

## LETTER

# Grouped Scan Slice Repetition Method for Reducing Test Data Volume and Test Application Time

Yongjoon KIM<sup>†</sup>, Myung-Hoon YANG<sup>†</sup>, Jaeseok PARK<sup>†</sup>, Eunsei PARK<sup>†</sup>, *Nonmembers*,  
and Sungho KANG<sup>†a)</sup>, *Member*

**SUMMARY** This paper presents a grouped scan slice encoding technique using scan slice repetition to simultaneously reduce test data volume and test application time. Using this method, many scan slices that would be incompatible with the conventional selective scan slice method can be encoded as compatible scan slices. Experiments were performed with IS-CAS'89 and ITC'99 benchmark circuits, and results show the effectiveness of the proposed method.

**key words:** *design for testability (DfT), scan testing, test data compression*

## 1. Introduction

In modern semiconductor manufacturing testing, the increase in test data volume is a major contributor to high test costs. A large test data volume requires not only a long test application time but also automatic test equipment (ATE) with greater memory capability. Moreover, test data volume continues to increase as the design complexity grows. Integration of embedded cores in system-on-chip (SoC) applications also increases the test data volume. Multiple scan chains may reduce the test application time, but the limited number of ATE channels cannot support a high enough pin count for internal scan chains.

A test data compression technique may prove to be a promising solution to this problem. Test data compression methodologies use a specific test data encoding scheme to compress huge test data volumes. The original test data is decompressed by an on-chip decoding logic. The reduced test data volume can be loaded to the ATE memory at a time, and the many internal scan chains are also applied with small ATE channels. Many compression techniques have been developed to reduce test data volume and test application time [1]–[3]. These methods utilize the fact that test patterns tend to contain many unspecified bits. Recent research shows that only a few specified bits are included in the test cubes [4]. In [1], a constrained automatic test pattern generation (ATPG) method was proposed to enhance the Illinois scan architecture. However, this ATPG requires very complex procedures and increases the number of test patterns. A decompression hardware generation scheme for a linear decompression network was developed to share a single input for many scan chains [2]. However, the compo-

sition of the linear decompression network is based on the test set. Therefore, redesign of the on-chip decoder is also required if any design modification is required that changes the test set. The selective scan slice encoding method has been proposed to reduce the test data volume and the test application time [3]. However, this method shows good results only when the density of the unspecified bits is very high. In addition, the various decoding modes require a complex on-chip decoder.

In this paper, we propose a test data compression technique based on grouped scan slice repetition. Using this method, both the test data volume and the test application time are dramatically reduced. Moreover, the on-chip decoder, based on a one-hot decoder, does not require any information regarding the design or the test set. Therefore, the proposed test data compression method could be a promising solution for efficient scan-based testing.

## 2. Proposed Test Data Compression Technique

In proposing our test data compression technique, it is important to define some terms concerning the relationship of scan slices. First, two scan slices are *compatible* if they can be identical with each other. The unspecified bits can be properly allocated for making compatible scan slices. For example, scan slices XX0 and 01X are compatible if the second X bit of XX0 is assigned as 1 and the remaining Xs are assigned as 0s. Two scan slices are *inverse-compatible* if each bit of a scan slice is reversed as compared to the other one. For example, scan slices XX0 and 01X are inverse-compatible if the second X bit of XX0 is assigned as 0 and the remaining Xs are assigned as 1s. Two scan slices are *incompatible* if they are neither compatible nor inverse-compatible. A *conflict bit* depicts the position of a bit which makes scan slices incompatible. For example, scan slices 01100 and 11000 are incompatible due to the first and third bits of each scan slice; these two bits are conflict bits. Of course, the second, fourth, and fifth bits can be conflict bits from an inverse-compatibility point of view, but the fewer number of conflict bits is preferred for a better compression ratio.

The overall architecture of the proposed method is shown in Fig. 1. This structure is based on the repetition of successive scan slices. When successive scan slices are compatible or inverse-compatible, the latter scan slice can be decoded without detailed data bits. However, this scheme

Manuscript received October 3, 2008.

Manuscript revised January 31, 2009.

<sup>†</sup>The authors are with the Dept. of Electrical & Electronic Eng., Yonsei University, Seoul, Korea.

a) E-mail: shkang@yonsei.ac.kr

DOI: 10.1587/transinf.E92.D.1462

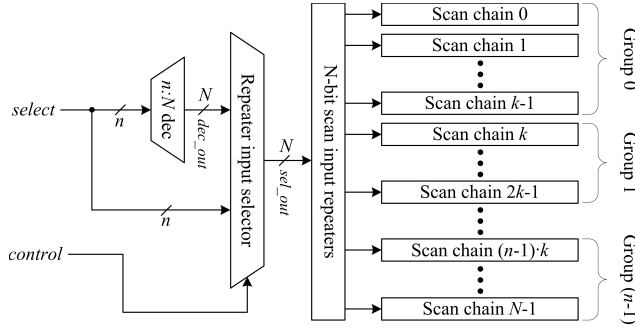


Fig. 1 Grouped scan slice repetition architecture.

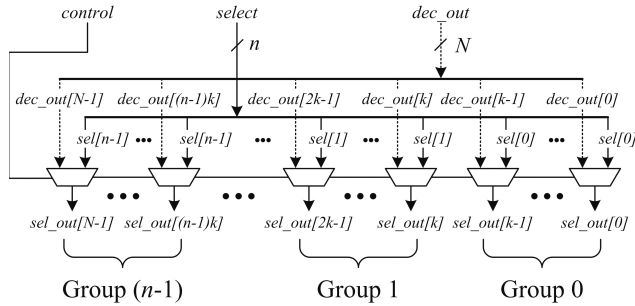


Fig. 2 Repeater input selector.

is not good enough to make compatible scan slices composed of large scan chains, since the conflict bits may exist at shorter intervals. Therefore, we also propose a scan chain grouping method. Using this method, each scan group shares separate common inputs, leading to the enhancement of encoding efficiency. To implement the conventional selective scan method [3], at least  $n = \lceil \log_2 N \rceil$  input signals are required to compose  $n : N$  decoding logic. Accordingly, we can use these input signals as the group inputs without any pin overhead as compared to the conventional method. Therefore, the number of scan groups is determined as  $n$ , and each scan group holds  $\lceil \frac{N}{n} \rceil$  scan chains.

Figure 2 shows the repeater input selector. It controls the  $N$ -bit scan input repeaters with  $N$ -bit  $sel\_out$  signals. When the successive scan slices are compatible or inverse-compatible, the  $control$  signal goes logic 0 and the  $select[k]$  signal is directly delivered to the scan input repeater, which is connected to the  $k$ -th group using the criteria ( $0 \leq k \leq n$ ). If  $select[k]$  is logic 0, the  $k$ -th group is in a compatible mode. If  $select[k]$  is logic 1, the  $k$ -th group is in an inverse-compatible mode. For an incompatible mode, the  $control$  signal goes logic 1, and the  $n : N$  decoder selects a single scan chain to reverse the present logic value. The  $n : N$  decoder is composed of a one-hot decoder which enables only a single output within  $N$  outputs. Therefore, a conflict bit can be generated by selecting a single scan input repeater during an incompatible mode. The scan input repeater is composed of  $N$ -bit toggle flip-flops as shown in Fig. 3 ( $0 \leq i \leq N$ ). Figure 3 shows the  $i$ -th scan input repeater cell which is connected to  $sel\_out[i]$  and  $scan\_in[i]$ , and  $scan\_in[i]$  is directly connected to the input of the  $i$ -th

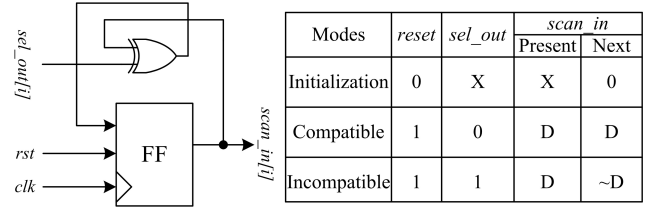


Fig. 3 Scan input repeater cell.

Table 1 Scan slice encoding example.

Scan slice	Present state	Encoding data		Next state
		Control	Select	
1X1X1X0X	00000000	0	110	11111100
X1100X10	11111100	0	010	11100000
	11100000	1	110	11100010

scan chain.

Table 1 shows an example of scan slice encoding for 8 scan chains. Each column depicts the input scan slice, the present scan input repeater state, the encoding data, and the next repeater state. The initial state of the repeater is assumed to be all 0s. At first, the scan chains are grouped with 3 groups, since  $\lceil \log_2 8 \rceil = 3$ . After that, each group can be encoded using the compatibility characteristic. For example, each group of the scan slice 1X1X1X0X is 1X1, X1X, and 0X; the first and second groups are inverse-compatible and the third group is compatible with the present repeater state. Therefore, the scan slice is encoded at a time as  $Control=0$  and  $select=110$ . Next, X1100X10 is also compatible for the first and second groups. They are compatible and inverse-compatible, respectively. However, the third group is incompatible. Therefore, this scan slice cannot be encoded with a single encoding unit.  $Control=0$  and  $select=010$  make the next state 11100000. In this state, the conflict bit can be decoded by  $control=1$ , and now  $select = 110$  points out the position of the conflict bit as 110. Consequently, additional encoding data is required to encode the position of the conflict bit. Using these encoding processes, many scan slices that would be incompatible with the conventional selective scan slice encoding method can be encoded as compatible scan slices. In addition, the number of conflict bits in the incompatible scan slices is dramatically reduced due to the grouping method. The detailed grouped scan slice encoding algorithm is shown in Fig. 4. First, a scan chain is segmented into  $N$  scan chains, and then these scan chains are grouped into  $n = \lceil \log_2 N \rceil$  scan groups. Hence, the bit width of the scan slice is  $N$ , and the control and select codes are encoded as follows. A scan slice is inserted, and each scan group  $G_x$  is compared with the present values ( $0 \leq x \leq n$ ). If  $G_x$  is compatible or inverse-compatible, the  $x$ -th bit of select code is 0 or 1, respectively. In this case, the control code holds logic 0. If there is a conflict bit, the control code is encoded as logic 1, and the select code is encoded as the position of the conflict value. The encoding procedure is completed when every scan slice is encoded under these criteria.

**Table 3** Comparisons of test data volume and test application time.

Circuits	[3]					Proposed method				
	SC/CH	$T_E$ (b)	$Red_{vol}$ (%)	TAT (cycles)	$Red_{TAT}$ (%)	SC/CH	$T_E$ (b)	$Red_{vol}$ (%)	TAT (cycles)	$Red_{TAT}$ (%)
s13207	127/9	41,751	61.28	4,808	60.49	64/7	25,921	75.96	3,703	75.36
s38584	63/8	270,400	69.42	34,420	69.16	128/8	142,808	83.85	17,851	83.45
b17	1023/12	257,652	71.19	22,150	70.61	128/8	175,312	80.40	21,914	79.96
b18	2047/13	1,814,813	87.27	144,254	86.92	256/9	1,224,702	91.41	136,078	91.16
b19	2047/13	5,871,628	91.66	508,068	90.65	512/10	3,868,940	94.50	386,894	94.35

```

//A scan chain is segmented as N scan chains
//N scan chains are grouped as n = ⌈log2 N⌉ scan groups
//Each scan group holds k = ⌈N/n⌉ scan chains
//Initialize the present values of repeater as all 0s

Grouped scan slice encoding procedure:
(1)For the given test pattern;
(2) for each scan slice
(3)  for each scan group Gx
(4)   Compare input data with the present values
(5)   if Gx is compatible
(6)    the x-th bit of select code=0;
(7)   else if Gx is inverse compatible
(8)    the x-th bit of select code=1;
(9)   else if compatible pairs are dominant
(10)    the x-th bit of select code=0;
(11)  else
(12)   the x-th bit of select code=1;
(13) end for (groups);
(14) control code=0;
(15) if there is the incompatible scan group
(16)  for each conflict value
(17)   control code=1;
(18)   select code=the position of the conflict value;
(19)  end for (conflict values);
(20) end if
(21) Update the present values
(22)end for (slices);

```

**Fig. 4** Encoding algorithm for grouped scan slice repetition.**Table 2** Information regarding ISCAS'89 and ITC'99 benchmark circuits.

Circuits	No. of cells	No. of pat	$T_D(b)$	$D_S$
s13207	638	169	107,822	9.56
s38584	1,426	620	884,120	5.78
b17	1,317	679	894,243	7.40
b18	3,064	4,653	14,256,792	2.54
b19	6,130	11,484	70,396,920	1.41

### 3. Experimental Results

Experiments were performed on the largest ISCAS'89 and ITC'99 benchmark circuits. Each circuit was synthesized using Design Compiler [5] and the test patterns were generated by TetraMax [6]. The main features of the benchmark circuits are presented in Table 2. Each column shows the circuit names, the number of scan cells, the number of test patterns, the size of the test data volume ( $T_D$ ), and the density of specified bits ( $D_S$ ). The density of the specified bits in the original test data averaged 4.78%.

Table 3 presents the comparison of the test data volume and the test application time. SC, CH,  $T_E$ , and TAT represent the number of scan chains, the number of ATE channels, the size of the encoded data, and the test application time, respectively. The CH of [3] and the proposed method is calculated as  $\lceil \log_2(N+1) \rceil + 2$  and  $\lceil \log_2 N \rceil + 1$ , where  $N$  is the number of scan chains. As shown in the table, the proposed method presents a better compression ratio than [3] due to the increase in the number of compatible scan slices and the decrease in the number of conflict bits. The test application time is also less than it is for [3], since it is proportional to the size of the compressed data. Moreover, the number of ATE channels for the proposed method is less than that of [3]. Therefore, more chips can be simultaneously tested when this new compression technique is applied, and grouped scan slice repetition can be used as an effective test data compression technique.

### 4. Conclusions

In this paper, we have proposed a test data compression technique for reducing test data volume and test application time using scan slice repetition and grouping. This new method dramatically increases the number of compatible scan slices and decreases the number of conflict values. In addition, the on-chip decoder used for the proposed method is determined by the number of scan chains, and no circuit information, such as design and test set data affects the decoder. Therefore, the proposed method shows promise as a solution for efficient scan-based testing.

### Acknowledgement

This work was supported by "System IC 2010" project of Korea Ministry of Knowledge Economy.

### References

- [1] F.F. Hsu, K.M. Butler, and J.H. Patel, "A case study on the implementation of the illinois scan architecture," Proc. IEEE International Test Conference (ITC), pp.538–547, 2001.
- [2] I. Bayraktaroglu and A. Orailoglu, "Decompression hardware determination for test volume and time reduction through unified test pattern compaction and compression," Proc. IEEE VLSI Test Symposium (VTS), pp.113–118, 2003.
- [3] 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, 2008.

- [4] T. Hiraide, K.O. Boateng, H. Konishi, K. Itaya, M. Emori, and H. Yamanake, "BIST-aided scan test - A new method for test cost reduction," Proc. IEEE VLSI Test Symposium (VTS), pp.359–364, 2003.
  - [5] Design Compiler User Guide Version 2006. 06, Synopsys, Inc., Mountain View, CA, 2005.
  - [6] TetraMax Reference Manual. Release 2006. 06, Synopsys Inc., Mountain View, CA, 2004.
-