LETTER

Selective Scan Slice Grouping Technique for Efficient Test Data Compression

Yongjoon KIM[†*a)], **Jaeseok PARK**[†], ***Nonmembers***, ***and*** **Sungho KANG**[†], ***Member***

SUMMARY This paper presents a selective scan slice grouping technique for test data compression. In conventional selective encoding methods, the existence of a conflict bit contributes to large encoding data. However, many conflict bits are efficiently removed using the scan slice grouping technique, which leads to a dramatic improvement of encoding efficiency. Experiments performed with large ITC'99 benchmark circuits presents the effectiveness of the proposed technique and the test data volume is reduced up to 92% compared to random-filled test patterns.

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

1. Introduction

In modern semiconductor manufacturing testing, test data compression may give a promising solution to reduce test costs [1]. Test data compression utilizes the specific test data encoding scheme to compress huge test data volume and the on-chip decoder to decompress the original test data. Many compression techniques have been developed to reduce test data volume [2]–[5].

Broadcast-scan-based schemes [2], [3] were proposed to support dynamic scan connection. Using these methods, high encoding efficiency can be achieved and broadcast-able test patterns can be generated using constrained Automatic Test Pattern Generation (ATPG). However, the constrained ATPG requires more test patterns than normal ATPG for obtaining the desired fault coverage and a higher compression ratio.

Linear decompressor-based scheme were proposed in [4]. In [4], multiple scan inputs share some primary inputs through combinational or sequential linear expansion networks. However, the constrained ATPG, which limits dynamic compaction from original test cube, is also required for making compression ratio high.

Selective scan slice encoding techniques [5], [6] were developed to achieve high test data compression ratio. Each scan slice is encoded based on the number of the 0 s and 1 s. These methods do not require detailed structural information about the CUT, and utilize a generic on-chip decoder that is independent of the CUT and the test set. However, these methods show good results only when the density of unspecified bits is very high, and the various decoding modes necessitate a complex on-chip decoder.

Manuscript received August 24, 2009.
Manuscript revised October 27, 2009.
[†]The authors are with the Dept. of Electrical & Electronic Eng., Yonsei University, Seoul, Korea.
a) E-mail: yjkim@soc.yonsei.ac.kr
DOI: 10.1587/transinf.E93.D.380

In this paper, we propose a selective scan slice grouping technique which is based on the selective scan slice encoding technique for which the encoding efficiency can be dramatically enhanced compared to the previous methods.

2. Proposed Test Data Compression Technique

The key terminology used in this paper is as follows:

- A *scan slice* is the input stimulus which is applied to the scan chains at a given cycle.
- Let $n_s(v)$ be the number of the v-specified bits within a scan slice. A scan slice is *constant* if $n_s(0) = 0$ or $n_s(1) = 0$.
- A scan slice is *variable* if $n_s(0) \neq 0$ and $n_s(1) \neq 0$.
- The *conflict bit* is the position of the specified bit that makes the scan slice variable. If $n_s(0) > n_s(1)$ ($n_s(1) > n_s(0)$), each position of logic 1 (0) is a conflict bit.

The proposed method is based on the selective scan slice encoding scheme proposed in [5]. In [5], the constant scan slice and the variable scan slice with a single conflict bit can be encoded with a single encoding data is composed of $\lceil \log_2(N+1) \rceil + 2$ bits, where N is the number of scan chains. Otherwise, each conflict bit should be encoded with a separate encoding data. Table 1 presents selective encoding example presented in [5]. Since $N = 16$, the size of data code is 5. As shown in Table 1, scan slice X11X XX1X X0X0 XX1X which has two conflict bits is encoded with two encoding data. At first, the unspecified bits are assigned as 1 while bit 9 is set to 0. The remaining conflict bit on bit 11 is encoded with additional encoding data. Scan slice 0XX1 1XXX X0XX XX01 is also encoded with three encoding data using the same method. [5] also presents an additional volume reduction method, but it may worsen the encoding efficiency if there are many conflict bits. Therefore, the number of conflict bits should be minimized to improve the encoding efficiency and the proposed method dramatically reduces the number of conflict bits using scan slice

Table 1 Selective encoding method of [5]. ($N = 16$)

Scan slice	Encoding data		Description
	Control	Data	
X11XXX1X	01	00101	Map Xs to 1, set bit 9 to 0
X0X0XX1X	10	00111	Set bit 11 to 0
0XX11XXX	00	00011	Map Xs to 0, set bit 3 to 1
X0XXXX01	10	00100	Set bit 4 to 1
	10	01111	Set bit 15 to 1

Table 2 Proposed scan slice encoding and decoding method. ($N = 16$)

Scan slice				Encoding data		Decoding data				Description
S_0	S_1	S_2	S_3	Control	Select	G_0	G_1	G_2	G_3	
X11X	XX1X	X0X0	XX1X	00	1101	1111	1111	0000	1111	Constant group mode
0XX1	1XXX	X0XX	XX01	01	0110	0101	1010	1010	0101	Variable group mode
X100	0XX0	1X1X	0X1X	00	0010	0000	0000	1111	0000	Constant group mode
				10	0001	0100	0000	1111	0000	Single mode: 1st bit data is flipped
				10	1110	0100	0000	1111	0010	Single mode: 14st bit data is flipped
01X1	XX1X	X10X	00X1	01	0100	0101	1010	0101	0101	Variable group mode
				10	1101	0101	1010	0101	0001	Single mode: 13st bit data is flipped

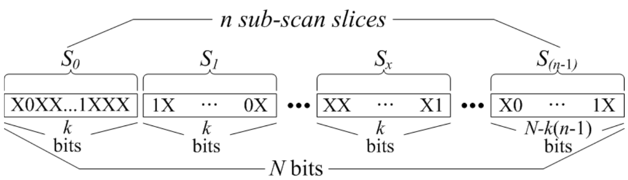

Fig. 1 Scan slice grouping method.

grouping. Figure 1 shows the scan slice grouping method of the proposed encoding technique. The scan slice composed of N bit data is partitioned into $n = \lceil \log_2 N \rceil$ sub-scan slices. Let S_x be the x-th sub-scan slice ($0 \leq x \leq n-1$). Each sub-scan slice except $S_n - 1$ is composed of k-bit data, where $k = \lceil N/n \rceil$. As shown in Fig. 1, $S_n - 1$ is composed of $N - k \cdot (n-1)$ bit data. Each sub-scan slice is encoded with the combination of n-bit select code, and 2-bit control code, therefore, the size of the encoding data is $n+2 = \lceil \log_2 N \rceil + 2$.

The selective scan slice grouping technique defines three modes, namely the constant group mode, the variable group mode, and the single mode. When each S_x is composed of all 0s or 1 s, the scan slice can be encoded with the constant group mode. In the constant group mode, the control code is encoded as 00 and each S_x is encoded with a single logic value as *select*[x]. When S_x is composed of all 0 s (1 s), *select*[x] is encoded as 0 (1). Therefore, the variable scan slice can be encoded with a single encoding data using the constant group mode, if each S_x is constant sub-scan slice. For example, Table 2 presents examples of scan slice encoding and decoding. Since $N = 16$, $n = 4$ and $k = 4$. In Table 2, the scan slice X11X XX1X X0X0 XX1X is encoded with a single encoding data using the constant group mode because S_0, S_1, S_2, and S_3 are separately constant sub-scan slices. Therefore, control code and select code are encoded as 00 and 1101, respectively.

However, the constant group mode is not enough for drastic improvement of volume reduction, since a scan slice is still variable if any S_x is variable. Therefore, another group mode is defined to improve the encoding efficiency. The variable group mode can be applied when each S_x is composed of $0101\cdots$, or $1010\cdots$. In the variable group mode, the control code is encoded as 01 and each S_x is also encoded with a single logic value as *select*[x]. When S_x is composed of $0101\cdots$ ($1010\cdots$), *select*[x] is encoded as 0 (1). Therefore, another variable scan slice can be encoded with a single encoding data, if each S_x can be encoded as the variable group mode. For example, the scan slice 0XX1 1XXX X0XX XX01 is encoded with a single encoding data, since S_0, S_1, S_2, and S_3 can be treated as 0101, 1010, 1010, and 0101, respectively.

These two group modes dramatically increase the encoding efficiency due to reduction of many conflict bits. Since the proposed method can encode both logic values with a single encoding data, the conflict bit should be redefined as follows:

- The *conflict bit* is the position of the specified bit that cannot be encoded with a single constant or variable group mode.

Because the proposed method cannot remove entire conflict bits, each conflict bit should also be encoded with separate encoding data. In the single mode, control code is encoded as 10, and the position of conflict bit is encoded with *select*[$n-1:0$]. For example, the scan slice X100 0XX0 1X1X 0X1X requires two single modes after the constant group mode. And the scan slice 01X1 XX1X X10X 00X1 requires a single mode after the variable group mode. These conflict bits should be separately encoded as presented in Table 2.

3. Decompression Architecture

The overall decompression scheme is presented in Fig. 2. To apply the test data, a scan chain is partitioned into N scan chains then scan chains are also grouped into n scan groups. Let G_x be the x-th scan group. The input selection module (ISM) decodes encoding data using either *select*[$n-1:0$] or *dec*[$N-1:0$], and the $n:N$ one-hot decoder decodes encoding data to select a single ISM to flip the present value in the single mode. ISM is composed of N ISM cells which are shown in Fig. 3. In the constant group mode (*ctrl* = 00), *select*[x] is directly inserted into the ISM cells connected to scan chains in G_x. Therefore, each G_x can be differently broadcasted as *select*[x]. To implement the variable group mode (*ctrl*=01), an XOR gate is inserted into the y-th ISM cells of G_x, where y is the odd number ($0 \leq y \leq k-1$). Therefore, the odd ISM cells and the even ISM cells in G_x can differently decode test data. When *select*[x] is 0 (1), $0101\cdots$ ($1010\cdots$) is inserted into ISM cell connected to G_x. In the single mode (*ctrl* = 10), a single $ISM\_in[i]$, which is directly connected to *dec*[i], is selected by the $n:N$ one-hot

Table 4 Comparison of test data volume for ITC'99 benchmark circuits.

Circuits	[2]				[5]				Proposed method			
	CH	*N*	T_E (bits)	*Red*(%)	*CH*	*N*	T_E (bits)	*Red*(%)	*CH*	*N*	T_E (bits)	*Red*(%)
b17	9	108	182,574	74.42	9	127	268,038	62.45	9	128	96,543	86.48
	10	120	164,472	76.96	10	255	257,460	63.93	10	256	81,160	88.63
b18	9	108	848,160	76.46	9	127	1,032,201	71.35	9	128	489,717	86.41
	10	120	836,640	76.78	10	255	988,050	72.58	10	256	335,180	90.70
b19	9	108	2,884,860	77.56	9	127	2,878,758	77.61	9	128	1,713,042	86.67
	10	120	2,931,040	77.20	10	255	2,520,330	80.39	10	256	1,082,310	91.58

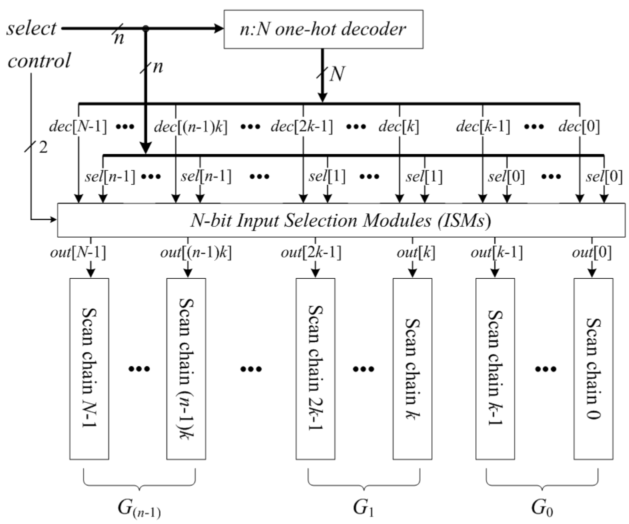

Fig. 2 The decompression architecture.

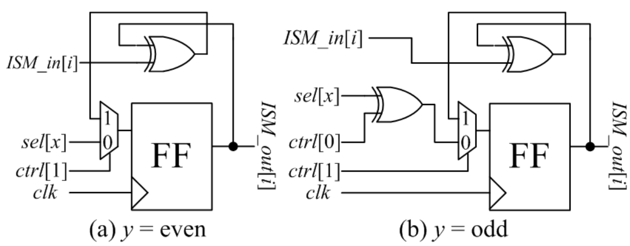

Fig. 3 The y-th ISM cell of G_x.

decoder and the present value of the i-th ISM cell is flipped, where $i = kx + y$.

To manage the scan operation, the scan shift operation should be disabled during the single mode, since the scan slice is not exactly decoded until the single mode operation is completed. Therefore, the scan clock should be blocked during the single mode ($ctrl[1] = 1$) to freeze the scan cells. A gated clock scheme can be applied to implement scan control scheme. When the single mode is completed, the scan slice is shifted to the scan chains by applying the group mode of next scan slice. Using this method, the overall scan operation can be appropriately controlled.

4. Experimental Results

In order to verify the effectiveness of the proposed method, we compared the test data volumes of the large ITC'99 benchmark circuits for various scan chain numbers. Design Compiler [7] was used to synthesize each circuit and TetraMax [8] was utilized to generate the test patterns with dynamic compaction turned on and random-fill turned off. As shown in Table 3, the number of test patterns may increase without the random-fill option since the X bits disturb additional fault dropping. In Table 3, the number of test patterns and test data volume (T_D) are presented. Although the random-fill option enables more reduction in T_D, the test data compression schemes generally do not utilize the random-fill option since the unspecified bits are very useful for obtaining high compression ratio. Moreover, the test data volume is dramatically reduced after applying the test data compression process.

Table 3 Information of ITC'99 benchmark circuits.

Circuits	No. of cells	With random-fill		W/O random-fill	
		P	$T_D(bits)$	*P*	$T_D(bits)$
b17	1,317	542	713,814	543	715,131
b18	3,064	1,176	3,603,264	1,682	5,153,648
b19	6,130	2,097	12,854,610	3,497	21,436,610

Table 4 presents the comparison of the test data volume for various test data compression techniques. CH, N, T_E, and Red denote the number of ATE channels, the number of scan chains, the size of the encoded data, and the reduction ratio, respectively. In Table 4, T_E was evaluated based on the same number of CH, since CH means the pin overhead for each compression method. The broadcast scan-based method represented as [2] shows limited volume reduction since it requires the constrained ATPG, which increases the number of original test patterns. For example, if nine ATE channels are available, the number of test patterns for b18 is 3,040, while 1,682 patterns can cover the same number of faults without the random-fill option as shown in Table 3. The conventional selective encoding technique [5] also presents lower volume reduction than the proposed method, since there are many conflict bits to be encoded. Therefore, the test data volume can be effectively compressed using the proposed technique.

When estimating the performance of the test data compression technique, hardware overhead of the on-chip decoder is another important factor. Table 5 shows the hardware overhead of the proposed method. CH, N, DEC, and ISM denote the number of ATE channels, the number of scan chains, the size of one-hot decoder, and the size of ISMs, respectively. The size of the on-chip decoder depends on the number of scan chains, since the area of the on-chip

Table 5 Area overheads of the on-chip decoder.

CH	N	DEC (NAND)	ISM (NAND)	Total (NAND)
7	32	50	340	390
8	64	96	681	777
9	128	170	1,362	1,532
10	256	309	2,726	3,035
11	512	601	5,452	6,053

decoder is mainly dedicated to the ISMs which are implemented as many as the number of scan chains. As shown in the Table 5, the proposed on-chip decoder requires acceptable area overhead compared to the large commercial circuits which often implemented with more than multi-million gate designs.

5. Conclusions

In this paper, we presented a selective scan slice grouping technique to reduce test data volume. Using the proposed method, many variable scan slices are encoded with a single encoding data and the number of conflict bits is dramatically reduced. Therefore, the proposed method can be an efficient test data compression for reducing test data volume.

References

[1] L.-T. Wang, C.E. Stroud, and N.A. Touba, System on Chip Test Architectures, Morgan Kaufmann Publishers, 2008.
[2] N. Sitchinava, S. Samaranayake, R. Kapur, E. Gizdarski, F. Neuveux, and T.W. Williams, "Changing the scan enable during shift," Proc. IEEE VLSI Test Symposium, pp.73–78, 2004.
[3] L.-T. Wang, X. Wen, H. Furukawa, F.-S. Hsu, S.-H. Lin, S.-W. Tsai, K.S. Abdel-Hafez, and S. Wu, "VirtualScan: A new compressed scan technology for test cost reduction," Proc. IEEE International Test Conference, pp.916–925, 2004.
[4] J. Rajski, J. Tyszer, M. Kassab, and N. Mukherje, "Embedded deterministic test," IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst., vol.23, no.5, pp.776–792, 2004.
[5] 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.
[6] Y. Kim, J. Park, and S. Kang, "Grouped scan slice repetition method for reducing test data volume and test application time," IEICE Trans. Inf. & Syst., vol.E92-D, no.7, pp.1462–1465, July 2009.
[7] Design Compiler User Guide, Version B-2008.09, Synopsys, Mountain View, CA, 2008.
[8] TetraMax ATPG User Guide, Version B-2008.09, Synopsys, Mountain View, CA, 2008.