

Regular Paper

Two-Stage Stuck-at Fault Test Data Compression Using Scan Flip-Flops with Delay Fault Testability

KENTAROH KATOH,^{†1} KAZUTERU NAMBA^{†1}
and HIDEO ITO^{†1}

This paper presents a stuck-at fault test data compression using the scan flip flops with delay fault testability namely the Chiba scan flip-flops. The feature of the proposed method is two-stage test data compression. First, test data is compressed utilizing the structure of the Chiba scan flip flops (the first stage compression). Second, the compressed test data is further compressed by conventional test data compression utilizing X bits (the second stage compression). Evaluation shows that when Huffman test data compression is used in the second stage compression, the volume of test data for the proposed test data compression in ATE is reduced 35.8% in maximum, 25.7% on average of the one of the test data compressed by the conventional method. The difference of the area overhead of the proposed method from the conventional method is 9.5 percent point.

1. Introduction

Increasing of the complexity of system-on-a-chip (SoC) makes the test data volume (TDV) huge. Nevertheless, in testing system-on-a-chip, the number of I/O channels, and memory size of traditional automatic test equipment (ATE) are limited. Therefore reduction of test data volume is indispensable for low cost testing. Test data compression technique is one of the attractive solutions.

In manufacturing test, both stuck-at fault test and delay fault test are important. Several scan-based stuck-at fault test data compression methods have been proposed. These conventional works compress test data using conventional coding method^{2),3)}, fan-out scan chain^{4),5)}, seed encoding method⁶⁾, or template-based encoding method^{7)–9)}. These works are standard scan based one. A delay test data compression technique with statistical coding using the enhanced scan

design was proposed¹⁰⁾.

On the other hand, recently, the scan design for delay fault testability was proposed¹¹⁾. This paper calls the scan design¹¹⁾ *Chiba scan design*. This scan design is almost equivalent to the conventional enhanced scan design from the viewpoint of fault coverage, test application time, and required memory size for ATE in spite of the same area overhead as standard scan design. However, no efficient test data compression technique under the Chiba scan design has been proposed.

This paper proposes an efficient two-stage stuck-at fault test data compression method using the Chiba scan design. The proposed method compresses test data with two-stage test data compression process. In the first stage, the proposed method compresses the test data utilizing the structure of the Chiba scan flip flops (the first stage compression). The compressed test data still remains many X bits. The X bits represent don't care bits. These bits can be assigned either 0 or 1. In the second stage, the compressed test data is further compressed utilizing these X bits (the second stage compression).

This two-stage test data compression realizes higher compression ratio than conventional test data compression using standard scan architecture. In addition, since the proposed method uses the Chiba scan design, it also reduces test cost for delay fault testing.

The rest of this paper is organized as follows. Section 2 presents a brief introduction of the Chiba scan design. Section 3 describes the detail of the proposed method. Section 4 evaluates the effectiveness of the method. Finally Section 5 concludes this paper.

2. Chiba Scan

2.1 Brief Introduction of Architecture

The Chiba scan flip flops which are used in the Chiba scan design are based on master-slave flip-flop. Unlike standard scan flip-flops, the test output does not branch from the flip-flop output, but branches from the output of master latch. Therefore, the scan path of this scan design consists of all master latches.

Figure 1 shows four Chiba scan flip-flops, $FF_0 \sim FF_3$, which construct a scan path of 4bit in length in test mode. Input *sdi* is the scan input. Output *sdo*

^{†1} Chiba University

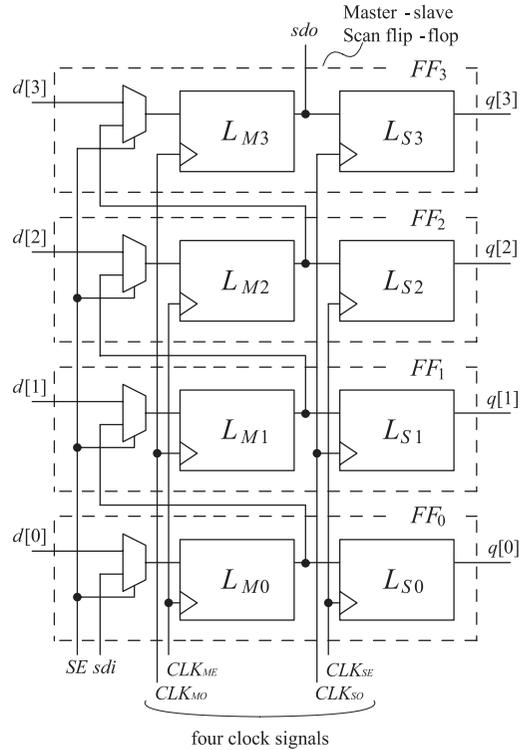


Fig. 1 Assumed scan FF with delay fault testability¹¹⁾.

is the scan output. In each FF_i ($0 \leq i \leq 3$), the master latch L_{M_i} , and the slave latch L_{S_i} construct the master slave FF_i in normal operation mode. A test input and a normal input $d[i]$ are connected to the input of L_{M_i} through a 2-to-1 selector. A normal output $q[i]$ is connected to the output of L_{S_i} . This scan design requires four clock lines, CLK_{ME} , CLK_{MO} , CLK_{SE} , CLK_{SO} . Even numbered master latches, L_{M_0} and L_{M_2} , are controlled by CLK_{ME} . Odd numbered master latches, L_{M_1} and L_{M_3} , are controlled by CLK_{MO} . Even numbered slave latches, L_{S_0} and L_{S_2} , are controlled by CLK_{SE} . Odd numbered slave latches, L_{S_1} and L_{S_3} , are controlled by CLK_{SO} . These four clock lines are independent each other.

In test mode, L_{M_0} , L_{M_1} , L_{M_2} , L_{M_3} construct a scan path whose input is sdi , and output is sdo . The pairs of (L_{M_0}, L_{M_1}) and (L_{M_2}, L_{M_3}) construct the master-slave flip-flops for scan-in and scan-out operations, respectively. The even numbered latches, L_{M_0} and L_{M_2} , play the role of a master latch, while the odd numbered latches, L_{M_1} and L_{M_3} , play the role of a slave latch. Note that the bit length of constructed shift register of this scan path is the half of the one of standard scan path. It is because the scan path is constructed by only master latches. CLK_{ME} and CLK_{MO} control scan-in and scan-out operation. For these operations, CLK_{MO} should be provided with inverse signal of CLK_{ME} .

On the other hand, in normal operation mode, each pair of (L_{M_0}, L_{S_0}) , (L_{M_1}, L_{S_1}) , (L_{M_2}, L_{S_2}) , (L_{M_3}, L_{S_3}) constructs a master-slave flip-flop, respectively. During this mode, both master clocks, CLK_{ME} and CLK_{MO} are provided the same signal. Both slave clocks, CLK_{SE} and CLK_{SO} are provided the inverse signal of the one provided into CLK_{ME} and CLK_{MO} .

This architecture has higher delay fault testability without extra latches^{*1}. Of course, it achieves complete fault coverage of stuck-at fault testing. The next subsection explains the detail of the stuck-at fault testing using the Chiba scan design.

2.2 Outline of the Scan Operation

This section explains the outline of the scan operation for one pattern testing for stuck-at fault testing. Section 2.2.1 shows the basic scan operation of the Chiba scan design. Section 2.2.2 shows the scan operation with test data reduction.

2.2.1 Basic Scan Operation

Unlike standard scan design, the Chiba scan design constructs the scan path using only master latches in test mode.

Here, let show an example to apply a 4-bit stuck-at fault test vector, (x_0, x_1, x_2, x_3) . The test response is (y_0, y_1, y_2, y_3) . The sequence of scan-in operation of the test vector, test execution, and scan-out operation of the test response is as follows.

Step 1 The even numbered bits of the test vector (x_0, x_2) are scanned in. After that, these values are set to the corresponding slave latches, L_{S_0} and L_{S_2} .

*1 The application of delay fault testing is explained in the previous work of our group¹¹⁾.

Step 2 The odd numbered bits of the test vector (x_1, x_3) are scanned in. After that, these values are set to the corresponding slave latches, L_{S1} and L_{S3} .

Step 3 The test is executed. One clock later, the test response, (y_0, y_1, y_2, y_3) , is captured to master latches, L_{Mi} ($0 \leq i \leq 3$). The test vector stored in slave latches is preserved until Step 5.

Step 4 The odd numbered bits of the test vector (y_1, y_3) are scanned out.

Step 5 The even numbered bits of the test vector (y_0, y_2) are scanned out.

As shown above, to scan-in an arbitrary test vector using the scan path using only master latches in test mode, the scan-in operation should be performed twice, first, even numbered bits are scanned in Step 1, second, odd numbered bits are scanned in Step 2.

2.2.2 Scan Operation with Test Data Reduction

The volume of test vectors satisfying the specific conditions can be compressed. For example, assigning $(L_{M0}, L_{M1}, L_{M2}, L_{M3}) = (0, 0, 1, 1)$, and $(L_{M0}, L_{M1}, L_{M2}, L_{M3}) = (0, 1, 1, 0)$ do not require to scan-in the all bits test vector $(0, 0, 1, 1)$, or $(0, 1, 1, 0)$ unlike the basic scan operation explained in the previous subsection.

Shifting-in the 2-bit test vector $(0, 1)$ once with two clock cycles assigns $(L_{M0}, L_{M1}, L_{M2}, L_{M3}) = (0, 0, 1, 1)$. On the other hand, shifting-in the 3-bit vector $(0, 1, 0)$ once with 2.5 clock cycles assigns $(L_{M0}, L_{M1}, L_{M2}, L_{M3}) = (0, 1, 1, 0)$. Therefore, 2-bit or 3-bit test vector can assign the values for these 4-bit FFs. In other words, 4-bit test vector is not required to assign the values. Therefore, the data volume of test vector can be reduced. To assign $(L_{M0}, L_{M1}, L_{M2}, L_{M3}) = (0, 0, 1, 1)$, the scan-in operation must be finished with the even numbered latches, L_{M0} and L_{M2} closed. On the other hand, to assign $(L_{M0}, L_{M1}, L_{M2}, L_{M3}) = (0, 1, 1, 0)$, the scan-in operation must be finished with the odd numbered latches, L_{M1} and L_{M3} closed. Like this example, scan operation with reduction of data volume of test vector requires two clock operation modes. One is the mode that the scan-in operation is finished with even numbered master latches closed. The other is the one that it is finished with odd numbered master latches closed. We call the former one “the even-bits scan-in operation mode”, and the latter one “the odd-bits scan-in operation mode”. In general, the test vector satisfying the condition formulated by Eq. (1) can be

scanned-in with “the even-bits scan-in operation mode” to reduce the volume of test vector.

$$L_{M(2i)} = L_{M(2i+1)} (0 \leq i \leq (l-2)/2), \quad (1)$$

where l represents the scan length.

On the other hand, the test vector satisfying the condition formulated by Eq. (2) can be scanned-in with “the odd-bits scan-in operation mode”.

$$L_{M(2i-1)} = L_{M(2i)} (1 \leq i \leq (l-1)/2), \quad (2)$$

where l represents the scan length, too.

Although, the basic scan-in operation of the Chiba scan requires two steps, this scan-in operation requires single step. The first stage compression of the proposed method uses both of these one-step and two-step scan-in operations. In this paper, we call the single scan-in operation of the Chiba scan design “one time scan-in operation”, while the two times scan-in operation of the Chiba scan design for arbitrary one pattern vector “two times scan-in operation”.

3. Proposed Two-Stage Stuck-at Test Data Compression

The proposed two-stage stuck-at test data compression consists of the first stage compression and the following second stage compression.

Figure 2 illustrates the test architecture of the proposed method. The architecture consists of ATE part and chip part. The chip part consists of three sub-parts, test data decoder part, scan chain part, and test response compaction part. Scan chains of the scan chain part consist of the Chiba scan flip-flops. The test response compaction part consists of MISR. The data decoder part consists of a decoder circuit and a scan-in operation controller. Sections 3.1 and 3.2 explain the test data compression and decompression process, briefly.

3.1 Test Data Compression

The proposed test compression mainly includes 4 steps:

Step 1 The test data TD is divided into sub-test data td_0, \dots, td_{N-1} , for scan chains, sc_0, \dots, sc_{N-1} . Each sub-test data of each scan chain td_i ($0 \leq i \leq N-1$) consists of test vectors, $v_{i0}, \dots, v_{i(M-1)}$, where M is the number of test vectors.

Step 2 The test vectors of each sub-test data are compressed using the first stage compression. The first stage compression compresses most of the

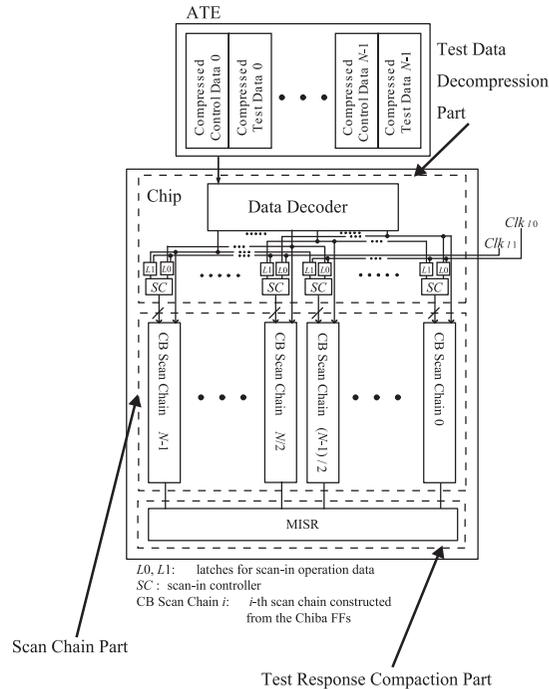


Fig. 2 Basic test architecture for the proposed test compression.

test vectors and generates the 2-bit scan-in operation control data for each test vector (the decompression process requires the scan-in operation control data).

Step 3 Add each scan-in operation control data to the corresponding compressed test vector as header.

Step 4 The second stage compression is applied to the test vectors with the header.

After these steps mentioned above, the ATE data is generated.

The proposed compression process includes the two stages mentioned above. Note that in the proposed two stage test data compression, the conventional test data compression approach utilizing X bits, such as Huffman compression, can be applied as the second stage compression.

3.2 Test Data Decompression

The decompression process also includes two stages, namely the first stage decompression and the second stage decompression. The first stage decompression is the reverse process of the second stage compression. The second stage decompression is the reverse process of the first stage compression. The decoder circuit of Fig. 2 takes the first stage decompression, while the scan-in operation controller takes the second stage decompression. The first stage decompression decodes both the scan-in operation control data and the corresponding test vector by the same decoder circuit, consecutively. Because each scan chain has 2-bit scan-in operation control data, the amount of scan-in operation control data per a test vector is twice of the number of scan chains. On the other hand, the output number of the decoder circuit is the same as the number of scan chains. Therefore, the process of decoding the scan-in operation control data is divided into two phases. The first phase decodes the scan-in operation control data of the scan chains, $sc_0 \sim sc_{(N-1)/2}$. The second phase decodes the one of the scan chains, $sc_{N/2} \sim sc_{N-1}$. The decoding sequence is as follows.

Step 1 Decode the scan-in operation control data of scan chains, $sc_0 \sim sc_{(N-1)/2}$.

Step 2 Latch the data decoded in Step 1 to corresponding latches, $L0$ and $L1$, of each scan chain. The latch operation is controlled by Clk_{i0} , which is for latches, $L0$ and $L1$, of scan chains, $sc_0 \sim sc_{(N-1)/2}$.

Step 3 Decode the scan-in operation control data of scan chains, $sc_{N/2} \sim sc_{N-1}$.

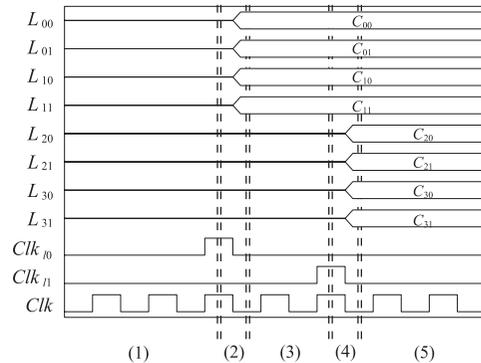
Step 4 Latch the data decoded in Step 2 to corresponding latches, $L0$ and $L1$, of each scan chain. The latch operation is controlled by Clk_{i1} , which is for latches, $L0$ and $L1$, of scan chains, $sc_{N/2} \sim sc_{N-1}$.

Step 5 Decode the corresponding test vector.

The timing of setting scan-in operation control data of test compression architecture with 4 scan chains is shown in Fig. 3. The symbols, L_{i0} and L_{i1} represent the $L0$ and $L1$ of the scan chain sc_i , respectively. The following subsections explain the proposed compression in details.

3.3 Basics of the First Stage Compression

Unlike standard scan design, the Chiba scan design constructs a scan path with



(1) Decoding scan-in operation control data for latches, $L_{00} - L_{11}$
 (2) Latching decoded scan-in operation control data for latches, $L_{00} - L_{11}$
 (3) Decoding scan-in operation control data for latches, $L_{10} - L_{21}$
 (4) Latching decoded scan-in operation control data for latches, $L_{10} - L_{21}$
 (5) Decoding 1st bit of test data

Fig. 3 Timing of setting the scan-in operation control data.

only master latches. The length of the shift register of the Chiba scan chain is the half of the one of standard scan chain. Therefore, the required amount of scan-in data for the scan-in operation is about the half of the one of standard scan operation.

As the number of test vector which can be scanned in using the one time scan-in operation (Refer to Section 2.2) increases, the required test data volume reduces. In the first stage compression, we make an effort to increase such test vectors with keeping the number of test vectors and the fault coverage.

Figure 4 shows an example of the first stage compression. This example assumes single scan chain, and test data are five test vectors with 6 bits. In Fig. 4, the original test data are shown in Fig. 4 (a), and the compressed test data are in Fig. 4 (b). The test vector noted with “1 Time” in the scan-in mode can be scanned-in with the one time scan-in operation. Test vectors, tp_1 and tp_3 are the examples. The pattern tp_1 is (X, 1, 1, 0, 0, 0). Therefore, this vector satisfies the condition formulated by Eq. (2). Thus this vector can be scanned in using the one time scan-in operation with the odd-bits scan-in operation mode. On the other hand, tp_3 is (1, X, X, X, 0, X). Therefore, this vector satisfies both

the conditions formulated by Eq. (1), and by Eq. (2). Thus this vector can be scanned in using the one time scan-in operation with either the even-bits scan-in operation mode or odd-bits scan-in operation mode. If tp_3 is scanned in with the one time scan-in operation with the even-bits scan-in operation mode, the test data volume is reduced from 30 bits to 25 bits.

The volume of the original test data TP is described as Eq. (3):

$$TP = \sum_{i=0}^{N-1} Ml_i, \tag{3}$$

where M is the number of test vector, N is the number of scan chains, and l_i is the original test vector length of the i th scan chain.

The lower bound of the length of compressed test vector in the first stage compression is the half of the one before the compression, which yields the following inequility.

$$TP' \geq \sum_{i=0}^{N-1} Ml_i/2. \tag{4}$$

3.4 Improving the First Stage Compression Ratio with Flip-Flops Re-ordering

The compression ratio of the first stage compression is improved using the reordering approach of the scan flip-flops. Figure 4 and **Fig. 5** explain it. Note that in these figures, $bit_i(0 \leq i \leq 5)$ represents the value stored in i -th flip flop, FF_i .

In Fig. 4, the scan chain is routed in the order of $FF_0, FF_1, FF_2, FF_3, FF_4, FF_5$ sequentially. In this routing, although tp_1 and tp_3 can be scanned in using the one time scan-in operation, while tp_0, tp_2, tp_4 cannot be scanned in using the one time scan-in operation. tp_0, tp_2, tp_4 requires two times scan-in operation. As shown in Fig. 5, however, if scan chain is re-routed in the order of $FF_0, FF_2, FF_1, FF_3, FF_5, FF_4$ sequentially, tp_0, tp_1, tp_2, tp_3 can be scanned in using the one time scan-in operation. Only tp_4 requires two times scan-in operation. In this example, the test data volume is reduced from 25 bits to 19 bits. As can be seen, re-ordering of flip-flops can improve the first stage compression ratio.

No. of test vectors	Bit of test vector						Scan-in Mode	Bit of test vector					
	<i>bit</i> ₀	<i>bit</i> ₁	<i>bit</i> ₂	<i>bit</i> ₃	<i>bit</i> ₄	<i>bit</i> ₅		<i>bit</i> ₀	<i>bit</i> ₁	<i>bit</i> ₂	<i>bit</i> ₃	<i>bit</i> ₄	<i>bit</i> ₅
<i>tp</i> ₀ :	1	0	1	0	X	X	→ 2 Times	1	0	1	0	X	X
<i>tp</i> ₁ :	X	X	1	1	0	0	→ 1 Time Odd	X	1	0	0		
<i>tp</i> ₂ :	0	1	0	1	0	0	→ 2 Times	0	1	0	1	0	0
<i>tp</i> ₃ :	X	X	X	X	0	X	→ 1 Time Even	1	X	0			
<i>tp</i> ₄ :	1	1	0	1	1	0	→ 1 Time Odd	1	X	0	X		
							→ 2 Times	1	1	0	1	1	0

(a) Original test data (b) Compressed test data
Fig. 4 An example of the first stage compression.

No. of test vectors	Bit of test vector						Scan-in Mode	Bit of test vector					
	<i>bit</i> ₀	<i>bit</i> ₂	<i>bit</i> ₁	<i>bit</i> ₃	<i>bit</i> ₅	<i>bit</i> ₄		<i>bit</i> ₀	<i>bit</i> ₂	<i>bit</i> ₁	<i>bit</i> ₃	<i>bit</i> ₅	<i>bit</i> ₄
<i>tp</i> ₀ :	1	0	1	0	X	X	→ 1 Time Even	1	0	X			
<i>tp</i> ₁ :	X	X	1	1	0	0	→ 1 Time Odd	X	1	0	0		
<i>tp</i> ₂ :	0	1	0	1	0	0	→ 1 Time Even	0	1	0			
<i>tp</i> ₃ :	X	X	X	X	0	X	→ 1 Time Even	1	X	0			
<i>tp</i> ₄ :	1	0	1	1	0	1	→ 1 Time Odd	1	X	X	0		
							→ 2 Times	1	0	1	1	0	1

(a) Original test data (b) Compressed test data
Fig. 5 An example of re-ordering for improving the first stage compression ratio.

3.5 Construction of the Scan-in Operation Control Data

The decompression process of the test data compressed using the first stage compression requires the information of the scan-in operation. Therefore, the scan-in operation of each test vector is encoded when the first stage compression is performed. Here, the encoded data is called scan-in operation control data. Because the proposed method uses three scan-in operations, 2-bit data is used to encode them.

Table 1 shows the encode method of the scan-in operation control data. Each scan-in operation of each test vector is encoded to 2-bit scan-in operation control data. The first column represents the scan-in operation. The second column represents the corresponding scan-in operation control data. Most of test data can be applied multiple scan-in operation for unspecified X bits in test data. Scan-in operation of this test data is encoded to xx, which is unspecified bit of scan-in operation control data.

Table 2 shows a construction of scan-in operation control data from the test data. The first column represents scan chains. This is four 4-bit test vectors for a test architecture with four scan chains. The second column represents test data of each scan chain. The third column represents scan-in operation control data for each corresponding test data.

The test data of Scan Chain 1 “0011” can be scanned in using the one time scan-in operation with the even-bits scan-in operation mode. Since its scan-in operation control data is 00 according to Table 1, |00| is added to the scan-in operation control data. The test data of Scan Chain 2 “0101” cannot be scanned in using the one time scan-in operation but be scanned in using the two times scan-in operation. Therefore, since its scan-in operation control data is 01 according to Table 1, the scan-in operation control data is updated to |00|01|. The test data of Scan Chain 3 “00xx” can be scanned in using the one time scan-in operation with either the even-bits or the odd-bits scan-in operation mode.

Table 1 Encode table for construction of scan-in operation control data.

Scan-in Op.		Control Code
One Time	Even	00
	Odd	11
Two Times		01
Don't Care		xx

Table 2 An example of construction of scan-in operation control data.

	Test Data	Scan-in Op.
Scan Chain 1	0011	00
Scan Chain 2	0101	01
Scan Chain 3	00xx	xx
Scan Chain 4	1x1x	xx
Scan Cont. Data	00 01 xx xx	

Since its scan-in operation control data is xx according to Table 1, the scan-in operation control data is updated to |00|01|xx|. The test data of Scan Chain 4 “1x1x” is like the case of the data of Scan Chain 3. Finally the scan-in operation control data is |00|01|xx|xx|.

As shown in this example, scan-in operation control data has unspecified bits.

3.6 Scan-in Operation Controller

This subsection describes the architecture of the scan-in operation controller.

Figure 6 illustrates an example of the scan-in operation controller. This controller is for the test architecture with four scan chains of 4 bits in length. This structure consists of a scan chain part and a clock controller part. The clock controller has four sets of 2-bit input for four scan chains. This input is connected to 2 latches, $L0$ and $L1$, which store scan-in operation control data. Each 2-bit input controls the test clock cycles of the corresponding scan chain. In case of this example, the one time scan-in operation with the even-bits scan-in operation mode requires 2 test clock cycles. The one time scan-in operation with the odd-bits scan-in operation mode requires 2.5 test clock cycles. The two times scan-in operation requires 4 test clock cycles. Therefore, this controller provides 2 test clock cycles if 2-bit input is 00, 2.5 test clock cycles if 2-bit input is 11, and 4 test clock cycles if 2-bit input is 01.

The clock controller part consists of 3 bit counter, logic network, three D-latches

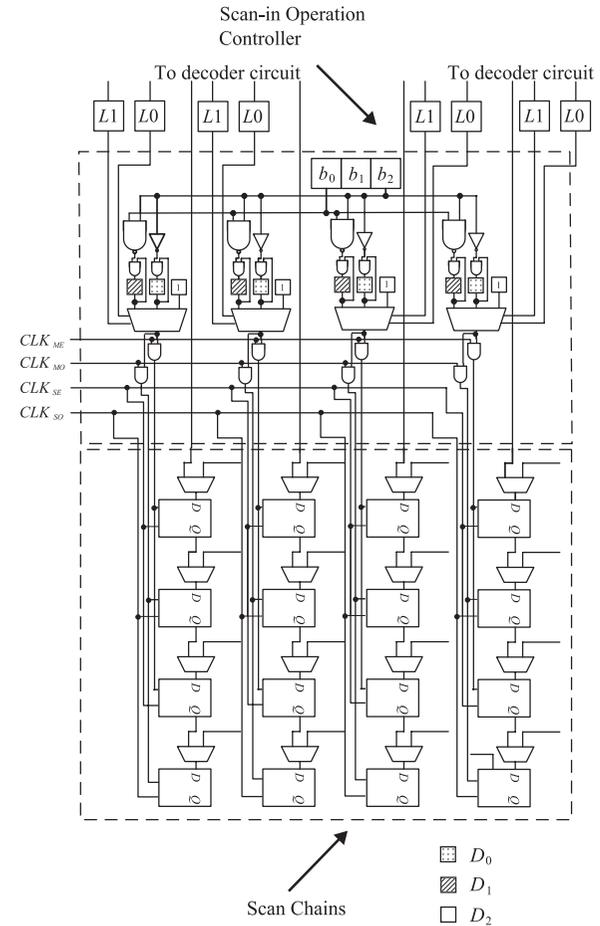


Fig. 6 Structure of scan-in operation controller.

connected to the 3-to-1 selector, D_0 , D_1 , D_2 , a 3-to-1 selector, and four AND gates controlling the clock signal from the upper part to the lower part in this figure.

MSB of the 3 bit counter is b_2 , while LSB is b_0 . Initial values of 3 D-latches are all 1, while the ones of the 3 bit counter are 0. The clock frequency of 3 bit

counter is two times faster than the one of normal test clock. Thus, 2 test clock cycles are equal to 4 counter clock cycles, while 2.5 test clock cycles are equal to 5 counter clock cycles.

The scan-in operation is selected by controlling the 3-to-1 selector. When the counter value is 4, the value 0 is captured in D_0 . Thus, the clock signal of flip-flops is disabled after 2 test clock cycles. Therefore, if the inputs connected to D_0 are selected, the controller works as the one for the one time scan-in operation with the even-bits scan-in operation mode.

When the counter is 5, the value 0 is captured in D_1 . Thus, the clock signal of flip-flops is disabled after 2.5 test clock cycles. Therefore, if the inputs connected to D_1 are selected, the controller works as the one for the one time scan-in operation with the odd-bits scan-in operation mode.

The value 1 is stored in D_2 constantly, and thus the clock is always provided into flip-flops. Therefore, if the inputs connected to D_2 are selected, the controller works as the one for the two times scan-in operation.

4. Evaluation

Efficiency of the proposed method is evaluated with respect to the data reduction ratio of the whole two-stage compression process, the specification of the first stage compression and the area overhead. Section 4.1 evaluates the compression ratio of the proposed two-stage compression. The compression ratio of the second stage compression depends on the volume and the care bit rate of test data after the first stage compression. Section 4.2 evaluates the volume and the care bit rate of the test data after the first stage compression. Finally, Section 4.3 evaluates the area overhead of the scan-in operation controller. In these evaluations, the four largest ISCAS'89 benchmark circuits, s13207, s15850, s35932, s38584, are used.

In this evaluation, Huffman test data compression is applied as the second stage compression of the proposed method as a case study. Several Huffman test data compression has been proposed^{12),13)}. Here, we treat the typical Huffman test data compression. The block length of the Huffman coding is 4 or 8. The scan length is 1, 8, 16, 32, or 64. In the first stage compression, routing of scan chains are optimized using 20,000 times random optimization. In each circuit,

we consider four cases of filling the unspecified bits, filling the parameter xx of scan-in operation control data with all 00 or all 11, and the X bit of the test data compressed by the first step compression with all 0 or all 1. Considering the range of each parameter, the combination of these parameters is the product of the combination of the scan length, the value for xx bits, and the value for X bits in each block length. Therefore, it is 20 cases in each block length of each circuit. The test compression is performed under each condition, and we choose the best one as the data of each block length of each circuit. These best data are compared with the one compressed by the conventional Huffman test data compression using standard scan architecture. The order of flip-flops of the conventional method is the same as the proposed one to evaluate the effect of the proposed two-stage compression.

4.1 Data Reduction Ratio

In this subsection, the efficiency of the proposed test compression is evaluated compared with the conventional Huffman test data compression. We evaluate the reduction of the ATE data compressed by the proposed two-stage compression compared with the ATE data compressed by the conventional method. For the evaluation, the data reduction ratio is introduced. The data reduction ratio is calculated by the following formula:

$$1 - \frac{(\text{TD}'' \text{ of the proposed method})}{(\text{TD}'' \text{ of the conventional method})}, \quad (5)$$

where TD'' represents the volume of ATE data.

In the evaluation, the compression ratios of the first and the second stage compression are also evaluated compared with the conventional method. The compression ratios are calculated by the following formula:

$$1 - \frac{(\text{the volume of the compressed data})}{(\text{the volume of the original data})}. \quad (6)$$

Table 3 shows the result. The first column represents benchmark circuit name. The second column TD shows the test data volume before the compression. The third column BL shows the block length of the Huffman coding. The "Conventional" column shows the results of the conventional Huffman test data compression. The first subcolumn TD'' shows the volume of the compressed ATE data. The second subcolumn CR shows the percentage of the compression ratio of the

Table 3 Result of data reduction ratio.

Circuit	TD	BL	Conventional		Proposed							DR		
			TD"	CR	#S	SF	DF	TD'	TD"	1 st CR	2 nd CR			
s13207	165,200	4	53,124	67.8	8	00	0	89,581	34,773	45.8	61.2	34.5		
		8	36,869	77.7	8	00	0	89,581	27,687	45.8	69.1	24.9		
s15850	76,986	4	31,997	58.4	16	00	0	41,419	24,510	41.0	46.1	23.4		
		8	26,706	65.3	16	00	0	41,419	22,533	41.0	50.4	15.6		
s35932	28,208	4	22,456	20.4	64	11	1	15,880	14,420	36.4	19.6	35.8		
		8	19,936	29.3	64	11	1	15,880	13,004	36.4	27.5	34.8		
s38584	199,104	4	89,296	55.5	32	00	0	110,140	69,698	40.3	41.4	21.9		
		8	77,196	61.2	32	00	0	110,140	65,862	40.3	44.6	14.7		
Average		-	54.5		-							40.9	45.0	25.7

TD, test data volume of original test data; TD", ATE data

TD', test data volume of test data after the first stage compression

SF, filled value of xx bits of scan-in operation control data

DF, filled value of X bits of test data after the first stage compression

1stCR, compression ratio of 1st stage compression; 2ndCR, compression ratio of 2nd stage compression

conventional Huffman test data compression. The compression ratio is calculated using Eq. (6), where the volume of the original data is TD, and the volume of the compressed data is TD" of the conventional method. The "Proposed" column shows the results of the proposed two-stage test data compression. The subcolumns, #S, SF, DF, TD', TD" show the number of scan chains, the value with which filled xx bits of the scan-in operation control data, value with which filled the X bits of the test data after the first stage compression, the volume of the test data compressed by the first stage compression, and the volume of test data compressed by the second stage compression i.e. Huffman test data compression, respectively. The columns, 1stCR and 2ndCR show the percentage of the compression ratio of the first stage compression and the second stage compression, respectively. The former is calculated using Eq. (6) where the volume of the original data is TD, and the volume of the compressed data is TD'. The latter is calculated using Eq. (6) where the volume of the original data is TD', and the volume of the compressed data is TD" of the proposed method. The column DR shows the percentage of the data reduction ratio.

The maximum data reduction ratio is 35.8%. The average and minimum is 25.7% and 14.7%, respectively. The number of scan chains that get the best result is different in each circuit. The data reduction ratio in case that the

block length is 8 is smaller than the one in case that the block length is 4 in all evaluated circuits. In the worst case, the decrease of the reduction ratio is 9.6%. The maximum and minimum first stage compression ratio is 45.8% and 36.4%, respectively. The average ratio is 40.9%. From the average value, about 80% of test vectors are scanned in using the one time scan-in operation. For the pre-compression process using the first stage compression, the compression ratio of the second stage Huffman compression of the proposed method is lower than the one of the conventional method. In the worst case, the decrease is 16.6% (s38584). The average and the minimum are 9.5%, 0.8% (s35932), respectively. However the ATE data volume of the proposed method is smaller than the one of the conventional method in all evaluated circuits. The compression ratio of the first stage compression is lower than the one of the standard Huffman compression. To achieve better compression than the standard Huffman compression, both the first and second stage compression are required.

4.2 First Stage Compression

This subsection evaluates the first stage compression in detail. Usually, compression ratio of a test data compression method depends on the care bit rate of the original test data. Thus, first, the relation between the care bit rate of the original test data and the compression ratio of the first stage compression is

Table 4 Data volume and care bit rate of each test data.

Circuit	TD		TP'		TC'		TD'		r.t.c.	r.c.b.	1 st CR	2 nd CR
	VL	CB	VL	CB	VL	CB	VL	CB				
s13207	165,200	6.9	85,805	12.9	3,776	9.2	89,581	12.7	4.2	5.8	45.8	69.1
s15850	76,986	16.4	41,419	28.9	4,032	23.7	45,451	28.4	8.9	12.0	41.0	50.4
s35932	28,208	64.7	15,880	69.8	2,048	31.6	17,928	65.4	11.4	0.7	36.4	27.5
s38584	199,104	17.7	110,140	30.7	8,704	26.0	118,844	30.4	7.3	12.7	40.3	44.6
Average									8.0	7.8	41.0	47.9

$$TD' = TP' + TC', \text{ r.t.c.} = (\text{VL of } TC')/(\text{VL of } TD'), \text{ r.c.b.} = (\text{CB of } TD') - (\text{CB of } TD)$$

evaluated. After the first stage compression, the scan-in operation control data is generated. Second, the specification of this data is evaluated. The compression ratio of the second stage compression depends on the care bit rate of the test data compressed by the first stage compression. Third, the relation between the care bit rate of the data compressed by the first stage compression and the compression ratio of the second stage compression is evaluated.

Table 4 shows the result. The results are obtained under the conditions of Table 3 when the block length of Huffman coding is 8. The first column represents benchmark circuit name. The second and the third columns show the number of test vector and scan chain, respectively. The column TD shows the specification of the original data. The column TP' shows that of the test data applied the first test compression. The column TC' shows that of the scan-in operation control data. The column TD' shows that of the data made by merging the compressed data with the scan-in operation control data. Each above column has the subcolumns VL and CB. The subcolumn VL shows the data volume, while CB shows the care bit rate. The column r.t.c. shows the rate of the volume of TC' in the volume of TD'. The column, r.c.b. shows the increase of the care bit rate of test data due to the first stage compression. The unit of r.t.c. and r.c.b. is percent and percent point, respectively. The columns, 1stCR and 2ndCR show the percentage of the compression ratios of the first stage compression and the second stage compression, respectively. Note that in this evaluation, the block length of the second stage compression is 8. As said in the previous subsection, the average of the first stage compression ratio is 40.9%. The compression ratio of the first stage compression decreases as the care bit rate of the original test data increases. The compressed data TD' is divided into the set of test vectors

TP' and the set of scan-in operation control data TC'. Because the 2-bit scan-in operation control data is assigned each test vectors, the volume of the set of scan-in operation control data is calculated by two times of the product of the number of test vector and the number of scan chains. Although the volume of the set of scan-in operation control data gives bad influence on the second stage compression, the r.t.c. is 8.0% on average.

The care bit rate of test data increases for the first stage compression. The maximum r.c.b. is 12.7 percent point (s38584). On the other hand, the minimum is 0.7 percent point (s35932). The average is 7.8 percent point. This is the reason why the compression ratio of the Huffman compression of the proposed method is lower than the one of the conventional method.

In every evaluated circuit, the care bit rate of TC' is smaller than the ones of TP'. This result means that the unspecified bit-rate of TC' is higher than that of TP'. Therefore, we can conclude that the care bit rate of TD' is smaller than TP' for TC'. It gives good influence into the Huffman compression of the second stage compression. In this evaluation, for the Huffman compression, the unspecified bits of TC' are filled with simply all 00 or all 11. If a test vector is scanned in using one time scan-in operation, the required scan-in data volume and care bit rate changes in its scan-in mode. Thus, filling the unspecified bits with all 0 or all 1 does not guarantee the best compression ratio. Therefore, much more optimized filling of unspecified bits will exist.

Table 5 shows the effect of the re-ordering to the first compression. The column DF shows the volume of test data applied the first compression when the re-ordering is not applied. On the other hand, RE shows the volume of test data applied the first compression when the re-ordering is applied. The last column

Table 5 Effect of re-ordering of scan chains.

Circuit	DF	RE	RE/DF
s13207	90,712	85,805	94.6
s15850	45,468	41,419	91.1
s35932	16,467	15,880	96.4
s38584	118,615	110,140	92.9
Average	-	-	93.7

shows the percentage of RE/DF. As shown in this result, the re-ordering reduces the compressed data 6.3% on average.

4.3 Area Overhead

Finally the area overhead of the proposed test compression architecture is evaluated. The area of the proposed test compression architecture is compared with the one of the conventional test compression architecture. Both of them consist of test data decoder part, scan chain part, and test response compaction part. The test data decoder part of the conventional test compression architecture consists of the data decoder circuit, while the one of the proposed architecture consists of the data decoder circuit, scan-in operation controller, and the latch circuits for capturing the inputs value. The scan chain part of the conventional test compression architecture consists of standard scan chains, while the one of the proposed architecture consists of the Chiba scan chains. The test response compaction parts of both architectures consist of MISR. The area of the Chiba scan design is the same as standard scan design. The proposed decoder circuit is different from the standard one, because the encoded data is different, and the additional control for the decode process. The additional area for the proposed architecture is the area of the scan-in operation controller and the latch circuits for capturing the inputs value. Therefore, the area of the decoder circuit, the scan-in operation controller, and the latch circuits are evaluated. To evaluate the required hardware resources, the decoder circuit, the scan-in operation controller, and the latch circuits are implemented in verilog and synthesized with Synopsys design compiler using Rohm 0.35 μm standard cells. The results are shown in **Table 6**. The first column represents benchmark circuit name. The column circ. shows the area of the benchmark circuit without any DFT. The column scan cont. shows the area of the proposed scan-in operation controller. The column

Table 6 Synthesis result ($\times 10^{-3}\text{mm}^2$).

	circ.	scan cont.	latch	dec. prp.	dec. std.	area. prp.	area. std.
s13207	179.3	10.0	3.6	37.0	25.7	230.0	205.0
s15850	386.5	13.8	7.1	43.5	31.1	450.9	417.5
s35932	1,029.1	57.7	28.5	90.8	63.8	1,206.2	1,093.0
s38584	1,182.4	19.9	14.3	60.5	42.5	1,277.1	1,224.9

circ., the area of benchmark circuits without DFT

scan cont., the area of the proposed scan-in operation controller

latch, the area of latches put in the inputs of the scan-in controller

dec. prp., the area of the proposed Huffman decoder circuit

dec. std., the area of the standard Huffman decoder circuit

area. prp. = circ. + scan cont. + latch + dec. prp., area. std. = circ. + dec. std.

latch shows the area of latches put in the inputs of the scan-in controller. The columns dec. prp. and dec. std. show the area of the proposed and the standard Huffman decoder circuit, respectively. The area prp. and area std. show the whole area of the circuits applied the proposed and standard test compression. The whole area of the circuit applied the proposed compression is the sum of the area of the benchmark circuit, scan-in operation controller, the latches, and the proposed Huffman decoder circuits. The whole area of the circuit applied the standard compression is the sum of the benchmark circuit and the standard Huffman decoder circuits. Because the latches are put to the scan-in operation controller of each scan chain, the area is in proportion to the number of scan chains. Because the input data is different, and additional control is required for the proposed data decode process, the area of the proposed Huffman decoder is different from that of the conventional one. In every circuit, it is larger than that of the standard one. **Table 7** shows the area overhead of the proposed and the standard compression, and the difference of them.

The area overhead of the proposed compression is calculated by the formula, $(\text{area prp.}/\text{circ.} - 1) \times 100.0$. On the other hand, the area overhead of the standard compression is calculated by the formula, $(\text{area std.}/\text{circ.} - 1) \times 100.0$.

The first column of the table represents benchmark circuit name. The columns a.o. prp. and a.o. std. show the area overhead of the proposed method and the standard method, respectively. The column diff shows the difference between the overhead of the proposed method and the standard one. From this result, the average of the maximum difference of area overhead is 9.5 percent point.

Table 7 Area overhead.

	a.o. prp.	a.o. std.	diff.
s13207	28.2	14.3	13.9
s15850	16.7	8.0	8.6
s35932	17.2	6.2	11.0
s38584	8.0	3.6	4.4
Average	17.5	8.0	9.5

a.o. prp. = (area prp./circ. - 1)*100.0 (%)

a.o. std. = (area std./circ. - 1)*100.0 (%)

diff. = a.o. prp. - a.o. std. (percent point)

5. Concluding Remarks

This paper has presented a stuck-at fault test data compression method using the Chiba scan flip-flops. The proposed method compresses test data with the two-stage test data compression. First, test data is compressed utilizing the structure of the Chiba scan flop flops. Second, the compressed test data is further compressed utilizing remaining X bits. The two-stage test data compression realizes higher compression ratio than conventional single stage test data compression using standard scan architecture.

Evaluation shows that the data reduction ratio of the proposed two-stage test data compression is 35.8% in maximum, 25.7% on average, when the Huffman test data compression is applied as the second stage compression. The difference of the area overhead of the proposed method from the conventional method is 9.5 percent point.

We conclude that the Chiba scan structure is useful for not only delay fault testability but also test data compression.

One future work is the quantitative evaluation of the effect of the proposed two stage test data compression when other approaches are applied as the second stage compression. Another future work is test data compression for delay fault testing.

Acknowledgments This research was partially supported by the Grant-in-Aid for Scientific Research (C) No.19560335. This work is supported by VLSI Design and Education Center(VDEC), the University of Tokyo in collaboration with Synopsys, Inc. The VLSI chip in this study has been fabricated in the chip

fabrication program of VLSI Design and Education Center (VDEC), the University of Tokyo in collaboration with Rohm Corporation and Toppan Printing Corporation.

References

- 1) Wang, L., Wu, C. and Weng, X.: *VLSI Test Principles And Architectures: Design for Testability*, p.808, Morgan Kaufmann Pub (2006).
- 2) Chandra, A. and Chakrabarty, K.: Test data compression for system-on-a-chip using Golomb codes, *18th IEEE VLSI Test Symposium*, pp.113–120 (2000).
- 3) Chandra, A. and Chakrabarty, K.: A unified approach to reduce SOC test data volume, scan power and test time, *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, Vol.22, No.3, pp.352–362 (2003).
- 4) Hamazaoglu, I. and Patel, J.H.: Reducing test application time for full scan embedded cores, *29th IEEE International Symposium on Fault Tolerant Computing*, pp.260–267 (1999).
- 5) Markar, S.: A layout-based approach for ordering scan chain flip-flops, *29th IEEE International Test Conference*, pp.341–347 (1998).
- 6) Balakrishnan, K.J. and Touba, N.A.: Improving encoding efficiency for linear decompressors using scan inversion, *35th IEEE International Test Conference*, pp.936–944 (2004).
- 7) Arslan, B. and Orailoglu, A.: Circular scan: A scan architecture for test cost reduction, *7th IEEE Design Automation and Test in Europe*, pp.1290–1295 (2004).
- 8) Arslan, B. and Orailoglu, A.: Test cost reduction through a reconfigurable scan architecture, *35th IEEE International Test Conference*, pp.945–952 (2004).
- 9) Reda, S. and Orailoglu, A.: Reducing test application time through test data mutation encoding, *5th IEEE Design Automation and Test in Europe*, pp.387–393 (2002).
- 10) Namba, K. and Ito, H.: Interleaving of delay fault test data for efficient test compression with statistical coding, *15th IEEE Asian Test Symposium*, pp.389–394 (2006).
- 11) Namba, K. and Ito, H.: Scan design for two-pattern test without extra latches, *IEICE Trans. Inf. & Syst.*, Vol.E88-D, No.12, pp.2777–2785 (Dec. 2005).
- 12) Jas, A., Ghosh-Dastidar, J., Ng, M., and Touba, N.A.: An efficient test vector compression scheme using selective Huffman coding, *IEEE Trans. Comput.-Aided Des. of Integr. Circuits Syst.*, Vol.22, No.6. pp.797–806 (2003).
- 13) Iyengar, V., Chakrabarty, K., and Murray, B.T.: Built-in self testing of sequential circuits using precomputed test sets, *16th IEEE VLSI Test Symposium*, pp.418–423 (1996).

(Received December 25, 2007)

(Revised March 17, 2008)

(Accepted May 1, 2008)

(Released August 27, 2008)

(Recommended by Associate Editor: *Xiaoqing Wen*)



Kentaroh Katoh received the B.E. and M.E. degree from Nagoya University, Japan in 1997 and 1999, respectively. He joined in Fujitsu Limited in 1999 and engaged in the development of embedded control system of HDD from 1999 to 2001. Currently he is enrolled in a Ph.D. program at the Graduate School of Science and Technology, Chiba University. His research interests include testing method and fault-tolerant design of reconfigurable hardware and SoC. He is a member of the IEICE and the IEEE.



Kazuteru Namba received B.E., M.E. and Ph.D. from Tokyo Institute of Technology in 1997, 1999 and 2002, respectively. He joined Chiba University in 2002. He is currently an Assistant Professor of Graduate School of Advanced Integration Science, Chiba University. His current research interests include dependable computing. He is a member of the IEEE, the IEICE and the IPSJ.



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 Graduate School of Advanced Integration Science. His research interests include easily testable VLSI design, defect-tolerant VLSI design, VLSI architecture, fault-tolerant computing, and dependable computing. He is a Fellow of the IEICE and a member of the IEEE and the IPSJ.