PAPER

Extended Selective Encoding of Scan Slices for Reducing Test Data and Test Power

Jun LIU[†,††a)], ***Student Member***, **Yinhe HAN**[†], ***Member***, ***and*** **Xiaowei LI**[†], ***Nonmember***

SUMMARY Test data volume and test power are two major concerns when testing modern large circuits. Recently, selective encoding of scan slices is proposed to compress test data. This encoding technique, unlike many other compression techniques encoding all the bits, only encodes the target-symbol by specifying a single bit index and copying group data. In this paper, we propose an extended selective encoding which presents two new techniques to optimize this method: a flexible grouping strategy, X bits exploitation and filling strategy. Flexible grouping strategy can decrease the number of groups which need to be encoded and improve test data compression ratio. X bits exploitation and filling strategy can exploit a large number of don't care bits to reduce testing power with no compression ratio loss. Experimental results show that the proposed technique needs less test data storage volume and reduces average weighted switching activity by 25.6% and peak weighted switching activity by 9.68% during scan shift compared to selective encoding.
key words: *selective encoding, test data compression, test power reduction, flexible grouping, X-filling*

1. Introduction

The increasing number of IP cores integrated in system-on-chip(SoC) has caused an exponential growth in test data volume. Large test data volume has been a major factor contributing to high test cost. Test data compression is proven to be an efficient method to reduce test data volume. Many compression techniques, such as selective Huffman coding [1], run-length coding[2],[3], test compression technique based on multiple scan chains[4]–[9] and deterministic BIST [10],[11], have been proposed. A detailed survey of these techniques is presented in [12]. Besides test data volume, test power is another concern during testing. Circuits have more switching activities in test mode than in function mode, which results in more power consumption. The increasing average power and peak power will have an adverse impact on the reliability of circuits and cause chip malfunction. Many techniques, such as scan chains modification [13] and reorder [16], power-aware ATPG [14], and X-filling[15], have been proposed to reduce test power.

However, by considering these two metrics at the same time, we can find that an effective compression technique may not be efficient in reducing test power. [17]–[21] focus on this problem and develop techniques to reduce test data and test power simultaneously. Some of these works usually considered a tradeoff between compression ratio and test power because test data compression techniques and test power reduction techniques based on X-filling utilize the same X bits in test cubes. Therefore, reducing test power will result in compression ratio loss. For example, [20] firstly fills all don't care bits by X-filling technique, such as 0-filling, then selective encoding technique[4] is used to compress test data. The method results in more test data storage volume compared to [4].

However, it is possible to reduce test power with no compression ratio loss. The proposed technique, extended selective encoding of scan slices, can achieve the goal. The proposed technique can further reduce test data storage through new grouping strategy compared to selective encoding[4]. Moreover, with no compression ratio loss, a large number of don't care bits are exploited to reduce test power by grouping merging technique and identifying some special scan slices.

The rest of the paper is organized as follows: Section 2 introduces the selective encoding of scan slices. The proposed technique is shown in Sect. 3. Section 4 is the hardware decompression architecture. Experimental results are shown in Sect. 5 and section 6 concludes this paper.

2. Selective Encoding

To help understand the proposed technique, we introduce the selective encoding technique[4] firstly.The test structure of selective encoding is shown in Fig. 1, where the N-bits test data loaded into N scan cells in parallel(as grey color marked) is called a scan slice.The c-bits data from ATE channels is called slice-code in this paper.

In selective encoding technique, a N-bits scan slice is encoded to one or several c-bits slice-codes, where $c = k+2$, $k = \lceil \log_2^{(N+1)} \rceil$. Several examples of selective encoding are

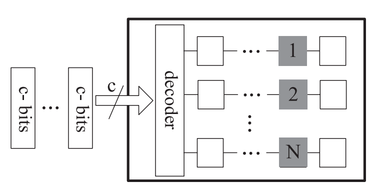


Fig. 1 Multiple scan chains test.

Manuscript received November 17, 2009.
Manuscript revised March 1, 2010.
[†]The authors are with the Key Laboratory of Computer System and Architecture, Institute of Computing Technology, Chinese Academy of Sciences and Graduate University of Chinese Academy of Sciences, China.
[††]The author is with the School of computer and information, Hefei University of Technology, China.
a) E-mail: liujun@ict.ac.cn
DOI: 10.1587/transinf.E93.D.2223

Table 1 Example of selective encoding.

No.	Scan slice	Slice codes	Description
1	XXXX 0XXX XXXX XXX	00 1111	Start a new slice, map all X to 0 and no bits are set to 1
2	XXX1 11XX X0XX 0XX	01 1001	Start a new slice, map all X to 1 and bit 9 is set to 0
		10 1100	Set bit 12 to 0
3	X111 XX00 XX01 X00	00 1011	Start a new slice, map all X to 0 and bit 11 is set to 1
		11 0000	Enter group copy mode from bit 0
		11 X111	The test data is X111
4	0XX0 00XX 1111 111	01 1111	Start a new slice, map all X to 1 and no bit is set to 0
		11 0000	Enter group copy mode from bit 0
		11 0XX0	The test data is 0XX0
		11 00XX	The test data is 00XX

illustrated in Table 1, in which there are 4 scan slices with $N = 15$ and $k = 4$. The slice-code in the third column consists of two parts: control-code and data-code. The first two bits are control-code and the following k bits are data-code.

The encoding method firstly determines the target-symbol of a scan slice by counting the number of value 0 and 1. If the number of value 0(1) is larger than the number of value 1(0), the ***target-symbol*** is 1(0) and all X bits are mapped to ***default value*** 0(1). The target-symbol is complementary to the default value. Then a scan slice is divided into several groups from bit 0. Each group contains k bits data (possible exception of the last group) and the starting position of a group is $0, k, \cdots, (\lceil N/k \rceil - 1)k$. To present the selective encoding easily, here we define three kinds of group:

1. M-type group: a group with two or more target-symbols.
2. S-type group: a group containing one target-symbol.
3. N-type group: a group without target-symbol.

Selective encoding only encodes S-type group and M-type group. S-type group is encoded by specifying target-symbol index and this is called single bit mode. M-type group data is directly copied into slice-codes and this is called group copy mode. Control-code 00(01) is the sign of a new slice and indicates the target-symbol is 1(0). Control-code 00,01,10 is the prefix of single bit mode and the k-bit data-code following them specify one target-symbol position in scan slice. Control-code 11 signifies the starting of group copy mode. Two slice-codes are needed to encode a M-type group. The k-bits data-code of the first slice-code specifies the starting position of the M-type group and the data-code of the second slice-code is the actual test data as slice 3 shows. Multiple consecutive M-type groups can be merged and only one data-code is used to indicate the starting position of these groups as slice 4 shows.

It's noted that the group copy mode should be interleaved by single bit mode. When the number of M-type group (multiple consecutive M-type groups are counted as one M-type group) is larger than the number of S-type group, additional dummy slice-codes (such as slice 1 and slice 4 in Table 1) need to be padded. The number of dummy slice-codes is the difference between the number of M-type groups and S-type groups.

3. Proposed Technique

3.1 Flexible Grouping Strategy

Selective encoding is effective in reducing test data because it only encodes S-type groups and M-type groups. We also find that M-type groups, especially consecutive M-type groups, are more efficient in reducing test data. The following is an extreme example. For a scan slice with given m target-symbols, if the m target-symbols are distributed in m S-type groups, it needs m slice-codes to encode the scan slice. However, if the m target-symbols are distributed in $\lceil m/k \rceil$ consecutive M-type groups, it only needs $\lceil m/k \rceil + 1$ slice-codes, where k is group length.Group length means the number of data bits contained in a group.

The above analysis motivates us to develop a new grouping strategy for further reducing test data. The new grouping strategy aims at reducing the number of S-type and M-type groups through assigning more target-symbols into consecutive M-type groups.

In [4], the starting position of a group is 0, k, $2k, \cdots, (\lceil N/k \rceil - 1)k$, in which N is the bits number of a scan slice, k is the group length and equals $\lceil \log_2^{(N+1)} \rceil$. So in [4], the starting position of a group is fixed. When encoding a S-type group, it needs one slice-code. The data-code in the slice-code is used to indicate the target-symbol position. When encoding a single M-type group, two slice-codes are needed. The data-code in the first slice-code is used to indicate the starting position of the M-type group. However, because data-code has $\lceil \log_2^{(N+1)} \rceil$ bits, it is enough to indicate any one position in the scan slice. So the starting position of a group doesn't need to start from the fixed position 0, k, $2k$, $\cdots, (\lceil N/k \rceil - 1)k$, it can start from other position, such as the target-symbol position. The data from bit 0 to precede the first target-symbol bit can be grouped into a N-type group. Under the condition of the same group length, a

Procedure: flexible grouping strategy

```
1.  s=0;
2.  find target-symbol from index s;
    s is set to the value of target-symbol index;
3.  if group(s, s+k) is S-type      // k is the group length
    { Sign S-type for the group;
      s=s+k+1;
      goto 2;
    }
4.  else if group(s, s+k) is N-tpye
    {  s=s+k+1;
       goto 2;
    }
5.  else if group(s, s+k) is M-tpye
6.  {  Sign M-type for the group;
         s=s+k+1;
      if (group(s, s+k) is M-type)
         goto 6;
      else
         goto 2;
    }
```

Fig. 2 Pseudocode for flexible grouping strategy.

group whose first bit is target-symbol will have higher probability to be M-type and contain more target-symbols than the group whose first bit is not target-symbol. In this way, the number of M-type groups can be improved.

For example, the scan slice (X1XX1X00XX01X00), the target-symbol is value 1. According to the grouping method in [4], it has the following 4 groups: X1XX 1X00 XX01 000, in which there is no M-type group. The corresponding slice-codes are: 00 0001, 10 0100, 10 1011. However, if the group starts from target-symbol, the scan slice has the following groups: X 1XX1 X00XX0 1000. It has one M-type group. The corresponding slice-codes are: 00 1011, 11 0001, 11 1XX1.

The pseudo-code of our proposed grouping strategy is shown in Fig. 2. To make the procedure read easily, goto statement is used and the statements that terminate the procedure are omitted in the pseudocode.

In the pseudocode, group(m,n) denotes one group from bit m to bit n. The new grouping method will find the first target-symbol. Then based on the group type, different operation will be carried out. The keys to the new grouping strategy are as following: Firstly, to make a group contain more target-symbols, in general, the first bit of S-type and M-type is target-symbol. Secondly, once finding a M-type group, the algorithm will judge the following k bits is whether a M-type group. If it is M-type, the group and the directly preceding M-type groups can be merged into new consecutive M-type groups. In this way, more target-symbols can be contained into consecutive M-type groups and test data can be further reduced.

After finishing the procedure of Fig. 2, the groups without assigned M-type or S-type belongs to N-type groups. The following is an execution procedure example of the new grouping strategy. A scan slice is shown in Table 2. For the scan slice, $N = 31$, $k = 5$, the target-symbol is **1** and default value is **0**.

Table 2 A scan slice illustrating group procedure.

	0	1	2	3	4	5	6	7
scan slice bit	0	0	X	X	1	0	X	X
	8	9	10	11	12	13	14	15
scan slice bit	X	1	X	0	1	X	X	1
	16	17	18	19	20	21	22	23
scan slice bit	1	X	X	X	0	1	X	X
	24	25	26	27	28	29	30	
scan slice bit	0	X	X	X	0	0	X	

Table 3 The proposed group results.

No.	Group Data	Group Type
(1)	00XX	N-type
(2)	10XXX	S-type
(3)	1X01X	M-type
(4)	X11XX	M-type
(5)	X0	N-type
(6)	1XX0X	S-type
(7)	XX00X	N-type

Firstly, the method finds the first target-symbol which is the 4th bit in the slice of Table 2 and judges the type of group(4,8). Group(4,8) has only one target-symbol and will be signed with S-type. Then the algorithm will search for next target-symbol from bit 9. Because bit 9 is the target-symbol, the algorithm will judge the type of group(9,13). Group(9,13) with two target-symbols will be signed with M-type. Because group(9,13) is M-type, the algorithm continues judging the type of group(14,18) and will sign group(14,18) with M-type for it contains two target-symbols. Then the type of group(19,23) will be judged. Because it is S-type, the group(19,23) is canceled and will not be signed with S-type. The algorithm will search for target-symbol from bit 19. The above procedure is repeated until each bit is assigned into a group.

After finishing the grouping procedure, the scan slice is divided into seven groups, as shown in Table 3.

In the new group results, group 1, 5, 7 is N-type, group 2, 6 is S-type and group 3, 4 is M-type. From the example, we can see the first bit of S-type group must be target-symbol. Moreover, for consecutive M-type groups, the first bit of the first M-type group is also target-symbol. Group 3 and 4 are the example of two consecutive M-type groups. Besides, the group length of S-type and M-type is k. However, the group length of N-type group is variable, which is different from selective encoding. In selective encoding, the group length of N-type is also k. The group results of selective encoding is shown in Table 4.

Based on the new group results, it needs 5 slice-codes to encode the scan slice. However, based on the group results of selective encoding which is shown in Table 4, it needs 6 slice-codes to encode the scan slice. In the example, the flexible grouping strategy can reduce one slice-

Table 5 Group results comparison between proposed technique and [4].

Circuits	#Sc	Group results of proposed method				Group results[4]			
		S-type num.	M-type num.			S-type num.	M-type num.		
			M_1	M_2	M_1+M_2		M_1	M_2	M_1+M_2
s13207	63	4567	449	589(165)	1038	4807	417	517(169)	934
	127	3096	487	758(201)	1245	3279	542	627(175)	1169
	255	2571	426	818(201)	1244	2676	475	701(186)	1176
	511	2379	359	882(212)	1241	2469	399	778(199)	1177
s15850	31	4028	651	512(196)	1163	4471	649	346(148)	995
	63	3021	618	833(260)	1451	3275	647	683(233)	1330
	127	2320	639	1039(297)	1678	2528	687	907(290)	1594
	255	1917	648	1162(296)	1810	2038	691	1023(311)	1937
s38417	127	9980	2387	3139(950)	5526	15506	2473	2680(885)	5153
	255	7611	2035	3778(1107)	5813	8171	2151	3360(1055)	5511
	511	5647	1867	4108(1156)	5975	6015	2126	3664(1091)	5790
	1023	4410	1503	4483(1153)	5986	4696	1661	4219(1144)	5880
s38584	63	7322	1252	2444(695)	3696	8151	1373	2037(647)	3410
	127	5937	1223	2758(659)	3981	6342	1297	2458(717)	3755
	255	5155	1123	3147(653)	4270	5511	1238	2791(686)	4029
	511	4794	1108	3184(608)	4292	5091	1171	2952(644)	4123
	1023	4299	1109	3269(602)	4378	4481	1113	3154(667)	4267
b22s	63	12893	4763	2478(994)	7241	14467	4819	1917(821)	6736
	127	9026	4352	4025(1584)	8377	9978	4699	3413(1369)	8112
	255	6228	3242	5013(1849)	8225	6466	3535	4710(1797)	8245
	511	3878	1773	6010(2028)	7783	4297	1968	5746(2035)	7714
b17s	63	20654	2911	1720(576)	4631	22497	2595	1408(493)	4003
	127	15600	3868	2411(866)	6279	17010	3841	1881(678)	5722
	255	12846	3947	3284(1115)	7231	13991	3993	2881(1033)	6874
	511	10113	3506	4393(1493)	7899	10695	3984	3756(1345)	7740
	1023	6651	3396	4852(1576)	8248	7378	3748	4272(1450)	7350

Table 4 Group results of selective encoding [4].

No.	Group Data	Group Type
(1)	00XX1	S-type
(2)	0XXX1	S-type
(3)	X01XX	S-type
(4)	11XXX	M-type
(5)	01XX0	S-type
(6)	XXX00	N-type
(7)	X	N-type

code. The advantage of the proposed grouping strategy lies in that more target-symbols can be contained in consecutive M-type groups. In this way, the number of groups that needs to be encoded are reduced. So less test data are needed to encode the scan slice.

The group results comparison between our proposed method and selective encoding[4] is shown in Table 5. The character of the circuits and corresponding test set are shown in Table 8 of section 5. The #sc in the second column is the number of scan chains. The third column is the number of S-type groups. The following three columns show the number of M-type groups, in which, M_1 is the number of M-type groups that are not adjacent to any other M-type groups. Each group in M_1 needs two slice-codes to encode it. In the column of M_2, the data is given in the format of $a(b)$, in which, a is the number of M-type groups that are adjacent to other M-type and b means b data blocks. $a(b)$ means a M-type groups are distributed in b data blocks. Each block consists of two or multiple consecutive M-type groups. The value of a/b means how many M-type groups are contained in a data block averagely. The higher the value of a/b is, the better the compression results will be. M_1+M_2 is the total number of M-type groups.

From Table 5, it can be seen that with the increased number of scan chains, the number of S-type groups decreases and the number of M-type groups, especially the number of consecutive M-type groups, increases. Because group length(the data bits contained in a group) equals $\lceil \log_2^{N+1} \rceil$. With the increased number of scan chains, the group length will also increase. So the probability that a group is a M-type is increased.

Compared with selective encoding[4], our proposed method reduces the number of S-type and increase the number of M-type groups, especially the consecutive M-type groups increases. Furthermore, more consecutive M-type groups can be contained in a block averagely. So the group results achieves our desired goal and test data can be further reduced based on the new group results.

3.2 X Bits Exploitation for Power Reduction

In selective encoding mode, original test data in M-type group are directly copied into slice-codes. For example, the slice-code (11 X111) in slice 3 of Table 1, (X111) is the original test data. When the slice-codes are stored in ATE, the X bits among them can be filled with either value 1 or value 0. This motivates us to exploit more X bits and utilize these X bits to reduce test power.

To obtain more X bits, M-type and S-type group merging technique is developed to increase the number of X bits in M-type groups. Moreover, another kind of X bits is exploited by identifying some special scan slices. The special scan slices need the same number of slice-codes when target-symbol is set to value 0 and 1, separately. The two procedures will not result in compression ratio loss.

Group merging strategy: In M-type and S-type group merging technique, the S-type groups that can be merged must be adjacent to M-type groups. The group merging procedure changes S-type groups into M-type and doesn't increase test data. For example, in Table 3, it needs five slice-codes to encode the scan slice. However, if S-type group 2 is encoded as M-type, the scan slice is also encoded to five slice-codes. One slice-code for group 6, four slice-codes for three consecutive M-type groups 2,3,4. After group merging procedure, all the X bits in group 2 can also be utilized to reduce test power.

The merging procedure will decrease the number of S-type groups. When the number of S-type groups is smaller than the number of M-type groups (multiple consecutive M-type groups are counted as one M-type group), additional dummy slice-codes need be padded as explained in Sect. 2, which will result in the increase of test data. So the number of S-type groups that can be merged must be limited. If the number of S-type groups and M-type groups are num_s,num_m, the number of S-type allowed to be merged is $num_s - mum_m - 1$.

Identifying special scan slices: Besides the X bits in M-type groups, another kind of X bits can be exploited to reduce test power. Among test sets, we find there are many scan slices which need the same number of slice-codes when the target-symbol is set to 0 and 1 separately. In this paper, the kind of slices is named ***PRS-type*** scan slices (Power Reduction scan Slice). Other scan slices belong to ***NS-type***(Normal scan Slice). The X bits in PRS-scan slices can also be used to reduce test power. However, they are different from the X bits in M-type groups. The X bits in M-type groups can be filled with arbitrary value. The X bits in PRS-type scan slices must be filled with the same value. It is noted that the X bits in M-type groups of PRS-type scan slices can also be filled with arbitrary value.

The difference between NS-type and PRS-type scan slices lies in that the target-symbol of NS-type can be determined for the purpose of compression. However, the target-symbol of PRS-type scan slices is determined for the purpose of reducing test power.

Table 6 Proposed filling results and filling results [4].

Scan slices	Filling Results[4]	Proposed Results
1.XXXX1XX	1111111	0000100
2.XX0XXXX	0000000	0000100
3.X101X0X	1101101	0101000

Several filling example of PRS-type scan slices are shown in Table 6. The first column is 3 scan slices of PRS-type. The second column and third column is the filling results of selective encoding and our proposed method, separately. Two filling strategies need the same number of slice-codes. The filling results in [4] result in 12 switching activities. However, our filling strategy only has 3 switching activities. The proposed X-filling strategy will be presented in Sect. 3.3.

To identify PRS-type slices, an accurate method is that each slice is encoded twice when target-symbol is set to value 0 and 1 separately. If test set is very large and this procedure is time-consuming, the following three criteria can be used.

1. A scan slice without care bit;
2. A scan slice with only one care bit;
3. A scan slice with $k_0 = k_1 (k_0, k_1 \geq 1)$ and encoding the scan slice needs the same number of slice-codes when target-symbol is set to value 0 and 1 separately, where k_0 and k_1 is the number of value 0,1 in the scan slice.

A scan slice meeting one of three criteria can be categorized into PRS-type. The method may cause a small fraction of PRS-type scan slices misjudged into NS-type. For example, the scan slice(1011 XXXX XXX0 XXX), will be categorized into NS-type based on the above method. However, it needs the same number of slice-codes when target-symbol is 0 and 1 separately.

The percentage of exploited X bits using the proposed method is shown in Table 7. In the table, X_m, X_{gm} is the percentage of X bits in M-type groups before group merging and after group merging. X_{prs} is the percentage of X bits in PRS-type scan slices. The final column is the sum of X_{gm} and X_{prs}.

From the table, it can be seen that with the increased number of scan chains, the value of X_m,X_{gm} also increases,X_{prs} decreases except circuits b22s. Because a scan slice will have more data bits with the increased number of scan chains, the corresponding M-type group length is increased and may contain more X bits. So the value of X_m,X_{gm} increases. However, when the bits number of a scan slice increase, the probability that a scan slice is PRS-type turns smaller and this results in the decrease of X_{prs}. The X_{prs} in b22s increases, which is due to the low X bits density. When number of scan chains are increased, the number of PRS-type scan slices meeting the third criteria are increased due to the low X bits density. From Table 8 in Sect. 5, we can see that the test set of b22s has the lowest X bits density among all circuits.

Table 7 The percentage of X bits for power reduction.

Circuits	#Sc	The percentage of exploited X bits			
		X_m	X_{gm}	X_{prs}	Sum
s13207	63	0.83	1.15	66.4	67.55
	127	1.33	1.86	48.3	50.16
s15850	31	1.90	5.54	44.2	49.74
	63	3.63	8.45	23.9	32.35
s38417	127	2.53	3.77	76.8	80.57
	255	3.47	5.10	74.9	80.0
s38584	63	2.54	3.54	47.7	51.24
	127	3.99	5.59	33.2	38.79
b22s	63	4.99	7.06	15.0	22.06
	127	5.84	7.83	22.3	30.13
b17s	63	1.11	1.45	29.1	30.55
	127	2.19	2.85	16.6	19.45

3.3 X-Filling Strategy

The proposed X-filling strategy is a two-pass procedure. The first is partial X-filling procedure, the other is full X-filling procedure.

Partial X-filling procedure only fills the X-bits in N-type and S-type groups of NS-type scan slices with default value (complementary to target-symbol).

After partial X-filling procedure, the X bits in M-type groups of NS-type scan slices and all X bits in PRS-type scan slices are still unfilled. Full X-filling procedure will fill the X bits. Corresponding to different test vector type, full X-filling procedure has different filling strategies. Test vector can be categorized into the following three types:

1. Type-1: All the scan slices contained in the test vector are NS-type.
2. Type-2: Partial scan slices contained in it are NS-type, partial are PRS-type.
3. Type-3: All the scan slices contained in it are PRS-type.

Type-1 test vectors filling strategy: Because all the scan slices contained in this kind of test vector are NS-type, after partial X-filling procedure, the rest X bits all lies in M-type groups. So they can be filled arbitrarily with value 0 and 1. Adjacent filling strategy[15](also known as Minimum Transition filling) can be used.

Type-2 test vectors filling strategy: For the kind of test vector, the unfilled X bits includes the X bits in M-type groups of NS-type scan slice and all the X bits in PRS-type scan slice. The minimum transition filling results can be obtained if each PRS-type scan slice is filled twice with value 0 and 1 separately. However, the time complexity is 2^m, where m is the number of PRS-type scan slices in the test vector. To simplify the procedure, a greedy algorithm is developed to fill the kind of test vector.

The first step of the algorithm is to find a PRS-type scan slice adjacent to NS-type and determine the target-symbol

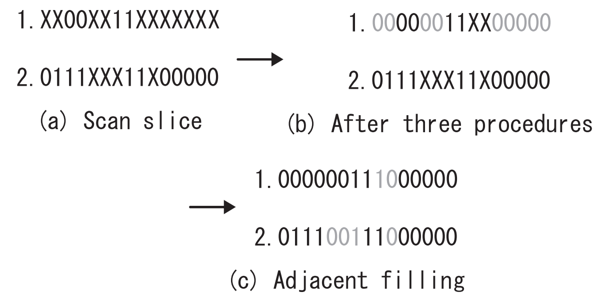


Fig. 3 Type-2 vector filling example.

of the PRS-type scan slice. If the PRS-type scan slice is above NS-type, the number of two-tuples (X,0) and (X,1) are counted separately, where X is the don't care bits in PRS-type, value 0 and 1 are the care bits in NS-type. If the number of (X,1) is larger than the number of (X,0), value 0 will be the target-symbol of the PRS-type scan slice. Otherwise, value 1 is the target-symbol. If PRS-type is below the NS-type, the number of two-tuples (0,X) and (1,X) will be counted to determine the target-symbol. After determining the target-symbol, scan slice grouping procedure, group merging procedure and partial X-filling procedure is used to deal with the scan slice. Then the PRS-type scan slice is changed to NS-type. The algorithm will repeat the above procedure until all the PRS-type scan slices are changed to NS-type. Finally the Type-2 test vector is changed to Type-1. Then Adjacent filling strategy is used to fill the rest X bits.

A filling example is shown in Fig. 3, in which scan slice 1 is PRS-type and the second is a NS-type scan slice after partial X-filling. The unfilled X bits in scan slice 2 are in M-type groups. In the example, the number of two-tuples (X,0), (X,1) is 7 and 1. So the target-symbol of the first slice is value 1. Figure 3 (b) is the result of flexible grouping, grouping merging and partial X-filling procedure. After the procedures, the type of slice 1 is changed to NS-type. Figure 3 (c) is the results of adjacent filling strategy.

It is noted that if a scan slice doesn't contain target-symbol after full X-filling procedure, one initial X bit can be filled with target-symbol. For example, the second slice in Table 6, the target-symbol is value 1 and will not be contained in the slice after full X-filling procedure. As shown in the column of proposed filling results, the fourth bit of the second slice can be filled with value 1 and this doesn't result in compression ratio loss.

Type-3 test vectors filling strategy: The filling strategy for Type-3 test vector is similar to Type-2.

The first step is to find a scan slice with care bit (for some test vectors, first several scan slices are all X bits). If the first scan slice in the test vector has care bit, the target-symbol of the second slice will be determined based on the number of (0, X) and (1, X), where 0,1 is the care bit of first slice, X is the don't care bits in the second slice. The method of determining the target-symbol is the same as Type-2 test vector. If the scan slice with care bit is not the first one,

Complete encoding procedure:

1.Test vectors are formatted into scan slices
2. **For** each test vector
3. **For** each scan slice
4. **If** slice type is PRS-type
5. Sign PRS-type for the scan slice
6. **else**
7. { Sign NS-type for the scan slice;
8. **Determine the target-symbol and default value;**
9. **Scan slice grouping procedure;**
10. **S-type and M-type group merging procedure;**
11. **Partial x-filling procedure;** }
12. **End for** (scan slice);
13. **Determine the test vector type;**
14. **Full x-filling procedure based on test vector type;**
15. **Generate the final slice-codes;**
16. **End for** (test vector);

Fig. 4 Pseudocode for complete encoding procedure.

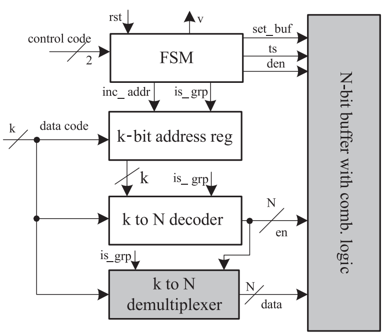


Fig. 5 Decompression architecture.

assuming it is the ith, the target-symbol of the $(i-1)$th scan slice can be determined based on the number of (X, 0) and (X,1), where X are the don't care bits of the $(i-1)$th scan slice, 0,1 are the care bits of the ith scan slice. Once the target-symbol is determined, scan slice grouping, group merging procedure and partial X-filling procedure will deal with the scan slice. Then the scan slice is changed into NS-type. After this procedure, the type of this vector is changed into Type-2 and the filling strategy for Type-2 can be used.

After full X-filling procedure, all the X bits in scan slice is determined and the needed slice-codes can be generated using selective encoding mode. The complete encoding procedure of the proposed technique is shown in Fig. 4. The scan slice grouping procedure in line 9 has been presented in Sect. 3.1, group merging procedure of line 10 is shown in Sect. 3.2.

4. Decompression Architecture

The decompression architecture is shown in Fig. 5, which has similar decoding procedure as [4].

The k-bits address register works in group-copy-mode and receives the starting address of M-type group. When consecutive M-type groups need to be decoded, the value of register will be added k per cycle, where k is the group length.

The k to N decoder is a single hot decoder. At any time only one of the N outputs will be asserted. In group-copy-mode, additional combinational logic is added in buffers to select k consecutive buffer cells.

The k to N demultiplexer is the new added model, which works in group-copy-mode and allocates k-bits data to k consecutive buffer cells. In [4], the data is directly connected to buffer cells and the model is not needed. The detailed decoding procedure refers to [4].

The purpose of the buffer is to form a N-bits scan slice. Once the scan slice is formed, the N-bits data in the buffer will be loaded into scan flip-flops in parallel. The procedure of forming the scan slice in buffer likes the procedure of writing data into memory cells. When encoding a S-type group, the data-code in slice-code is the address of the memory cell and sets the value of the memory cell to target-symbol. When encoding M-type groups, the data-code in the first slice-code is the starting address of memory cells, the following data are the contents to be written into memory cells from the starting address.

We synthesized the two decoders in the proposed technique and [4] by Synopsys Design Compiler using the SMIC18 library based on 0.18 micron technology. The area overhead of the two decoders for various number of scan chains are reported in Table 8. In the table, the first column is the number of scan chains. A_1 and A_2 denote the decoder area overhead in our proposed technique and [4], separately. The last column is the area overhead times of the two decoders. Compared with [4], the proposed technique needs larger area overhead. However, even the number of scan chains is 1023, the decoder area overhead in the proposed technique is less than 2% for million-gate circuits.

Table 8 Area overhead comparison.

#Sc	Area Overhead(cm^2)		A_1/A_2
	A_1: Prop.	A_2: [4]	
31	0.108×10^{-3}	0.106×10^{-3}	1.02
63	0.235×10^{-3}	0.199×10^{-3}	1.18
127	0.532×10^{-3}	0.385×10^{-3}	1.38
255	1.18×10^{-3}	0.752×10^{-3}	1.57
511	2.55×10^{-3}	1.49×10^{-3}	1.71
1023	5.22×10^{-3}	2.95×10^{-3}	1.77

Table 9 Character of experimental circuits.

Circuits	SFF	PI	T_D(bit)	FC	X ratio
s38584	1426	38	245952	100%	84.6%
s38417	1636	28	547456	100%	91.2%
s13207	638	62	205800	100%	93.2%
s15850	534	77	89817	100%	82.6%
b22s	735	32	329810	99.76%	75.3%
b17s	1415	37	801504	99.92%	89.9%

Table 10 Comparison of proposed technique and [4].

Circuits	#Sc	Test data reduction ratio			WSA during scan in and scan out					
					Average WSA			Peak WSA		
		[4]	Proposed	Inc.	[4]	Proposed	Red.	[4]	Proposed	Red.
s13207	63	3.88	4.08	5.15%	181.5	103.6	42.9%	366	319	12.8%
	127	4.08	4.48	9.80%	167.6	98.1	41.5%	396	383	3.28%
	255	4.16	4.6	10.58%	127.6	89.56	29.8%	453	421	3.06%
	511	3.96	4.45	12.37%	119.38	67.9	51.50%	489	473	3.27%
s15850	31	1.96	2.06	5.10%	150.7	97.6	35.2%	283	256	9.5%
	63	1.92	2.07	7.81%	147.2	96.5	34.4%	309	280	9.4%
	127	1.78	2.0	12.36%	126.37	90.95	28.03%	336	243	27.68%
	255	1.68	1.9	13.10%	112.58	67.39	40.14%	360	329	8.6%
s38417	127	2.92	3.19	9.25%	242.3	179.1	26.1%	1037	834	19.6%
	255	2.95	3.28	11.2%	183.5	156.7	14.6%	1046	1002	4.2%
	511	2.94	3.38	14.97%	223.6	162.73	27.2%	1182	1168	1.2%
	1023	3.17	3.49	10.09%	213.87	132.83	37.89%	1171	1135	3.07%
s38584	63	2.09	2.32	11.0%	430.9	348.1	19.2%	767	725	5.5%
	127	2.00	2.29	14.5%	423.0	341.2	19.3%	833	741	11.0%
	255	1.88	2.18	15.96%	310.08	291.53	5.98%	877	837	4.56%
	511	1.71	2.06	20.47%	286.0	213.1	25.49%	968	934	3.5%
	1023	1.80	1.96	8.89%	159.92	124.1	22.4%	1004	981	2.29%
b22s	63	1.45	1.54	6.21%	166	122.7	26.1%	505	443	12.3%
	127	1.39	1.51	8.63%	142.4	98.8	30.6%	564	457	18.97%
	255	1.42	1.62	14.08%	144.4	94.23	34.7%	587	456	28.86 %
	511	1.51	1.86	23.18%	84.43	77.59	8.1%	598	547	8.53%
b17s	63	3.24	3.39	4.63%	361.9	269.9	25.4%	940	796	15.3%
	127	3.11	3.27	5.14%	325.6	289.4	11.1%	1070	937	12.4%
	255	2.92	3.14	7.53%	245.73	236.59	3.72%	1174	1057	9.97%
	511	2.85	3.14	10.18%	203.86	185.34	9.08%	1204	1109	7.89%
	1023	2.91	3.29	13.06%	238.46	181.9	23.72%	1236	1144	7.44%
Ave.		2.53	2.79	10.97%	212.26	162.21	25.6%	759.81	692.58	9.68%

5. Experimental Results

The proposed technique and selective encoding[4] are applied to several large circuits in ISCAS'89 and ITC'99 benchmark circuits. ATPG tool "TetraMAX" from Synopsys is used to generate test set with no random-filling. The character of the benchmark circuits and corresponding test set are reported in Table 9. In the table , SFF in the second column denotes the number of scan flip flops. PI is the number of primary inputs. T_D is the size of generated test set and FC is the fault coverage of circuits. The last column is the percentage of X bits in T_D.

It is known that the commercial tools based on broadcasting scan and linear decompressor are available. However, the two methods need to interactive with ATPG and fault simulation. This means the circuit structure must be known. However, for some designs based on IP cores, the circuit structure of some IP cores are not known because IP cores vendors keep it as commercial secrets[4]. In this situation, system integrators only compress the test vectors supplied by the IP cores vendors. During compression, the ATPG and fault simulation tool can not be used. The selective encoding technique[4] is proposed to compress the IP cores test data. Our proposed technique is an extension to selective encoding, also applies to IP cores compression and doesn't need to interactive with ATPG and fault simulation. So it is unfair to compare our proposed technique with the commercial compression tools based on linear decompressor and broadcasting scan.

Besides, if we directly use the techniques based on linear decompressor and broadcasting scan to compress the existed test set, test data compression ratio will be very low. This is due to the character of test set: large majority of bits in the first several test cubes is care bits. To compress the test cubes, the number of external inputs to the decomprssor will be very large. In the two techniques, test data compression ratio can be calculated as: $1 - N_E/N_I$, in which N_I is the number of internal scan chain inputs, N_E is the number of external inputs(the inputs to decompressor). For a circuit, N_I is constant. When N_E turns larger, the compression ratio turns smaller. In the paper, we compared our proposed technique with the most closely related work [4] and [20].

The comparison results of the proposed technique and [4] are shown in Table 10. The #sc in the second column is the number of scan chains. The test data reduction ratio[4] in the table is defined as T_D/T_E, in which T_E is the test data bits after compression. The column of "Inc." is the percentage of increased test data reduction ratio compared to [4]. The proposed technique improve test data reduction

Table 11 Comparison between [20] and proposed technique.

Circuits	#Sc	Test data storage bits			WSA during scan in and scan out					
					Average WSA			Peak WSA		
		Proposed	[20]	Inc.	Proposed	[20]	Red.	Proposed	[20]	Red.
s13207	63	50440	61272	21.48%	103.6	72.555	29.97%	319	318	0.3%
	127	45981	57393	24.82%	98.12	60.87	37.96%	383	322	15.93%
	255	44740	56450	26.17%	89.56	47.4	47.07%	421	303	28.03%
	511	46299	58619	26.61%	67.9	46.13	32.06%	473	264	44.19%
s15850	31	43582	51674	18.57%	97.59	54.4	44.26%	256	250	2.34%
	63	43424	51528	18.66%	96.46	50.4	47.75%	280	255	8.93%
	127	44892	53109	18.30%	90.95	40.35	55.63%	243	217	10.70%
	255	47260	55130	16.65%	67.39	54.24	19.51%	329	228	30.70%
s38417	127	171783	210879	22.76%	179.1	138.4	22.72%	834	793	4.92%
	255	166730	205210	23.08%	156.71	120.88	22.86%	1002	758	24.35%
	511	162074	197043	21.58%	162.73	118.41	27.24%	1168	716	38.69%
	1023	156972	190944	21.64%	132.83	122.79	7.56%	1135	577	49.16%
s38584	63	105840	136144	28.63%	348.123	134.1	61.48%	725	697	3.86%
	127	107379	137106	27.68%	341.2	123.84	63.70%	741	688	7.15%
	255	112920	143410	27.00%	291.53	109.51	62.44%	837	666	20.43%
	511	119251	150161	25.92%	213.1	102.34	51.98%	934	598	35.97%
	1023	125184	157164	25.55%	124.1	81.18	34.59%	981	618	37.00%
b22s	63	214320	238328	11.20%	122.7	81.21	33.81%	443	333	24.83%
	127	217944	239787	10.02%	98.8	72.09	27.03%	457	312	31.73%
	255	203510	222450	9.31%	94.23	60.15	36.17%	456	343	24.78%
	511	177309	200354	13.00%	77.59	62.53	19.41%	547	465	14.99%
b17s	63	236352	324048	37.10%	269.9	153.98	42.95%	796	690	13.32%
	127	245250	321219	30.98%	289.4	136.72	52.76%	937	698	25.51%
	255	255220	322190	26.24%	236.59	142.86	39.62%	1057	771	27.06%
	511	255607	307109	20.15%	185.34	109.16	41.10%	1109	606	45.36%
	1023	243468	284976	17.05%	181.9	117.9	35.18%	1144	886	22.55%
Ave.		140144	170524	21.93%	162.21	92.86	38.34%	693	514	22.80%

ratio 10.97% averagely. Besides, with the increased number of scan chains, inc is also increased. Because the group length increases with the increased number of scan chains, the proposed flexible grouping strategy will have more effective grouping results compared to selective encoding[4].

The second part of the table is the comparison results of test power. The weighted switching activity (WSA) is used to evaluate the power value. The average WSA is calculated by total transitions counts(including scan in and scan out) / shift cycles during test. The peak WSA is the maximum transition counts in a cycle during test. Compared to [4], our proposed technique reduces average WSA and peak WSA by 25.6%, 9.68% during scan in and scan out.

Table 11 is the comparison results between [20] and our proposed technique. The primary objective in [20] is reducing test power. It firstly fills all the X bits by 0-filling, 1-filling and adjacent-filling, separately, then selective encoding is used to compress test data. However, the primary objective in our proposed technique is reducing test data. Then with no loss of compression ratio, large number of X bits is exploited to reduce test power. So compared with [20], it can be inferred that our proposed technique needs fewer test data bits, but the test power consumption is larger than [20]. In Table 11, the results in the column [20] are obtained by using 0-filling strategy fills X bits, then selective encoding is used to compress test data. Compared with our proposed technique, [20] needs 21.93% larger test data storage and reduces average and peak WSA 38.34%, 22.80%, respectively.

6. Conclusion

In this paper, an extended selective encoding of scan slices is proposed. The proposed technique reduces test data storage by a flexible grouping strategy. By assigning more targets-symbols into consecutive M-type groups, the flexible grouping strategy can decrease the number of groups to be encoded and increases the number of consecutive M-type groups. Moreover, with no test compression ratio loss, a large number of X bits are exploited by S-type and M-type group merging and identifying PRS-type scan slices. The corresponding X-filling strategy is proposed to fill the X bits. Compared to selective encoding, the average and peak WSA is reduced 25.6% and 9.68%, respectively, during scan shift.

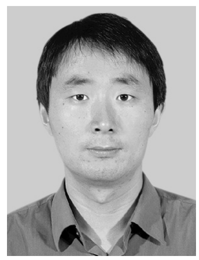
Acknowledgment

The work was supported in part by National Natural Science

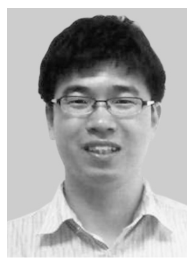
Foundation of China (NSFC) under grant No.(60633060, 60806014, 60831160526, 60803031), in part by National Basic Research Program of China (973) under grant No.2005CB321604, and in part by Hi-Tech Research and Development Program of China (863) under grant No.(2007AA01Z109, 2007AA01Z113, 2009AA01Z126).

References

[1] A. Jas, J. Gosh-Dastidar, M.-E. Ng, and N.A. Touba, "An efficient test vector compression scheme using selective Huffman codin," IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst., vol.22, no.6, pp.797–806, 2003.
[2] A. Chandra and K. Chakrabarty, "System-on-a-chip test data compression and decompression based on internal scan chains and Golomb coding," IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst., vol.21, no.6, pp.715–722, June 2002.
[3] A. Chandra and K. Chakrabarty, "Test data compression and test resource partitioning for system-on-a-chip using frequency-directed runlength (FDR) codes," IEEE Trans. Comput., vol.52, no.8, pp.1076–1088, 2003.
[4] Z. Wang and K. Chakrabarty. "Test data compression using selective encoding of scan slices," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol.16, no.11, pp.1429–1440, Nov. 2008.
[5] I. Hamzaoglu and J.H. Patel, "Reducing test application time for full scan embedded cores," Proc. IEEE Annual International Symposium on Fault-Tolerant Computing, pp.260–267, 1999.
[6] W. Rao, A. Orailoglu, and G. Su, "Frugal linear network-based test decompression for drastic test cost reductions," Proc. International conference on computer-aided design, pp.721–725, 2004.
[7] Y. Han, Y. Hu, X. Li, H. Li, and A. Chandra, "Embedded test decompressor to reduce the required channels and vector memory of tester for complex processor circuit," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol.15, no.5, pp.531–540, May 2007.
[8] L.-T. Wang, X. Wen, S. Wu, Z. Wang, Z. Jiang, B. Sheu, and X. Gu, "VirtualScan: Test compression technology using combinational logic and one-pass ATPG," IEEE Des. Test Comput., vol.25, no.2, pp.122–130, 2008.
[9] C.-W. Tzeng and S.-Y. Huang, "UMC-Scan test methodology: Exploiting the maximum freedom of multicasting," IEEE Des. Test Comput., vol.25, no.2, pp.132–140, 2008.
[10] J. Rajski, J. Tyszer, M. Kassab, and N. Mukherjee, "Embedded deterministic test," IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst., vol.23, no.5, pp.776–792, May 2004.
[11] H.G. Liang, S. Hellebrand, and H.-J. Wunderlich, "Two-dimensional test data compression for scan-based deterministic BIST," Proc. International Test Conference, pp.894–902, Nov. 2001.
[12] N.A. Touba, "Survey of test vector compression techniques," IEEE Des. Test Mag., vol.23, no.4, pp.294–303, 2006.
[13] P.M. Rosinger, B.M. Al-Hashimi, and N. Nicolici, "Scan architecture with mutually exclusive scan segment activation for shift- and capture-power reduction," IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst. vol.23, no.7, pp.1142–1153, July 2004.
[14] S. Wang and S.K. Gupta, "ATPG for heat dissipation minimization during test application," IEEE Trans. Comput., vol.47, no.2, pp.256–262, Feb. 1998.
[15] A. Chandra and R. Kapur, "Bounded adjacent fill for low capture power scan testing," Proc. VLSI Test Symp., pp.131–138, 2008.
[16] Y.-Z. Wu and M.C.-T.Chao, "Scan chain reordering for minimizing scan-shift power based on non-specified test cubes," Proc.VLSI Test Symp., pp.147–154, 2008.
[17] A. Chandra and K. Chakrabarty, "Combining low-power scan testing and test data compression for system-on-chip," Proc. Design Automation Conference (DAC), pp.166–169, 2001.
[18] C.-Y. Lin and H.-M. Chen, "A selective pattern-compression scheme for power and test-data reduction," Proc. International Conference on Computer-Aided Design, pp.520–525, 2007.
[19] M. Wu, J. Huang, X. Wen, and K. Miyase, "Reducing power supply noise in linear-decompressor-based test data compression environment for at-speed scan testing," Proc. International Test Conference, pp.1–10, Oct. 2008.
[20] N. Badereddine, Z. Wang, P. Girard, K. Chakrabarty, A. Virazel, S. Pravossoudovitch, and C. Landrault, "A selective scan slice encoding technique for test data volume and test power reduction," J. Electronic Testing: Theory and Applications, vol.24, no.4, pp.353–364, 2008.
[21] J. Li, X. Liu, Y. Zhang, Y. Hu, X. Li, and Q. Xu, "On capture power-aware test data compression for scan-based testing," Proc. IEEE International Conference on Computer-Aided Design, pp.67–72, Nov. 2008.



Jun Liu received the B.S. degree from North East Normal University, Changchun, China, in 2002 and M.S. degree from Hefei Unversity of Technology, Hefei, China, in 2007, repectively, both in Computer Science. He joined Hefei University of Technology as an assistant in 2002 and was promoted to Assistant Professor in 2008. He is currently working toward the Ph.D. degree on computer science at the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China. His research interests include test data compression, built-in self-test(BIST).



Yinhe Han received the B. Eng. degree from Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2001, and the M. Eng. and Ph.D. degree in computer science from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, in 2003 and 2006, respectively. He is currently an Associate Professor at the Institute of Computing Technology, Chinese Academy of Sciences. His research interests include VLSI Design and Test, Reliable System and Architecture. Dr. Han was a recipient of the Test Technology Technical Council Best Paper Award at the Asian Test Symposium 2003. He is also a member of ACM/IEEE.

Xiaowei Li received his B.Eng. and M.Eng. degrees in computer science from Hefei University of Technology (China) in 1985 and 1988 respectively, and his Ph.D. degree in computer science from the Institute of Computing Technology, Chinese Academy of Sciences in 1991. Dr. Li joined Peking University as a Postdoctoral Research Associate in 1991, and was promoted to Associate Professor in 1993, all with the Department of Computer Science and Technology. From 1997 to 1998, he was a Visiting Research Fellow in the Department of Electrical and Electronic Engineering at the University of Hong Kong. In 1999 and 2000, he was a Visiting Professor in the Graduate School of Information Science, Nara Institute of Science and Technology, Japan. He Joined the Institute of Computing Technology, Chinese Academy of Sciences as a professor in 2000. His research interests include VLSI/Soc design verification and test generation, design for testability, low-power design, dependable computing. Dr. Li received the Natural Science Award from the Chinese Academy of Sciences in 1992, the Certificate of Appreciation from IEEE Computer Society in 2001. He is a senior member of IEEE. He is an area editor of the Journal of Computer Science and Technology and an associate editor-in-chief of the Journal of Computer-Aided Design & Computer Graphics (in Chinese).