IEICE TRANS. INF. & SYST., VOL.E90–D, NO.1 JANUARY 2007

PAPER

Low-Cost IP Core Test Using Tri-Template-Based Codes

Gang ZENG[†a)], ***Nonmember and*** **Hideo ITO**[††], ***Fellow***

SUMMARY A tri-template-based codes (TTBC) method is proposed to reduce test cost of intellectual property (IP) cores. In order to reduce test data volume (TDV), the approach utilizes three templates, i.e., all 0, all 1, and the previously applied test data, for generating the subsequent test data by flipping the inconsistent bits. The approach employs a small number of test channels I to supply a large number of internal scan chains $2^I - 3$ such that it can achieve significant reduction in test application time (TAT). Furthermore, as a non-intrusive and automatic test pattern generation (ATPG) independent solution, the approach is suitable for IP core testing because it requires neither redesign of the core under test (CUT) nor running any additional ATPG for the encoding procedure. In addition, the decoder has low hardware overhead, and its design is independent of the CUT and the given test set. Theoretical analysis and experimental results for ISCAS 89 benchmark circuits have proven the efficiency of the proposed approach.

key words: *design for testability, IP core testing, test cost reduction, test data compression*

1. Introduction

A system-on-a-chip (SOC) typically contains various predesigned and prevalidated embedded intellectual property (IP) cores provided by different core vendors. These embedded cores can be classified as hard, firm, and soft cores [1]. With the increase in the cost of manufacturing test of SOC, new design for testability (DFT) techniques that address the test problem from a low-cost point of view are required [2], [3]. In addition, in the SOC paradigm, the system integrator has only knowledge about the core functionality but not about its implementation details. Thus, there is need for a non-intrusive DFT test technique that is independent of automatic test pattern generation (ATPG). That is, the test solution does not require the redesign of the core under test (CUT) so that it can speed up the time to market and imposes the minimum performance penalties on the CUT [5]. Meanwhile, it does not require the netlist of the CUT for running the fault simulation and ATPG, thus it is applicable to various IP cores test [22].

DFT technology has matured over the years with scan design becoming common to most design flows. The cost of scan testing is measured by test data volume (TDV) and test application time (TAT). The time taken to apply a scan test pattern is dominated by the shift clock cycles (SCC) required to shift it into the scan chain. To reduce the SCC, a general approach taken is to create more parallel scan chains. However, the number of available I/O pins on the chip limits the number of parallel scan chains in the CUT. Therefore, new DFT technique is required to reduce the TAT of scan testing for the embedded IP core.

To reduce test cost, one direction to explore is the fact that only a small number of bits in a test set generated by ATPG are specified. As reported in [2], [6], the percentage of specified bits in test sets for large industry circuits was typically in the range of 1% to 5%, even with compaction during ATPG. By taking advantage of this feature of test set, a number of DFT techniques have been proposed to reduce test cost, which can be broadly classified into the following four categories:

(1) Traditional encoding method. This method exploits the redundant data or the unbalanced frequencies of various patterns in an unspecified test set by using the traditional lossless compression methods such as Huffman codes [7], Golomb codes [8], FDR codes [9] etc. An attractive feature with this technique is that it is a non-intrusive and ATPG independent technique such that it is applicable to IP core testing. However, this technique cannot reduce the actual SCC because the decoded test vector has the same length as the original test vector. Moreover, because its decoder is typically designed with one scan input, this technique cannot support multiple scan inputs to further reduce TAT.

(2) Fan-out scan chain design method. In this method, a single scan input is used to fan out to multiple internal scan chains. The examples using fan-out scan chain include Illinois scan [15], fan-out scan chain with feedback [16], and scan tree [17], etc. The Illinois scan and scan tree typically require reordering scan cells according to their compatible relation such that these compatible scan cells can receive the same test data when scanning in test vectors. While these methods can achieve significant test cost reduction at the minimum hardware overhead, reordering scan cells is very expensive because it requires changing the place and route information of the scan cells and may change the timing of the circuit [18]. As reported in [15] and [17], when using the default scan chain configuration or considering the layout constraint, Illinois scan chain and scan tree only achieved average 39% and 32% reduction in test cost, respectively.

(3) Seed encoding method. This method tries to encode a test cube into a seed by solving a set of linear equations. During test, the seed is decompressed by using a linear decompressor which may be a sequential linear feedback shift

Manuscript received January 11, 2006.
Manuscript revised June 30, 2006.
[†]The author is with the Graduate School of Information Science, Nagoya University, Nagoya-shi, 464–8603 Japan.
[††]The author is with the Faculty of Engineering, Chiba University, Chiba-shi, 263–8522 Japan.
a) E-mail: sogo@ertl.jp
DOI: 10.1093/ietisy/e90–d.1.288

Copyright © 2007 The Institute of Electronics, Information and Communication Engineers

register (LFSR) [12], ring generator [10] or a combinational XOR network [13]. While this method can achieve the most effective reduction in test cost, it requires more DFT efforts including more hardware overhead, additional ATPG and fault simulation procedure. As mentioned in [11], to successfully encode a test cube into a seed, the size of the LFSR should be 20 more than the number of specified bits in the test cube. For this reason, the method may encounter the problem that the given test cube has too many specified bits to be encoded into a seed when using a predefined linear decompressor [14]. Therefore, to effectively apply this method, special ATPG algorithms have to be incorporated into the conventional test generation procedure to control the number of specified bits in the generated test cube.

(4) Template-based encoding method. Unlike the traditional encoding method, this method does not need the synchronization and handshaking between the decoder and automatic test equipment (ATE). Moreover, it can effectively reduce TAT by employing multiple scan inputs. The basic idea is to generate the expected test data based on a given template by only flipping the inconsistent bits between them. The employed templates include the previously applied test data [23] and the captured response of previously applied test pattern [24], [25]. In [23], I internal inputs are utilized to support 2^I internal scan chains by using a general I to 2^I decoder. In addition, an I-bit shift register is used to serially input test data to the internal I inputs through one external input. While this method achieved significant reduction in TDV and TAT by exploring the capability of shifting in the overlapped encoded data, it cannot employ multiple external inputs to further reduce TAT. Although the test architecture of [24] can use I external inputs to support 2^{I-1} internal scan chains, the method cannot achieve effective reduction in TDV because the randomness of test response leads to too many flipping operations. To improve the effectiveness of [24], the broadcast ability and the reconfigurable feature were added to the scan architecture in [25] where I inputs were employed to support $2^{I-1} - 2$ internal scan chains. However, this method is intrusive because it required replacing the standard scan cell with a modified one. Moreover, an additional test generation procedure has to be used to obtain effective compression in [24], [25] because of the employed response-based template. As a result, these methods cannot be applied to the IP cores that their structural information is not known. Recently, a coding method using dictionaries with corrections [22] was proposed where test data is generated by using dictionary entry as template and flipping the inconsistent bit. However, because the number of flipping bits for generating each test data is limited to one, the method has to use a large dictionary with many entries to generate all test data, thus resulting in large hardware overhead for the construction of on-chip dictionary.

Typically, the challenge in the coding-based approaches lies in the tradeoff between the hardware complexity of the decoder and the compression performance. Generally, the more complexity (i.e., more hardware overhead) the decoder is, the more compression it can obtain. In this paper, a tri-template-based codes (TTBC) method is proposed for IP core testing. Its objective is to achieve significant reduction in IP test cost at the expense of a simple decoder. In comparison with previous template-based encoding method [22]–[25], the approach has the following distinct features. First, it employs three templates (i.e., all 0, all 1, and previously applied test data) for the encoding but not one template, which improves the flexibility and efficiency of encoding. Second, it does not use test response as template and does not require any modification to the CUT, thus eliminating the need for additional fault simulation and test generation, and making it applicable to IP core testing. Third, it does not need dictionary to store the used templates, thus leading to low hardware overhead. Finally, the proposed test architecture can use I ($I > 2$) inputs to support $2^I - 3$ internal scan chains, which is greater than $2^{I-1} - 2$ [25] or 2^{I-1} [24]. Therefore, it can more effectively use the tester channels for test cost reduction. Note that recently, a similar idea was presented in [26] for test data compression, in which all 0, all 1, and previous slice data were employed as initial test data, and then the inconsistent bits were inverted using configuration data. However, there are some differences in comparison with the proposed one here. First, while [26] directly encoded test data based on each slice, the proposed approach encodes test data using a heuristic algorithm based on continuous slice stream, thus improving the encoded efficiency as described in Sect. 3. Second, while [26] employed a dedicated decoder and a flip configuration network composed of D flip-flop, MUX, and XOR gate for decompression, the proposed method utilizes a simpler hardware which only consists of a general N to 2^N decoder and T flip-flops. Finally, [26] achieved dramatic compression by combining its method with scan chain clustering technique. It has been demonstrated in [27] that even merely applying scan chain clustering can achieve as high as average 89% test data compression. This compression is obtained by employing a fan-out architecture, in which a small number of external inputs is used to supply test data for a great number of short internal scan chains. However, this architecture is test set dependent, and the requirements for short scan chain and large routing overhead for the crossing lines between input and output limit its application in practice.

The rest of this paper is organized as follows. Section 2 presents an overview of the proposed test architecture. Section 3 presents the proposed TTBC method and the design of TTBC decoder. Section 4 gives a theoretical analysis of test cost reduction. In Sect. 5 the experimental results are provided. Finally, Sect. 6 concludes this paper.

2. Overview of the Proposed Test Architecture

As shown in Fig. 1, the proposed test architecture mainly consists of an on-chip decoder for decompression of test data and an output response analyzer (ORA) for compression of test response. The number of inputs to the decoder is I, and the largest number of outputs from the decoder is

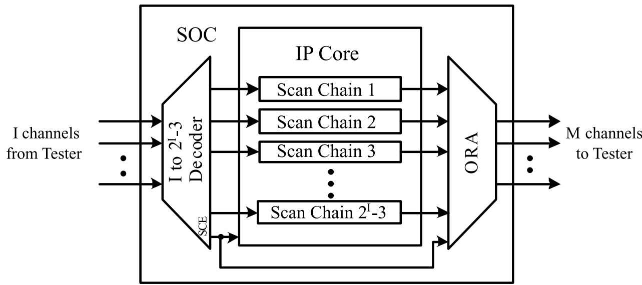


Fig. 1 Proposed test architecture.

$2^I - 3$, which indicates that the approach can use I input test channels to supply at most $2^I - 3$ internal scan chains. In addition, an augmented control signal, shift clock enable (SCE), is used to control the shift operation of scan chains and ORA. As suggested in [5], the incorporated DFT circuit is separate from the CUT to achieve the non-intrusive implementation. The test flow of the approach is as follows:

(1) According to the actual number of scan chains (S) in IP core, the corresponding number of input test channels (I) satisfying $\log_2(S + 3) \leq I < \log_2(S + 3) + 1$ can be solved. Then the I to S decoder is determined.
(2) Use the proposed TTBC method for encoding the precomputed test set.
(3) Store the encoded test set in the memory of ATE.
(4) Test the IP core using I ATE channels and an on-chip decoder.

From the above test flow, it is clear that the approach does not require the redesign of CUT except for the design of the decoder. Moreover, the approach does not need any structural information (e.g., netlist) of the IP core for the encoding procedure. Therefore, the approach is applicable to IP core testing and is suitable for SOC system integrator.

Although this paper intends to compress test stimulus data, the test response compression should also be considered to achieve true reduction in TAT. As for ORA, a commonly time-based compactor, i.e., MISR can be used as done in many papers. However, it is important to note that an X-tolerant ORA should be utilized to obtain a complete non-intrusive test strategy [5] since the conventional MISR cannot tolerate X values in test response [4]. To this end, a space-based X-tolerant compactor [19] can be employed. In this case, the number of output channels will be determined by the expected test quality and the expected compression rate of test responses.

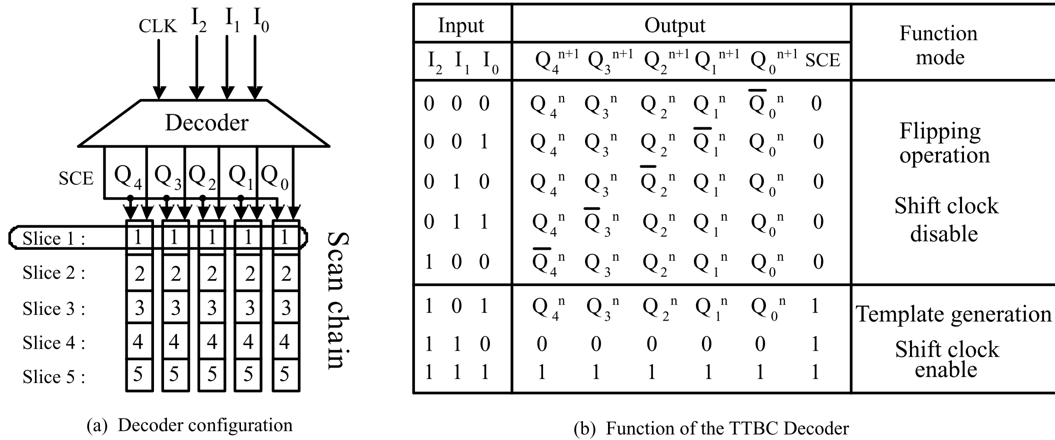
3. TTBC Method and the Design of the Decoder

The proposed TTBC method also exploits the unspecified feature of a given test cube set because these unspecified bits can be assigned to arbitrary logic value so as to achieve a large amount of test compression. To generate a slice data (i.e., test data which corresponds to the same bit position for multiple scan chains), the TTBC first generates a template in one clock, and then only the bits of the template which have the conflict values with the expected slice will be flipped. The employed templates may be all 0, all 1, and previous slice data. Therefore, for each slice data there are three possible templates to generate it. The objective for selecting template is to generate the expected slice from the template with the minimal number of flipping bits. However, it is not easy to select the optimal template for each slice because the selection for this slice may have effect on the subsequent slices, and the total possible selections for n slices are 3^n. To simplify the computation, a heuristic algorithm is proposed for selecting the optimal template for each slice. It is assumed that the template for previous slice has been determined. In this case, the algorithm determines the optimal template for current slice by considering the current slice and its subsequent slice. The optimal template is the one that can generate the current slice and the subsequent slice with the minimal number of flipping bits. The algorithm involves the following steps.

(1) Select one type of template and count the number of flipping bits for generating the current slice.
(2) Calculate the required minimal number of flipping bits for generating the subsequent slice by considering all three possible templates.
(3) Repeat step 1 until all three templates have been tried for generation of the current slice.
(4) Select one of three templates as the optimal template for the current slice, which can generate the two adjoining slices with minimal number of flipping bits.

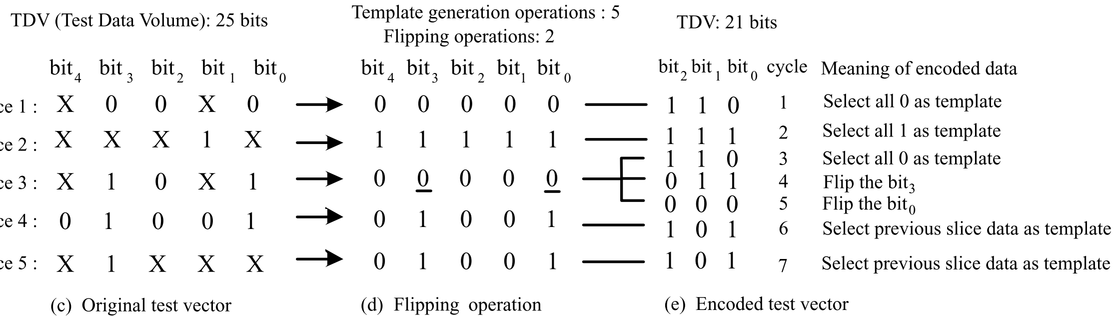
As can be seen, by only considering two adjoining slices the computational complexity has been reduced to $9n - 6$, where n represents the total number of slices needed to be generated.

To illustrate the proposed encoding method and the design of corresponding decoder, consider a simple example. Figure 2 (a) shows an IP core with 5 scan chains. As can be seen in Fig. 2 (c), to apply test to this core, one test vector with 25 bits is split into 5 slices which have 5-bit width corresponding to the 5 scan chains. First, the TTBC method is conducted and the obtained results are shown in Fig. 2 (d). As can be seen, for slice 1 the care bits are all 0s, all 0 is therefore selected as template to generate this slice. Similarly, all 1 is selected as template to generate the second slice. For slice 3, there are two bits 1 and one bit 0, if all 1 is selected as template, then only the bit 2 needs to be flipped, otherwise if all 0 or the previous second slice is selected as template then two bits i.e., bit 0 and bit 3 are needed to be flipped to generate the third slice. When only consider generating this slice, all 1 is the optimal template. However, when consider the effect of selected template on the generation of subsequent slice, the all 0 is the optimal template. This is because when selecting all 0 as template for slice 3, the slice 4 can be generated by using previous slice without any flipping bit; otherwise the slice 4 will need at least two flipping bits to generate it. As a result, the selection of template by considering two adjoining slices can save one flipping bit than that by considering only one slice.



Input $I_2\ I_1\ I_0$	Q_4^{n+1}	Q_3^{n+1}	Q_2^{n+1}	Q_1^{n+1}	Q_0^{n+1}	SCE	Function mode
0 0 0	Q_4^n	Q_3^n	Q_2^n	Q_1^n	$\overline{Q}_0^n$	0	Flipping operation Shift clock disable
0 0 1	Q_4^n	Q_3^n	Q_2^n	$\overline{Q}_1^n$	Q_0^n	0	
0 1 0	Q_4^n	Q_3^n	$\overline{Q}_2^n$	Q_1^n	Q_0^n	0	
0 1 1	Q_4^n	$\overline{Q}_3^n$	Q_2^n	Q_1^n	Q_0^n	0	
1 0 0	$\overline{Q}_4^n$	Q_3^n	Q_2^n	Q_1^n	Q_0^n	0	
1 0 1	Q_4^n	Q_3^n	Q_2^n	Q_1^n	Q_0^n	1	Template generation Shift clock enable
1 1 0	0	0	0	0	0	1	
1 1 1	1	1	1	1	1	1	

(a) Decoder configuration (b) Function of the TTBC Decoder



(c) Original test vector (d) Flipping operation (e) Encoded test vector

Fig. 2 An example for test set encoding and decoder design.

After the above operation, the obtained results are needed to be encoded in such a way that the original test vector can be restored by using a simple decoder. The idea is to represent each slice using two types of codewords. That is, one represents the selected template, and the other represents the bit position needed to be flipped. As mentioned earlier, the proposed test architecture uses I scan inputs to support the largest $2^I - 3$ scan chains but not 2^I scan chains. The reason is that the foremost $2^I - 3$ states of 2^I can be used to represent the bit positions needed to be flipped, and the last 3 states can be used to represent the selected templates. As for the above example, the codewords from "000" to "100" represent flipping the bit 0 to bit 4, respectively. The last three codewords, "101", "110", and "111" represent selecting "previous slice data", "all 0", and "all 1" as template, respectively. Finally, the encoded test vector is shown in Fig. 2 (e). As can be seen, the original test vector with 25 bits can be compressed to 21 bits and only needs 7 cycles to shift in the vector when using three scan inputs. The 7 cycles consist of 5 cycles for generating 5 templates, and 2 cycles for 2 flipped bits. It can be seen that the total number of cycles required by the proposed method is equal to the sum of the total number of test slices and the total number of flipping bits.

Note that a basic principle for TTBC is to generate one template for one slice. Even though the needed number of flipping bits for one slice varies with different slices, the required template is one. Moreover, the encoded data for each slice always begins with a selected template, and then is followed by needed flipping bits. Therefore, the beginning of a template indicates that the previous slice data has been loaded completely. By taking advantage of this fact, the generation of a template for the current slice can be done with the loading of previous slice into scan chain simultaneously. Furthermore, the number of generated templates can be utilized to distinguish one test vector from the continuous inputted test data because the required number of slices for one vector is fixed. As a result, without use of any external pin or additional data, the encoded template data and a simple counter can be utilized to generate the "load" and "capture" signal on chip, which is different from [23] in which an additional test channel has to be used to inform the scan chain when to load the decoded data into it.

To decompress the encoded data, a decoder is required to insert between the test channels and internal scan chains. The design of the TTBC decoder is similar to that of conventional I to 2^I synchronous decoder. Consider the given example in Fig. 2. The corresponding function table for this 3 to 5 TTBC decoder is listed in Fig. 2 (b). As can be seen, the decoder has two distinct function modes, i.e., flipping mode and template generation mode. The former is employed to flip the content of the flip-flops in the decoder at specified bit position. The latter is used to generate a template for the current slice, and at the same time load the content of the decoder (i.e., previously decoded slice data) into the scan chain in parallel. Note that only in the template generation mode, the decompressed slice data can be shifted into internal scan chains and the captured test responses can be shifted into

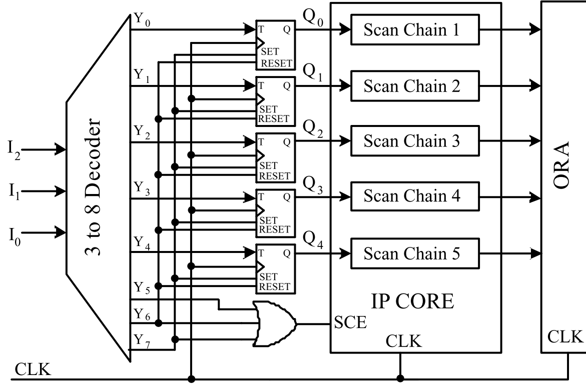


Fig. 3 A design example of the decoder.

Input			Output	Operation
T	RESET	SET	Q^{n+1}	
X	0	1	1	Set
X	1	0	0	Reset
0	0	0	Q^n	Hold
1	0	0	$\overline{Q^n}$	Toggle

Fig. 4 Characteristic table of employed T flip-flops in Fig. 3.

ORA. In contrast, in the flipping mode, the shift clock for all scan chains and ORA is disabling. To this end, an additional SCE signal is generated by the decoder to indicate whether shift in the test data to scan chains or not, as suggested in [20]. Because the flipping data and template data are represented by different codewords, the decoder can automatically generate this control signal based on the inputted data. This is different from the externally generated control signal as used in [20].

A design example of the decoder corresponding to the circuit in Fig. 2 is given in Fig. 3. In this example, the 3 to 5 TTBC decoder is composed of a general 3 to 8 decoder with high active output, 5 T flip-flops and an OR gate. The characteristic table of employed T flip-flop is given in Fig. 4. The number of T flip-flops is equal to the number of internal scan chains. For the sake of simplicity, the capture control logic is not shown in Fig. 3. In fact, a counter can be utilized for this purpose by counting the number of clock cycles when SCE is high level (i.e., the number of loaded templates). For the example as shown in Fig. 2, the counter should be set to 5, that is, when 5 slice data has been loaded into scan chain, the scan chain should take a transition from scan mode to capture mode to apply the completely loaded test vector to the circuit, and capture the test response into scan chain. After that, when next slice data is shifted into scan chain, the captured test response can be shifted out to ORA simultaneously.

For the design of any I to $2^I - 3$ TTBC decoder, the corresponding function table of the decoder is the same as that of the 3 to 5 decoder as shown in Fig. 2 (b) except that the number of input and output is different. It is important to note that the design of the decoder is independent of the CUT and the given test set, which is only determined by the number of internal scan chains. Therefore, the approach is capable of providing sufficient flexibility to the SOC system integrator.

4. Analysis of the Lower Bound of Reduced Test Cost

The theoretical reduction in TDV and SCC achieved by the approach is presented first. As discussed in Sect. 1, the TAT of scan testing is dominated by SCC. Therefore, only the theoretical reduction in SCC is given in this section to simplify the analysis. In normal scan design, the TDV and SCC of scan testing is determined by the number of test patterns (T), the number of scan cells (N), and the number of scan inputs (C). For the calculation, it is assumed that the scan chains are balanced and one scan input is used for one scan chain, then the TDV and SCC of normal scan design, can be calculated by the following equations:

$$TDV_{scan} = T \times N \tag{1}$$

$$SCC_{scan} = (N \times T)/C. \tag{2}$$

In the proposed approach, the TDV and SCC of the approach is determined by the number of external inputs (I), the number of internal scan chains (S), and the density of specified bits (D) as well as their distribution. As discussed in Sect. 3, the TDV of the approach consists of $TDV_{templatebit}$ and $TDV_{flipbit}$, which corresponds to the template selection data and flipping position data, respectively. The $TDV_{templatebit}$ is equal to $I \times (N/S) \times T$ due to the rule of one template for one slice. The $TDV_{flipbit}$ is dominated by the required number of flipping bits for each slice. To simplify the discussion, the effect of selecting previous slice as template is ignored, and only all 0 or all 1 is used as template for this analysis. Therefore, if the same probability (i.e., 50%) of occurring the specified bit 1 or 0 in one slice is assumed, then the required total number of flipping bits for the test set is $D \times N \times T/2$ after selecting all 0 or all 1 as template. Consequently, the theoretical TDV and SCC of the approach can be obtained by the following equations:

$$TDV_{proposal} = I \times (N/S) \times T + D \times N \times T \times I/2 \tag{3}$$

$$SCC_{proposal} = TDV_{proposal}/I = (N \times T)/S + (D \times N \times T)/2. \tag{4}$$

Let $S = 2^I - 3$ and assume the normal scan and the approach using the same number of external inputs, i.e. let $C = I$ in the above equations, then the percentage compression of TDV and SCC with the approach can be obtained by the following equation:

$$\textit{Percentage compression of } TDV \textit{ and } SCC = 1 - (TDV_{proposal}/TDV_{scan}) = 1 - (I/2^I - 3) - D \times I/2. \tag{5}$$

From Eq. (5), it can be seen that the achievable compression of TDV and SCC with the approach is dependent on I and D when $S = 2^I - 3$ is assumed. Figure 5 shows the

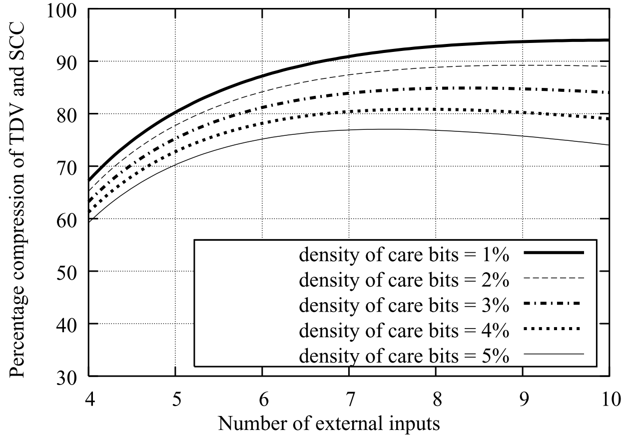


Fig. 5 Theoretical percentage compression of TDV and SCC under the condition of using the same number of external inputs for both normal scan and the proposed approach.

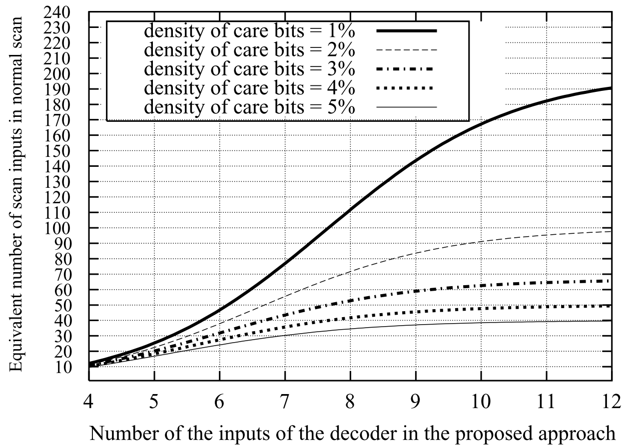


Fig. 6 Relation between the equivalent number of inputs in normal scan and the number of inputs in the proposed approach under the condition of achieving the same test time.

theoretical compression of TDV and SCC using the above Eq. (5). It can be observed that with the decrease in density of specified bits, the obtained compression increases. For example, when density of specified bits is 1%, the approach can obtain over 90% compression using 7 external inputs.

To evaluate the theoretical reduction in test input pins, let $SCC_{scan} = SCC_{proposal}$ and $S = 2^I - 3$ in the above equations, then the following equation can be obtained.

$$C_{equivalent} = 2(2^I - 3)/(2 + D \times (2^I - 3)) \quad (6)$$

where $C_{equivalent}$ represents the equivalent number of inputs in normal scan, which can achieve the same test time as obtained by using I inputs to the decoder in the proposed approach. Figure 6 shows the relation between the $C_{equivalent}$ in the normal scan and the I inputs in the proposed approach using the above Eq. (6). As can be seen, using 10 inputs in the proposed approach can achieve the same test time as obtained by using about 170 scan inputs in the normal scan when the density of care bits is 1%. In other words, the above Figs. 5 and 6 indicate that the approach can achieve significant reduction in test cost by using a small number of ATE channels.

Table 1 Characteristics of MINTEST sets for ISCAS 89 benchmark circuits.

Circuit name	No. of scan cells	No. of test vectors	No. of total bits in test set	Density of specified bit (%)
S5378	214	111	23,754	27.4
S9234	247	159	39,273	27.0
S13207	700	236	165,200	6.8
S15850	611	126	76,986	16.4
S38417	1664	99	164,736	31.9
S38584	1464	136	199,104	17.7

It is important to note that because the actual distribution of specified bit 1 and 0 in each slice is not uniform, the required total number of flipping bits will be less than $D \times N \times T/2$. Moreover, selecting the previous slice data as template can further reduce the required number of flipping bits. Therefore, the above theoretical analysis only gives the lower bound of reduced test cost, and the approach can be expected to achieve more reduction in test cost than that of the calculation using Eq. (5), which has been validated by the experimental results in Sect. 5.

5. Experimental Results

The approach is used to compress test sets for the large six ISCAS 89 benchmark circuits with full scan design. The test sets used for experiment are the compacted test cubes generated by MINTEST [21]. Since the size of benchmark circuits are small and the test sets have been generated with dynamic and static compaction, the density of specified bits of MINTEST sets is much higher than that of general industry circuits as shown in Table 1. In this experiment, only the results of reduced TDV are provided since the approach can achieve the similar reduction in both TDV and TAT as shown in Sect. 3. Table 2 shows the percentage compression of TDV with varying number of external inputs. The best compression results for each benchmark circuit are denoted with bold font. The results show that the proposed approach can achieve average 53% compression of TDV for benchmark circuits.

Figure 7 illustrates the relation between the obtained compression and the density of specified bit in test set. This figure shows the similar tendency to that of theoretical analysis. That is, the smaller the density of specified bits is, the larger the obtained compression can be. For example, while compressing test set with 27% density of specified bits can achieve 41.4% compression, compressing test set with 6.8% density of specified bits achieves 78.2% compression. An exception in Fig. 7 is the result of s38417. This is because using previous slice as template has greatly improved the result of s38417 by reducing the required number of flipping bits. It also can be seen from Fig. 7 that the actually achieved compression is better than that of theoretical analysis as pointed out in Sect. 4. Furthermore, as mentioned in Sect. 1, for test set of large industry circuit, the density

Table 2 Percentage compression of TDV with varying number of inputs.

Circuit	No. of original bits	Compression results: I = No. of inputs to the decoder (S = No. of scan chains)							
		$I = 4$ ($S = 13$)		$I = 5$ ($S = 29$)		$I = 6$ ($S = 61$)		$I = 7$ ($S = 125$)	
		Bits	Comp.	Bits	Comp.	Bits	Comp.	Bits	Comp.
S5378	23754	**14240**	**40.1**	15030	36.7	16872	28.9	17969	24.4
S9234	39273	**23028**	**41.4**	23705	39.6	26532	32.4	27160	30.8
S13207	165200	60068	63.6	43350	73.8	37050	77.6	**36078**	**78.2**
S15850	76986	35164	54.3	**31950**	**58.5**	32592	57.7	34993	54.5
S38417	164736	92012	44.1	**87850**	**46.7**	93918	43.0	102214	38.0
S38584	199104	95780	51.9	**88070**	**55.8**	92424	53.6	103474	48.0

Table 3 Comparing test data compression with previously published results.

Circuits	Template using previous data			Circular Scan [24]			Reconfigurable Scan [25]			The proposed approach		
	#Vec.	#Bits	#In	#Vec.	#Bits	#In	#Vec.	#Bits	#In	#Vec.	#Bits	#In
s13207	236	41230	7	299	62409	9	299	**27404**	8	236	36078	7
s15850	126	40795	5	186	62405	8	186	32424	8	126	**31950**	5
s38417	99	95520	5	270	250013	8	270	174479	8	99	**87850**	5
s38584	136	113140	5	251	162910	9	251	**56490**	9	136	88070	5

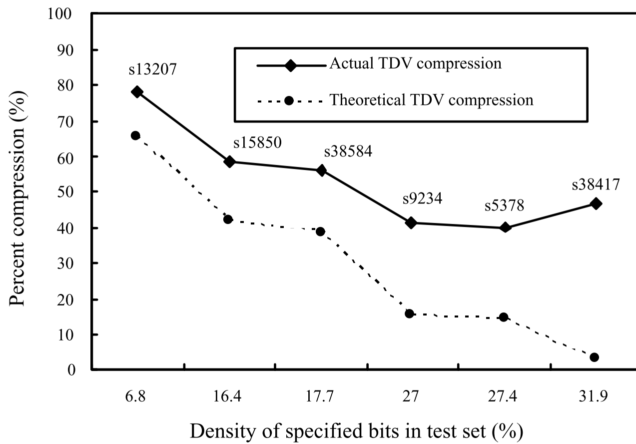


Fig. 7 Obtained compression vs. the density of specified bits in test set when using 5 scan inputs.

of specified bits is typically lower than that of benchmark circuits. Therefore, it can be further predicted that the proposed approach can achieve higher test cost reduction for the large industry circuit.

To validate the effectiveness of the proposed approach, previous template-based techniques [23]–[25] are selected for comparison since all these methods encode test data using template and have the similar test architecture as that of the proposed approach. Note that simply comparing the percentage compression is unfair because they use different test sets with different density of care bits. As discussed in Sect. 4, the lower the density, the higher the compression is. Therefore, the optimal compression results in terms of bit count of the compressed test set are selected for comparison. In addition, the serial-to-parallel shift registers used in [23] is deleted and only the effect of using previous slice data as template is evaluated because the test architecture of [23] required additional synchronous signal and cannot support multiple inputs to further reduce TAT, which is different from those in [24], [25] and this paper. As done in [24], [25], only the results corresponding to four larger benchmark circuits are given in Table 3. As can be seen from the table where the best results have been denoted with bold font, the proposed approach is superior to the method using previous slice data as template and [24] for all circuits, and it achieve comparable results to [25]. Note that the proposed approach achieves these results with fewer inputs than both [24] and [25]. Moreover, the approach is non-intrusive contrary to [25] as discussed in Sect. 1.

The experimental results have clearly demonstrated that the approach can effectively reduce the test cost of IP core by directly applying the TTBC methods on a precomputed test set. At the same time, the hardware overhead of the employed decoder is small, which is similar to the conventional I to 2^I synchronous decoder as shown in Sect. 3.

6. Conclusion

A TTBC method has been proposed for the IP core test. The proposed TTBC method is simple but effective for reducing test cost. Furthermore, as a non-intrusive and ATPG independent solution, the approach is suitable for IP core testing because it requires neither modification of the CUT nor running any additional ATPG or fault simulation for the encoding procedure. In addition, the decoder has low hardware overhead, and its design is independent of the CUT and the given test set. The experimental results for the ISCAS 89 benchmark circuits show that the approach achieves average 53% reduction in test cost. The experimental results and theoretical analysis further predicts higher test cost reduction for large industry circuits with lower density of specified bits in the test set.

Acknowledgments

The work of the first author is supported by a research grant from the Marubun Research Promotion Foundation of Japan. The authors would like to thank Lei Li and Dr. Krishnendu Chakrabarty from Duke University for providing the MINTEST sets.

References

[1] Y. Zorian, E.J. Marinissen, and S. Dey, "Testing embedded core based system chips," Proc. Int. Test Conf., pp.130–143, 1998.
[2] J. Bedsole, R. Raina, A. Crouch, and M.S. Abadir, "Very low cost testers: Opportunities and challenges," IEEE Des. Test Comput., vol.18, no.5, pp.60–69, Sept.-Oct. 2001.
[3] J. Rajski, "DFT for high-quality low cost manufacturing test," Proc. Asian Test Symposium, pp.3–8, 2001.
[4] C. Barnhart, V. Brunkhorst, F. Distler, O. Farnsworth, B. Keller, and B. Koenemann, "OPMISR: The foundation for compressed ATPG vectors," Proc. Int. Test Conf. (ITC), pp.748–757, 2001.
[5] G. Zeng and H. Ito, "Non-intrusive test compression for SOC using embedded FPGA core," Proc. IEEE Int. Symposium on Defect and Fault Tolerance In VLSI Systems, pp.420–428, 2004.
[6] H. Vranken, F. Hapke, S. Rogge, D. Chindamo, and E. Volkerink, "ATPG padding and ATE vector repeat per port for reducing test data volume," Proc. Int. Test Conf., pp.1069–1078, 2003.
[7] A. Jas, J.G. Dastidar, M.E. Ng, and N.A. Touba, "An efficient test vector compression scheme using selective Huffman coding," IEEE Trans. Comput.-Aided Des., vol.22, no.6, pp.797–806, June 2003.
[8] A. Chandra and K. Chakrabarty, "Test data compression for system-on-a-chip using Golomb codes," Proc. VLSI Test Symposium (VTS), pp.113–120, 2000.
[9] A. Chandra and K. Chakrabarty, "Test data compression and test resource partitioning for system-on-a-chip using frequency-directed run-length (FDR) codes," IEEE Trans. Comput., vol.52, no.8, pp.1076–1088, Aug. 2003.
[10] J. Rajski, J. Tyszer, M. Kassab, N. Mukherjee, R. Thompson, K. Tsai, A. Hertwig, N. Tamarapalli, G. Mrugalski, G. Eide, and J. Qian, "Embedded deterministic test for low cost manufacturing test," Proc. Int. Test Conf., pp.301–310, 2002.
[11] B. Koenemann, "LFSR-coded test patterns for scan designs," Proc. European Test Conference, pp.237–242, 1991.
[12] A. Jas, B. Pouya, and N.A. Touba, "Virtual scan chains: A means for reducing scan length in cores," Proc. Int. Test Conf., pp.73–78, 2000.
[13] I. Bayraktaroglu and A. Orailoglu, "Test volume and applicatioin time reduction through scan chain concealment," Proc. Design Automation Conf., pp.151–155, 2001.
[14] K.J. Balakrishnan and N.A. Touba, "Improving encoding efficiency for linear decompressors using scan inversion," Proc. Int. Test Conf. (ITC), pp.936–944, 2004.
[15] I. Hamzaoglu and J.H. Patel, "Reducing test application time for full scan embedded cores," Proc. Int. symposium on fault Tolerant Computing, pp.260–267, 1999.
[16] N. Oh, R. Kapur, T.W. Williams, and J. Sproch, "Test pattern compression using prelude vectors in fan-out scan chain with feedback architecture," Proc. Design, Automation and Test in Europe, pp.10110–10115, 2003.
[17] K. Miyase and S. Kajihara, "Optimal scan tree construction with test vector modification for test compression," Proc. Asian Test Symposium (ATS), pp.136–141, 2003.
[18] S. Makar, "A layout-based approach for ordering scan chain flip-flops," Proc. Int. Test Conf., pp.341–347, 1998.
[19] S. Mitra and K.S. Kim, "X-Compact an efficient response compaction technique for test cost reduction," Proc. Int. Test Conf., pp.311–320, 2002.
[20] B. Koenemann, C. Barnhart, B. Keller, T. Snethen, O. Farnsworth, and D. Wheater, "A SmartBIST variant with guaranteed encoding," Proc. Asian Test Symposium (ATS), pp.325–330, 2001.
[21] I. Hamzaoglu and J.H. Patel, "Test set compaction algorithms for combinational circuits," Proc. Int. Conf. Computer-Aided Design, pp.283–289, 1998.
[22] A. Wurtenberger, C.S. Tautermann, and S. Hellebrand, "Data compression for multiple scan chains using dictionaries with corrections," Proc. Int. Test Conf., pp.926–935, 2004.
[23] S. Reda and A. Orailoglu, "Reducing test application time through test data mutation encoding," Proc. Design, Automation and Test in Europe (DATE), pp.387–393, 2002.
[24] B. Arslan and A. Orailoglu, "CircularScan: A scan architecture for test cost reduction," Proc. Design, Automation and Test in Europe (DATE), pp.1290–1295, 2004.
[25] B. Arslan and A. Orailoglu, "Test cost reduction through a reconfigurable scan architecture," Proc. Int. Test Conf., pp.945–952, 2004.
[26] Y. Shi, N. Togawa, S. Kimura, M. Yanagisawa, and T. Ohtsuki, "Selective low-care coding: A means for test data compression in circuits with multiple scan chains," IEICE Trans. Fundamentals, vol.E89-A, no.4, pp.996–1003, April 2006.
[27] W. Rao, A. Orailoglu, and G. Su, "Frugal linear network-based test decompression for drastic test cost reduction," Proc. Int. Conf. on Computer-Aided Design (ICCAD), pp.721–725, 2004.

Gang Zeng graduated from Hunan University, Changsha, Hunan Province, China, with B.E., M.S. degrees in 1993, 2001, respectively. From 1993 to 1998, he joined Hunan University where he was a lecturer in the Institute of Electric and Information Engineering. From 2001 to 2002, he joined ZTE Corporation where he was a senior engineer. He received the Dr. Eng. degree in information science from Chiba University in 2006. He is currently a postdoctoral researcher in the Graduate School of Information Science of Nagoya University. His research interests include design for testability, built-in self-test, system-on-a-chip testing, and embedded systems low power design etc. He is a student member of the IEEE.

Hideo Ito was born in Chiba, Japan, on June 1, 1946. He received the B.E. degree from Chiba University in 1969 and the D.E. degree from Tokyo Institute of Technology in 1984. He joined Nippon Electric Co. Ltd. in 1969 and Kisarazu Technical College in 1971. Since 1973, he has been a member of Chiba University. He is currently a Professor of Department of Information and Image Sciences. His research interests include easily testable design, test generation, VLSI architecture, fault-tolerant design of parallel & distributed systems, and defect-tolerant VLSI design. He is a member of the IEEE and the IPSJ.