi, Nozomu Togawa, Masao Yanagisawa and Tatsuo Ohtsuki
School of Fundamental Science and Engineering
Waseda University, Tokyo, Japan 169-8555
Email: shi@yanagi.comm.waseda.ac.jp

***Abstract*—This paper introduces GECOM technology, a novel test compression method with seamless integration of test GEneration, test COmpression (i.e. integrated compression on scan stimulus and masking bits) and all unknown scan responses Masking for manufacturing test cost reduction. Unlike most of prior methods, the proposed method considers the unknown responses during ATPG procedure and selectively encodes the specified 1 or 0 bits (either 1s or 0s) in scan slices for compression while at the same time masks the unknown responses before sending them to the response compactor. The proposed GECOM technology consists of GECOM architecture and GECOM ATPG technique. In the GECOM architecture, for a circuit with N internal scan chains, only c tester channels, where $c = \lceil \log_2 N \rceil + 2$, are required. GECOM ATPG generates test patterns for the GECOM architecture thus not only the scan inputs could be efficiently compressed but also all the unknown responses would be masked. Experimental results on both benchmark circuits and real industrial designs indicated the effectiveness of the proposed GECOM technique.**

I. Introduction

Scan testing is the most powerful and widely adopted design-for-test (DFT) technique for ensuring the highest-quality during manufacturing test, nowadays which has been the foundation of most structured DFT methodologies. In spite of its effectiveness, scan testing results in excessive test data volume even for single-stuck-at faults with single-detection as the continuous increase in chip complexity [1]. Therefore, recently many researches have been conducted on test data reduction (i.e. scan stimulus compression and scan response compaction) to reduce test cost.

Scan stimulus compression techniques involve using an on-chip decompressor to decompress test vectors. The existing methods could be broadly classified into two main categories: Linear-decompression-based schemes and Non-linear lossless-code-based schemes. The first category includes techniques based on linear feedback shift register (LFSR) reseeding [2] and combinational linear expansion circuits [3]. There are also some commercial tools based on LFSR reseeding combined with on-chip decompression developed recently including Mentor Graphics' TestKompress [4], SmartBIST from IBM/Cadence [5] and Synopsys' DBIST [6]. The second class uses lossless source coding for test data reduction [18], [12].

On the other hand, scan response should be also compacted into a smaller vector using on-chip response compaction circuitry. Unfortunately, for complex designs, due to the presence tri-stated logic, floating buses, un-initialized non-scan flip-flops/latches and etc., unknown values (X's) often appear at scan chain outputs. The presence of Xs poses a major challenge for designs using test response compaction. In case of using multiple-input signature registers (MISRs) for response compaction, even just a single unknown bit in any one of the internal scan chains would invalidate the MISR signature once it is captured. Thus unlike scan stimulus compression, scan response compaction is complicated by the presence of unknown values (X's) that are captured in the scan cells.

Removal of all possible X sources by using special test pattern generation (TPG) or DFT structure such as test points insertion is not practical. To achieve the maximal observability of the internal scan chains, any response compaction scheme must be able to detect a defective chip in the presence of the residual X's that cannot be eliminated by DFT or by accurate modeling. To address this problem, many solutions have been proposed in the literature. Most of them attempt to mask the output response bits that are unknowns before compaction so as to reduce the possible impact the unknown states may have on test quality. To name a few [7], [8] discuss such X-masking techniques. However, one limitation of these methods is that these techniques mask the outputs of entire scan chains, which may overmask some non-X responses, causing test escapes. There are also some methods focused on implementing sophisticated compactor architectures in order to tolerate a given amount of X states such as i-Compact [9], X-Compact [10], and OCC [11]. In principle, all of these schemes use error-correcting codes (ECC) in one way or another to design compactors that may tolerate up to a given number of X states. Unfortunately, their problems are lack of guarantee on fault coverage and limited unknown tolerance – The problem of unknown responses still remains.

In general, test pattern generation, test compression and unknown response masking are usually treated separately as three different tasks, where scan stimulus compression and X-masking are conducted after test patterns are generated. In addition, to mask all unknown responses, masking bits compression and scan stimulus compression are usually conducted using different compression schemes. In this paper, we propose a method that performs test generation, test Compression (i.e. integrated compression on scan stimulus and masking bits) and all unknown response masking simultaneously in an integrated process completely based on the ATPG technology. A test compression and X-masking architecture is developed and integrated into the test generation process. Compared to the existing three-stage methods, the proposed technique has four significant advantages: (1) It can better utilize the ATPG tools for test compression and unknown response masking. (2) In test generation, unknown responses in the previous test vector and stimulus compression are taking into consideration when generating next test vector, thus it potentially achieves more significant compression. (3) Unknown responses are handled during test pattern generation, and then all unknown responses would be masked before entering into the response compactor, thus observable response loss due to unknown responses would be eliminated. (4) This approach could be combined with either space compactors (e.g XOR-tree) or time compactors (e.g. MISR) for scan response compaction.

It should be noted that in the literature there are some ATPG-dependent compression schemes such as [3], [4]. However they usually consider only stimulus compression during test generation rather than X-masking. Recently, a unified approach to test generation and test data reduction was presented in [17], where the ATPG tool

978-1-4244-1922-7/08/$25.00 ©2008 IEEE

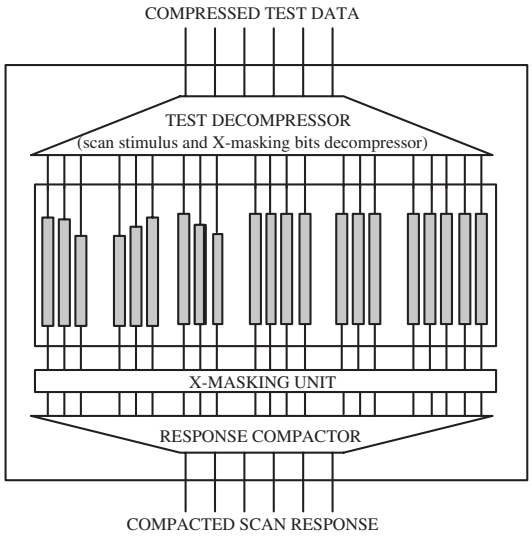


Fig. 1. GECOM Architecture

would directly handle the unknows during pattern generation and fault-simulation process. A fault would not be considered detected unless a fault effect appeared in at least one of the compactor outputs. By doing so, fault coverage loss due to unknown responses would be reduced; however the problem of unknown responses still remains. Unlike them, the proposed method considers both X-masking and stimulus compression during ATPG procedure and the masking bits are compressed together with the scan stimulus, thus it potentially achieves more significant compression and guarantees neither non-X response overmasking nor observable response loss. Here overmasking indicates the condition that the original non-X responses are masked; and observable response coverage loss means that the compacted signature is corrupted due to the existence of unknown responses.

The rest of this paper is organized as follows: Section 2 briefly overview the proposed GECOM technique. Section 3 describes the developed compression and X-masking architecture. Section 4 presents the proposed GECOM technique integrated test generation with compression and X-masking. Finally experimental analysis and conclusions are given in Section 5 and 6, respectively.

II. Basic Concept of GECOM

As shown in Fig.1, the GECOM architecture is wrapped outside the design, and consists of three main parts: a test control unit (not shown in the figure), an on-chip decompressor and a unknown masking unit. The performance of GECOM partially depends on the number of the internal scan chains and the number of the tester channels between the ATE and the CUT. The number of internal scan chains determines the number of required tester channels. The ratio of internal scan chains to tester channels usually sets the maximum compression level. As shown in Fig.1, we can use only c tester channels to drive N internal scan chains, $c = \lceil \log_2 N \rceil + 2$. The logarithmic reduction in the number of tester channels allows us to reduce the requirements on test channel bandwidth. Consequently from the tester's point of view, the design appears to have c short scan chains. In every clock cycle, c bits are applied to the GECOM decompressor inputs (one bit on each input channel), while the decompressor outputs load N scan chains and generates the corresponding unknown masking slice. Since the longest scan chain length is reduced by N/c times, theoretically in the best case we can achieve compression by a factor of N/c using only one tester clock cycle per slice, i.e. test data volume and test cycles are both reduced by N/c times.

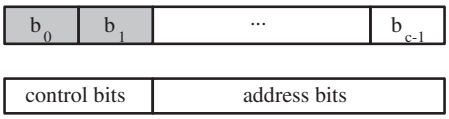


Fig. 2. Compressed Slice Form

TABLE I
Specification of Control Bits

mode	ctr. bits	description
reset mode	0 0	reset a new scan slice to all 0, and set masking bits to all 0.
	1 0	reset a new scan slice to all 1, and set masking bits to all 0.
configure mode	0 1	invert the corresponding scan-in bit as indicated by the address data only.
	1 1	invert the corresponding scan-in bit and set the corresponding masking bit to 1 as indicated by the address data.

The unknown masking unit can mask only the unknowns and leave unchanged all the other values in the scan responses before feeding them to the scan response compactor. Therefore the proposed method avoids both overmasking and observable response coverage loss. Here overmasking indicates the condition that the original non-X responses are masked; and observable response coverage loss means that the compacted signature is corrupted due to the existence of unknown responses.

III. Compression and X-Masking

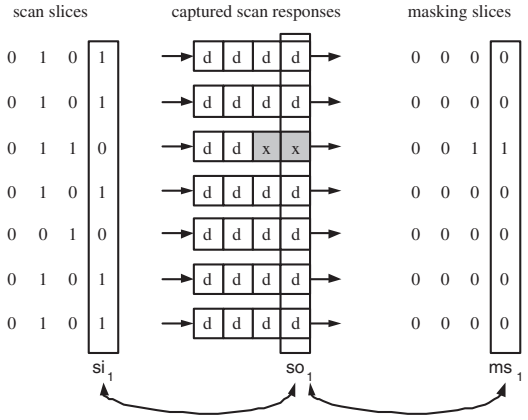
In this section we will illustrate the details of the proposed GECOM technique for compression and X-masking.

A. Test Compression

The proposed compression scheme compresses both scan stimulus and masking bits. In this compression method, each N-bit scan slice is encoded into a series of c-bit scan-in data, where $c = \lceil \log_2 N \rceil + 2$, and N is the number of internal scan chains in the CUT. As described in Section 2, the proposed approach only encodes a subset of the specified bits in a scan slice. Please keep in mind that the final compressed test data, including compressed scan stimulus and X-masking bits, are generated with the proposed test generation procedure (to be illustrated in Section 4) directly for the GECOM architecture, so that all the unknown scan responses could be actually masked before entering the response compactor.

The compressed form of a scan slice, to be called compressed slice in this paper, is shown in Fig.2, which contains the following:
(1) Control bits (2 bit): to indicate the state that the control unit need to tailor the decompressor and unknown masking unit how to work. The detail description of the control bits is shown in Table 1.
(2) Address bits ($(c-2)$ bits): to indicate the position in the scan slice that the decompressor and/or the unknown masking unit need to specify an inversion on the corresponding bit.

Example - Here let us use an example to illustrate the decompression procedure of the proposed method. Fig.3 shows the example data, where (a) shows the scan slices with the scan response captured in the previous capture cycle; and (b) describes the decompression procedure. There are 7 scan chains and 4 scan slices (si_1, si_2, si_3, si_4) with the corresponding captured scan responses (so_1, so_2, so_3, so_4). In the proposed method, instead of using 7 external test channels, only 5 are used. Look at the first slice (si_1), in the first cycle, we load compressed slice as $cs_1 = 10000$. In this paper we assume the leftmost bits in the compressed slice are the control bits,



(a)

	Compressed Slice		
	ctl. bits	adr. bits	description
s1	1 0	0 0 0	reset a new scan slice to all 1, and set the masking bits to all 0.
	0 1	0 1 1	invert the 3rd bit.
	1 1	1 0 1	invert the 5th bit and set the the corresponding masking bits to 1.
s2	0 0	0 0 0	shift the current scan slice into the scan chain, reset a new scan slice to all 0, and set the masking bits to all 0.
	0 1	0 1 1	invert the 3rd bit.
	1 1	1 0 1	invert the 5th bit and set the the corresponding masking bits to 1.
s3	1 0	0 0 0	shift the current scan slice into the scan chain, reset a new scan slice to all 0, and set the masking bits to all 0.
	0 1	0 1 1	invert the 3rd bit.
s4	0 0	0 0 0	shift the current scan slice into the scan chain, reset a new scan slice to all 0, and set the masking bits to all 0.

(b)

Fig. 3. GECOM Decompression Example

and the other bits are used to indicate address of the relative bit that needs processed as shown in Fig.2. In cs_1, the control bits are 10, which tells the control unit to reset a new scan slice to all 1s, and set the masking bits to all 0s. Thus the current scan slice is $s = 1111111$ and the masking slice is $ms = 0000000$. In the second cycle, we load a compressed slice as $cs_2 = 01011$. Thus the third bit in the scan slice is inverted, and then the current scan slice is $s = 1111011$ and the masking slice does not change value. In the next cycle, where $cs_3 = 11101$, not only the corresponding bit in the scan slice is inverted but also the corresponding unknown masking bit is set to 1. Thus the current scan slice turns out to be $s = 1101011$, and the unknown masking slice is $ms = 0010000$. Then in the next cycle, the new compressed slice arrives and the control bits are 10 (or00), which indicates the control unit to shift the current slice (s_1) into the scan chains and perform the unknown masking; while at the same time start decompressing a new slice. Here it should be mentioned that the example is just used as a representative to show the decompression process in GECOM. Because the example is too small, the proposed method does not achieve any compression; however for large designs more significant compression would be achieved as discussed in Section 5.

B. Hardware Structure

As shown in Fig.1, the GECOM architecture is wrapped outside the design, and mainly consists of three main parts: a test control unit, an on-chip decompressor and a unknown masking unit.

Control Unit – Control unit is implemented as a FSM, which is responsible for controlling the process of decompression and masking slice generation, and also controlling the scan clock for the CUT. The control unit performs different operations as specified by the control bits that it receives. Initially the control unit is in the initial state; when it receives the control bits as 00 or 10, it enters the reset mode (otherwise it enters the configuration mode, i.e. the control bits as 01 or 11) and performs a series of operations as explained in Table 1.

Decompressor – The decompressor consists of a $m-to-2^m$ decoder and a flip configuration unit (FCU). The $m-to-2^m$ decoder is used to generate the configuration signal (c) according to the address bits in the compressed slice. For example, if the address bits are 011 as shown above in the example, the configuration signal $c = 0000100$ will be generated. Since only one bit in the current scan slice can be updated at each clock cycle, multiple configuration cycles may be required to represent the original scan slice. Thus we design a flip configuration unit, which is inserted between the decoder and the internal scan chains. The FCU allows for each scan chain to be set with the initial data and updated with the configuration data. Fig.4 shows the decompressor structure for N scan chains, where s_i and so_i is the data to be shifted in and the response coming from the ith scan chain. It operates as follows: in the reset mode, $init\_en$ is set to 1 and d is the initial data for the scan slice (i.e. when the control bits are 00, d is 0; otherwise d is 1); In the configuration mode, $init\_en$ is set to 0, and c_i is the ith bit of the configuration signal c coming from the decoder. So in the configuration mode, the decompressor allows the pre-existing data in the register to be kept or inverted.

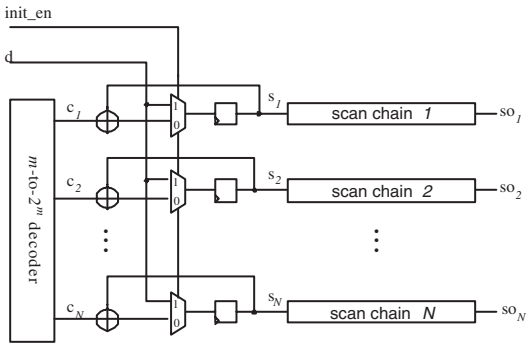


Fig. 4. Decompressor Unit

Unknown Masking Unit – The structure of the proposed unknown masking unit is shown in Fig.5. According to the control bits, if in reset mode (i.e. 00 or 10), the flip-flops in the unknown masking unit will be reset to all 0s; when in configuration mode, if the control bits are 01, the conf_en signal will be 0, so that the masking slice will be kept as it is in the previous cycle; if the control bits are 11, the $conf\_en$ signal will be set to 1 by the control unit, and according the data coming from the $m-to-2^m$ decoder, the corresponding position of the bit in the masking slice will be inverted while the other bits are changed. If the masking slice is all 0, then the scan response will pass the unknown masking unit without being masked; on the other hand, if some bit in the masking slice is 1, which means the corresponding scan response is unknown, then the response will be masked as 1. It is noted that the scan clock is controlled by the control unit, if not in the reset mode, no response from the scan chain will enter the response compactor.

To ensure its correct operation and evaluate the required hardware overhead, we implemented the proposed GECOM logic in Verilog and synthesized it using Synopsys Design Compiler. The synthesized control unit contains only two flip flops and 19 combinational gates. Since the decompressor and the unknown masking unit depend on

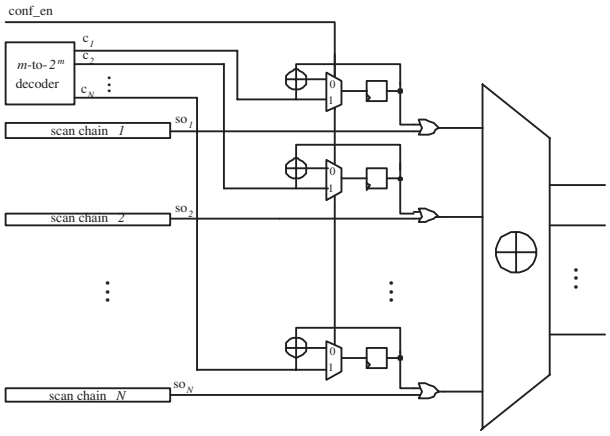


Fig. 5. Unknown Masking Unit

the number of the scan chains, we synthesized them separately. For example, in case of $N = 1000$, the synthesized circuit contains 1.6K gates and 1002 flip-flops, where we reuse the first scan cell in the CUT. For those million-gate large designs, the hardware overhead of the GECOM logic is very small, less than 0.1%.

IV. ATPG Procedure in GECOM

The most significant contributions of this work fall into two folds: (1) to integrate stimulus compression and unknown response masking with test generation, (2) to integrate scan stimulus and X-masking bits together into compression. Since we can't eliminate unknown responses, based on the low specified density of the test patterns, we can take advantage of scan stimulus to mask unknown responses during test generation. It would be desirable that unknown response masking be incorporated into test generation process in order to guarantee all unknowns masking. We present in this section a test generation process for generating compressed test data for the proposed approach; the resultant algorithm is given below.

1. Conduct ATPG to generate a fully specified test vector, run fault simulation and remove detected faults from the fault list.
2. Extract the exact positions of the Xs in the response and set constraints to the ATPG process.
   // the corresponding bits that have Xs in the response
   // should be unspecified in the next vector.
3. Conduct ATPG again for the remaining faults to obtain a partially specified test vector.
4. Count the number of specified 0s and 1s in each slice. If $(p(0) < p(1))$, then the unspecified bits with unknown responses in the previous vector are assigned 0s, and the other unspecified bits are assigned 1s; and vise versa.
5. Perform fault simulation and drop all detected faults from the fault list.
6. If undetected faults remains, go to Step 2.

Fig. 6. Test Generation Process in GECOM

At the beginning of test generation, a fully specified vector is generated for maximize the fault detection. In each iteration, when a new test cube is generated, the presence of Xs in the previous captured response is taken into consideration. All the Xs in the previous response need a fixed value in the corresponding position of the next scan stimulus, which should be filled with 0s or 1s based on compression analysis. In our work, we have studied the specified bits information and found that, in addition to low specified bit density, there exist other useful scan test properties. Specifically, in each scan slice, the number of the minority specified 1 or 0 bits (either 1 or 0) is definitely to be less than half of the total number of specified bits. Thus we selectively fill the corresponding bits, which are unknown in the previous response, with specified 1s or 0s and also fill the left unspecified bits as inverted values. For example, if the number of specified 1s is less than that of 0s in the current test cube, the bit positions that have unknown responses in the previous vector are filled with 1s and the other unspecified bits are filled with 0s. After that, fault simulation is conducted and the faults detected by this generated test vector are dropped from the fault list. This flow continues until no faults remains in the fault list.

Regarding output response compaction, typically only a fraction of the scan cells would produce an unknown state. Therefore, we can easily set constraints to the ATPG process to generate satisfied test cubes. In the proposed approach, the ATPG tool would directly handle the unknowns during it pattern-generation and fault-simulation processes, and furthermore the X-masking bits are compressed together with the scan stimulus, thus no more additional test channels are required and observable response loss due to unknown response and non-X response overmasking would be eliminated. In addition, any commercial ATPG tools can support the proposed test generation process, except that in the generation of each vector, the Xs in the previous response should be carefully filled according to the compression scheme. As compared to other ATPG-dependent compression methods [4], the proposed integrated scheme has the following advantages which make it more efficient in achieving significant compression. First, it is not necessary to solve a set of linear equations to find the compressibility between the decompressor and the test cubes generated from ATPG. It only involves a procedure of specified 0s and 1s counting and this has a neglectable computation overhead. Second, when some test cubes can not be compressed by the decompressor, ATPGs of conventional approaches have to iteratively try or change configuration of decompressors. For the proposed method, this problem does not exist and it is easy for the decompressor to apply any test cube and not necessary to change any configuration even with fully specified vectors. Third, the conventional ATPG-dependent methods fill unspecified bits only targeting scan stimulus compression, and the fillings basically don't consider X-masking, where the masking bits should be compressed in another stage. For the proposed method, not only the scan stimulus but also the masking bits are compressed together, which makes it potentially achieve more significant compression while guarantees no observable response loss.

V. Experimental Results

In this section, we present experimental results for four largest ISCAS'89 benchmark circuits and three ASICs to validate the effectiveness of the proposed GECOM technique.

Table 2 and 3 lists the number of flip-flops (N_{ff}), the number of scan chains (N_{sc}), the number of test patterns (N_v) of the ATPG results for both benchmark circuits and ASICs. Test patterns are generated by a commercial ATPG tool using the proposed test generation algorithm. In Table 3, the relative fault coverage ratio (F_g/F_a), and the relative run time (T_g/T_a) are also presented, where 'g' indicates the proposed method and 'a' refers the regular test pattern generation using dynamic and static compaction where the unspecified stimulus are randomly filled. As the ISCAS'89 benchmark circuits don't have X-sources such as tri-state buses or multiple clock domains, they don't produce X values at the outputs. Consequently, to imitate the situation where unknown values exist throughout the output responses, we injected unknown values randomly into the responses of the CUTs (i.e. a specified response is replaced by a unknown bit)

TABLE II
TEST GENERATION RESULTS FOR BENCHMARKS

Circuits	N_{ff}	N_{sc}	N_g
s13207	669	50	231
		100	257
		200	271
s15850	597	50	157
		100	170
		200	168
s38417	1636	50	198
		100	220
		200	239
s38584	1452	50	287
		100	299
		200	297

TABLE III
TEST GENERATION RESULTS FOR ASICS

Circuits	N_{ff}	N_{sc}	N_g	N_a	F_g/F_a	T_g/T_a	V_g/V_a
ASIC 1	8K	400	728	658	1.0	1.08	1.11
ASIC 2	20K	400	1735	1459	0.99	1.12	1.19
ASIC 3	40K	500	3362	2984	1.0	1.15	1.13

and the unknown response density (P_u) is set to be 0.5%, which is consistent with the numbers drawn from the industry practice. As can be seen in Table 3, for large designs, when compared to the regular ATPG, GECOM takes 8-15% more run time and achieves almost the same fault coverage except ASIC 2. The difference in fault coverage can be attributed to the different unspecified fills during the GECOM and the regular ATPG runs. And it is obvious that the number of vectors would slightly increase in GECOM, on average using GECOM test data size is increased by 10%.

Table 4 presents the results on the compression efficiency of the proposed GECOM technique for the seven circuits. In the table, the name of circuits, the number of scan chains, the number of required external test channels (c), the number of GECOM vectors and the size of uncompressed test data (T_D) are shown in the first five columns, respectively. Because we focus on test cost reduction for large designs with a great number of scan chains, thus we set the number of scan chains to be 50, 100 and 200 as representatives for the benchmark circuits and 400 or 500 for the ASICs. The compressed results for the proposed GECOM scheme (T_E) and the compression ratio (C_r) are also listed, where both the size of uncompressed test data (T_D) and the size the compressed test data (T_E) include the size of X-masking bits for all unknown response masking. The compression ratio of the proposed method is listed in the last column - they range from 74.9% to 96.9%. In general, significant compression, up to 32X, could be obtained even the circuits with a lot of X-resources.

Table 5 shows a comparison of the results on the ISCAS'89 benchmark circuits for the proposed method with CircularScan [13], Illinois Scan [14], and Dictionary Coding with Correction (DCC) [15], which are the representatives of the recently introduced test data compression schemes. The result of the Mintest ATPG-compacted test sets [16] is also listed for the sake of comparison. The result listed for the proposed method is the minimum size shown in boldface in Table 4. Since different ATPG tools may be used, Table 5 shows both the number of vectors and the total number of compressed bits for each case. It should be noted that the results of the proposed method include both the compressed scan stimulus and the masking bits, while the other methods only contain the compressed scan stimulus. As can be seen from the table, the results obtained using the proposed GECOM scheme are better than those of CircularScan [13] and Illinois Scan [14] for all the circuits. We should note that these previous methods can only be applied to test data compression, while our method can also be used to mask unknown responses, the details

TABLE IV
COMPRESSION RESULTS OF GECOM

circuits	N_{sc}	N_c	N_g	T_D	T_E	C_r
s13207	50	8	231	309,078	77,616	74.9%
	100	9	257	343,866	48,573	85.9%
	200	10	271	362,598	**43,360**	88%
s15850	50	8	157	187,458	30,144	83.9%
	100	9	170	202,980	27,540	86.4%
	200	10	168	200,592	**25,200**	87.4%
s38417	50	8	198	647,856	104,544	83.9%
	100	9	220	719,840	100,980	86.0%
	200	10	239	782,008	**64,530**	91.7%
s38584	50	8	287	833,448	137,760	83.5%
	100	9	299	868,296	80,730	90.7%
	200	10	297	862,488	**71,280**	91.7%
ASIC 1	400	11	728	11.7M	676K	94.2%
ASIC 2	400	11	1735	71M	2.2M	96.9%
ASIC 3	500	11	3362	270M	10.9M	96.0%

TABLE VI
COMPARISON WITH ATPG-DEPENDENT COMPRESSION METHOD [3]

Circuits	N_{sc}	SCC [3]		Proposed (stimulus + masking bits)	
		vectors	T_E	vectors	T_E
s13207	200	178	22,784	271	43,360
s15850	200	264	25,344	168	25,200
s38417	200	312	89,856	239	64,530
s38584	200	203	38,976	297	71,280

of which will be shown in the follows. More important is that when compared with real industrial designs, the used benchmark circuits are very smaller in size and have more overlapped logic cones, which will limit the compression efficiency of the proposed GECOM technique as we can see in Table 4 where we can achieve more significant compression ratios in large ASICs than those in benchmarks.

Table 6 compares the results of the proposed GECOM scheme with one of the published ATPG-dependent method SCC [3]. GECOM has better compression results on s15850 and s38417, and larger final compressed data volume on s13207 and s38584 than SCC [3]. However, it should be mentioned that GECOM targets not only on scan stimulus compression but also on all unknown response masking. In GECOM, the unknown response density (P_u) of the four benchmark circuits is set to be 0.5%; and the final compressed results of GECOM contains both the compressed scan stimulus and the compressed masking bits. Even so, the compression results obtained by GECOM are still comparable with those of SCC.

The second set of the experimental results is on the unknown response masking. As mentioned above, the ISCAS'89 benchmark circuits don't have X-sources, so we injected unknown values randomly into the responses of the CUTs and the unknown response density (P_u) is set to be 0.5%. In addition, three ASIC designs we used are with different x-management, where the unknown response densities ranges from 0.02% to 0.8%.

In Table 7, we show the test quality comparison with and without using GECOM masking. As a representative, here we use XOR based compactor, and the compaction ratio is set to 50X, which means one channel will receive an XOR result of 50 scan chains. Table 7 compares the number of unobservable responses and computes the observable responses loss with and without using GECOM masking. As the results show, using the proposed GECOM technique can guarantee no overmasking and maximized the observable responses as directly observe every scan-out response.

Finally, Table 8 shows the overall comparisons between the proposed GECOM and other ATPG-dependent compression techniques. The proposed GECOM technique presented in this paper is unique for the following reasons: (1) It is capable of integrating stimulus

TABLE V
COMPARISON WITH PREVIOUS WORKS

Circuits	Mintest [16]		Circular Scan [13]		Illinois Scan [14]		DCC [15]		Proposed (stimulus + masking bits)	
	vectors	T_E	vectors	T_E	vectors	T_E	vectors	T_E	vectors	T_E
s13207	233	163,100	299	62,415	236	82,546	236	31,772	271	43,360
s15850	94	57,434	186	62,408	126	76,030	126	27,721	168	25,200
s38417	68	113,152	270	250,016	99	129,732	99	84,896	239	64,530
s38584	110	161,040	251	162,909	136	129,580	136	65,396	297	71,280

TABLE VII
TEST QUALITY COMPARISON

CUT	wo GECOM masking		GECOM masking	
	U.O. Res	Obs. Loss(%)	U.O. Res	Obs. Loss(%)
s13207	933	18	0	0
s15850	908	16.99	0	0
s38417	3778	18.99	0	0
s38584	2770	16.1	0	0
ASIC 1	5460	4.5	0	0
ASIC 2	204730	29.5	0	0
ASIC 3	29585	1.0	0	0

TABLE VIII
OVERALL COMPARISON BETWEEN THE PROPOSED GECOM AND OTHER ATPG-DEPENDENT COMPRESSION METHODS

	ATPG-dependent Compression	GECOM
Traditional ATPG reusable	No	Yes
Fault Coverage Loss	Maybe	No
Integrated Compression on Stimulus and masking bits	No	Yes
Computation Overhead	High	Neglectable
All-X Masking	No	Yes
Encode/Decode Complexity	High	Low
Compression Efficiency	High	High

compression and unknown response masking with test generation, (2) It enables integrated compression on scan stimulus and X-masking bits, (3) It introduces a novel architecture that cause neither non-X responses overmasking nor observable response loss.

Techniques such as those presented in [9], [10], [11] using error correcting codes (ECC) to tolerate up to a given number of X states can also be used in conjunction with the proposed GECOM technique presented in this paper to further the effectiveness of response compaction technique. One of our future works will be concentrated on development of error-tolerate response compactor.

VI. CONCLUSIONS

This paper presents an integrated test approach, called GECOM, to test **G**E**n**eration, test **CO**mpression (i.e. integrated compression on scan stimulus and masking bits) and all unknown scan response **M**asking for manufacturing test cost reduction. The proposed scheme directly generates compressed scan stimulus and masking bits for a specific DFT architecture in an integrated ATPG process. In contrast to existing techniques, the proposed technique could potentially achieve more significant test data reduction and cause neither overmasking not observable response coverage loss. It can be used together with space compactors (e.g. X-Compactor) or time compactors (e.g. MISR) even if the total number of Xs exceeds the number of Xs that can be tolerated by the response compactor.

ACKNOWLEDGEMENTS

This research was supported by Grants-in-Aid for Young Scientists (B) (No.19700045) and "Ambient SoC Global COE Program of Waseda University" of the Ministry of Education, Culture, Sports, Science and Technology, Japan.

REFERENCES

[1] L.Wang, X.Wen, H.Furukawa and S.Wu, "VirtualScan: A New Compressed Scan Technology for Test Cost Reduction," in *Proceedings IEEE International Test Conference (ITC)*, pp. 916–925, Oct. 2004.

[2] S.Hellebrand, J.Rajski and el, "Built-in test for circuits with scan based on reseeding of multiple-polynomial linear feedback shift registers," *IEEE Transactions on Computers*, vol. 44, no. 22, pp. 223–233, Feb. 1995.

[3] I.Bayraktaroglu and A.Orailoglu, "Test volume and application time reduction through scan chain concealment," in *Proceedings ACM/IEEE Design Automation Conference (DAC)*, pp. 151–155, June 2001.

[4] J.Rajski, J.Tyszer, M.Kassab and N.Mukherjee, "Embedded Deterministic Test," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 23, no. 5, pp. 776–792, May 2004.

[5] B.Koenemann et al, "A smartBIST variant with guaranteed encoding," in *Proceedings IEEE Asian Test Symposium (ATS)*, pp. 325–330, Nov. 2001.

[6] S.P.Wohl, J.Waicukauski and M.Amin, "Efficient Compression and Application of Deterministic Patterns in a Logic BIST architecture," in *Proceedings ACM/IEEE Design Automation Conference (DAC)*, pp. 566–569, June 2003.

[7] V.Chickermane, B.Foutz and B.Keller, "Channel Masking Synthesis for Efficient On-Chip Test Compression," in *Proceedings IEEE International Test Conference (ITC)*, pp. 452–461, Oct. 2004.

[8] S.Mitra, S.S.Lumetta and M.Mitzenmacher, "X-tolerant signature analysis," in *Proceedings IEEE International Test Conference (ITC)*, pp. 432–441, Oct. 2004.

[9] J.Patel, S.Lumetta, and S.Reddy, "Application of Saluja-Karpovsky Compactors to Test Responses with Many Unknowns," in *Proceedings IEEE VLSI Test Symposium (VTS)*, pp. 107–112, May 2003.

[10] S.Mitra and K.Kim, "X-Compact: An Efficient Response Compaction Technique," *IEEE Transactions on Computer-Aided Design*, vol. 23, no. 3, pp. 421–432, Mar. 2004.

[11] P.Wohl and L.Huisman, "Analysis and Design of Optimal Combinational Compactors," in *Proceedings IEEE VLSI Test Symposium (VTS)*, pp. 101–106, May 2003.

[12] Y.Shi, N.Togawa, S.Kimura, M.Yanagisawa and T.Ohtsuki, "FCSCAN: An Efficient Multiscan-based Data Compression Technique for Test Cost Reduction," in *Proceedings IEEE Asia and South Pacific Design Automation Conference (ASP-DAC)*, pp. 653–658, Jan 2006.

[13] B.Arslan and A.Orailoglu, "CircularScan A Scan Architecture for Test Cost Reduction," in *Proceedings Design, Automation, and Test in Europe (DATE)*, pp. 1290–1295, Mar. 2004.

[14] M.Tehranipour, M.Nourani and K.Chakrabarty, "Nine-coded Compression Technique with Application to Reduced Pin-count Testing and Flexible On-chip Decompression," in *Proceedings Design, Automation, and Test in Europe (DATE)*, pp. 1284–1289, Mar. 2004.

[15] A.Wurtenberger, C.S.Tautermann and S.Hellebrand, "Data Compression for Multiple Scan Chains Using Dictionaries with Corrections," in *Proceedings IEEE International Test Conference (ITC)*, pp. 926–935, Oct. 2004.

[16] I.Hamzaoglu and J.H.Patel, "Test set compaction algorithms for combinational circuits," in *Proceedings IEEE International Test Conference (ITC)*, pp. 283–289, Oct. 1998.

[17] Y.Lin, and K.Cheng, "A Unified Approach to Test Generation and Test Data Volume Reduction," in *Proceedings IEEE International Test Conference (ITC)*, pp. 1–10, Oct. 2006.

[18] N.A. Touba, "Survey of Test Vector Compression Techniques," *IEEE Design and Test of Computers*, vol. 23, no. 4, pp. 294–303, Jul. 2006.