ISSN 1330-3651 (Print), ISSN 1848-6339 (Online) https://doi.org/10.17559/TV-20180718044904
Original scientific paper

Test Stimuli Segmentation and Coding Method

Chuandong CHEN, Haibo LUO, Yuxiang CHEN

Abstract: Test vector coding and data transmission are the key technologies in the design-for-test of digital integrated circuits (IC). Existing parallel input methods of test stimuli can reduce test application times; however, they need to occupy multiple input ports. Thus, a novel method of test stimuli coding and data transmission was proposed to reduce the test application time of the test vectors and reduce the number of input ports required for the parallel input of test stimuli. This method was based on the segmentation of test stimuli. First, the test stimuli were evenly segmented into eight-bit wide. Second, the eight-bit data of each segment were encoded to the five-bit data according to the compatibility between the test data of each segment. The eight-bit test stimuli input can be completed in one or two clock cycles of automatic test equipment (ATE) by using the five input ports of the chip. The corresponding decoding circuit was added inside the netlist of the circuit to realize the rapid input of the test stimuli. Lastly, the ISCAS'89 benchmark circuit was used to conduct experiments, results of this coding method were then compared with those of the serial input method. Results show that the encoding method proposed in this study can save an average of 37% of the parallel input data width and 81.7% of the test stimuli input time. The proposed method in this study can also reduce the test application time and the cost of the IC test. The findings of this study can provide guidance for improving the scan testing method of digital IC.

Keywords: design-for-test; scan testing; test stimuli; test vectors

1 INTRODUCTION

The integration density, testing difficulty, and testing cost of integrated circuits (IC) have gradually increased with the development of IC process technology. In relation to this, testing has become a key factor restricting the design and application of IC [1, 2]. The design-for-test (DFT) based on the scan chain is one of the most widely used technologies in this field [3, 4]. Moreover, despite the increase in the chip integration scale of the system-on-chip (SoC), the number of chip pins that can be used for testing remains limited, leading to the gradual increase of SoC testing difficulty and testing costs [5, 6]. Therefore, low-cost testing technology has become a hot topic in the DFT of SoCs.

The scan testing method of digital IC completes all test vectors using the automatic test equipment (ATE). The testing of a test vector must undergo three processes, namely, test stimulus scan-in, test response generation, and test response scan-out [7, 8]. The two main factors that affect the cost of the digital IC testing are test data storage overhead and test application time overhead [9, 10]. The test vector compressed coding method can reduce the storage cost of test data, but the transmission speed of test stimuli and test response have limitations.

The commonly used test vector compressed coding methods include pattern run-lenth (PRL) code [11,12], Huffman code [13], false discovery rate (FDR) code [14, 15], nine-coded (9C) code [16], block merging (BM) code [12, 17], and extended frequency-directed run-length (EFDR) code [18, 19]. These methods can effectively achieve good compression effect and reduce the storage cost of test data. However, the serial input and decoding operation of the test data takes almost 50% of test application time to complete the test vector testing after the former is compressed and coded. Therefore, the test vector compressed coding methods and the corresponding data transmission methods can effectively reduce the storage overhead of the test data, but they have shortcomings in reducing test application time. For the abovementioned reasons, the coding method of test vectors and the transmission method of test data must be improved to reduce the test application time of the test vectors.

A novel test stimulus coding and test data transmission method is proposed in this study to reduce the test application time of test vectors and reduce the number of input ports required by the test stimulus parallel input. In one or two clock cycles of ATE, eight bits of test stimuli can be rapidly scanned in using only five input ports of the chip. This approach significantly reduces the number of clock cycles of the test stimulus scan-in and shortens the test application time of IC.

The remainder of this study is arranged as follows. Section 2 introduces the research status of test data coding and transmission method. Section 3 introduces the segmentation method of test stimuli and explains the coding method and corresponding decoding method of the segmented data in detail. Section 4 carries out experiments using the ISCAS'89 benchmark circuit to analyze and discuss the experimental data. Section 5 summarizes the contents of the research and provides the conclusion.

2 STATE OF THE ART

Test vector coding and the corresponding test data transmission method are two key factors that affect the cost of IC testing, and are considered the core problems of digital IC scan testing. Scholars have proposed several solutions to solve this problem. For example, Sivanandam et al. [20] proposed a test data compression strategy based on scan chain technology. This method could compress the test data and reduce the test application time under the small area overhead, but it failed to propose improvement measures on test data transmission after compression. Papameletis et al. [21] introduced a three-dimensional IC testability design scheme based on test data compression, multiple dies, and intellectual property (IP) core reuse technology, but they did not propose an optimization strategy for the transmission mode of the test data after compression. Czysz et al. [22] divided the test data into blocks according to their characteristics with the aim of maintaining the test fault coverage. They then applied the test data reuse technology to reduce the frequency of data

transmission from the ATE interface, improve the compression rate of the test data, and reduce the test application time. However, they did not propose improvement measures on test data transmission after compression; they also encountered limitations on the optimization effect of the test application time. Kuang et al. [19] proposed an improved EFDR coding compression method and introduced improvement measures based on EFDR coding. The hardware cost of the decoding circuit is low, and the average compression rate could reach 68.01%. However, the method used the traditional test data transmission mode. Thus, improvement on the test application time was not apparent. Cheng et al. [23] alternately encoded 0 run-length and 1 run-length in the test vectors and proposed a test data compression method of double-run alternation. The decompression circuit in their proposed method was simple, and the compression rate of the test data was high. However, once again, the method did not propose an improvement strategy for test data transmission after compression. Thus, improvement on the test application time was not apparent. Liu et al. [24] introduced the cyclic shift processing method and a test data compression scheme, which they applied to multiple scan chains. The method improved the compatibility and direction compatibility between vectors based on the goal of retaining the don't care bits in the vectors and further improved the compression rate of the test data. However, this approach applied the traditional test data transmission mode. Thus, it failed to improve the test application time. Wu et al. [25] proposed a data block merging-based SoC test data compressed coding method using the compatibility of test data. The corresponding decompression circuit was easy to implement, enabling this method to solve the problem of IP core scan testing in SoCs. However, although the average compression ratio of the test data of the technology reached 68.02%, it did not propose an optimization strategy for the data transmission after compression and failed to provide a positive effect in reducing test application time. Seo et al. [26] proposed a three-state coding-based test data compression method using the last test data to generate the next one. The method did not require additional input ports, which further improved the compression ratio of the test data. However, similar to previously proposed methods, it also did not improve data transmission. On the premise of not changing the coverage rate of the test fault, Yuan et al. [27] introduced a count compatible pattern run-length (CCPRL) compressed coding method for test vector. Using their method, the test data compression ratio reached 71.73%, but it did not propose an effective optimization strategy for test data transmission after compression, thus failing to improve the test application time. Kim et al. [28] proposed a test data compressed coding method based on the scan chain reordering by reordering the scan chains according to the compatibility of test data. The method could achieve high test data compression ratio and test power optimization under small area overhead, but it neither proposed an optimization strategy for the transmission method of test data nor achieved good results in reducing the test application time of ATE. Vohra et al. [16] combined the 9C compressed coding method and the CCPRL compressed coding method to propose an optimal selective count compatible run length compressed coding method for test data. This method divided the test data into segments and compressed and encoded the data within each segment and among segments. In their proposed method, although the compression rate of the test data was 80%, it did not propose an improvement strategy for the data transmission after coding, and the improvement effect on the test application time was not observed. Zhan et al. [29] introduced an ERLC-based test data compressed coding scheme by filling the don't-care bits 0 or 1 according to the distribution of the don't-care bits of the test data. The scheme achieved high test data compression ratio under the low area overhead, but it did not propose an effective optimization strategy for the transmission method of test stimuli after coding. Meanwhile, Manjurathi et al. [30] introduced a test data compression algorithm based on test vector reordering, bit stuffing, and test vector coding. The method could improve the compression ratio of test data and reduce test power consumption, but it did not propose a strategy for improving test data transmission after coding and compression. Moreover, the optimization effect of this method on test application time had certain limitations.

Plenty of previous studies were conducted on the test vector compressed coding and test data transmission based on different methods. The existing research methods have generated good compression effect on the data of test stimuli or test response, but the transmission mode of the test data after compressed coding limits further improvements on the test speed. The scan-in of the test stimuli after compressed coding requires many clock cycles, and the parallel transmission of the test stimuli must occupy multiple chip input ports. Therefore, due to the lack of related research, we introduce a novel method of test stimulus coding and test data transmission based on scan chain segmentation and test vector coding technology. First, the test stimuli and scan chains are evenly segmented according to the lengths of the scan chain and the test stimulus. The data width of each segment is fixed at eight bits. Second, the eight-bit data of each segment are coded into the five-bit data according to the compatibility between the segmented data of the test stimulus. In one clock cycle of the ATE, the coded data are entered in parallel using five input ports of the chip; this is done to transmit the decoded eight-bit data to the flip-flops (FFs) corresponding to the scan chain. This approach reduces the input time of test stimuli and the number of ports needed for parallel input. Finally, in order to verify the above method, experiments are conducted by using the ISCAS'89 benchmark circuit.

3 METHODOLOGY

3.1 Test Application Time

A test vector mainly consists of the test stimuli and test responses. The longer the scan chain is, the longer the test stimuli and test responses are, and the bigger the number of clock cycles required by ATE to complete the testing of a test vector. Fig. 1 presents the test process of a scan chain at the length of N. The data lengths of the test stimuli and test responses are also presented as N bits.

The test clock period of ATE is assumed to be 1 s. The number of clock cycles needed to complete the scan testing of a test vector is given Eq. (1)

$$T_1 = (l_{N1} + 1 + l_{N2}), \tag{1}$$

where T_1 represents the time required for the test of a scan chain, in seconds; l_{N1} refers to the number of clock cycles required for the test stimuli scan-in; and l_{N2} refers to the number of clock cycles required for the test response scan-out.

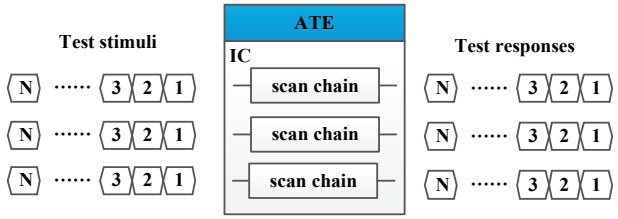

Figure 1 Test of the scan chain at the length of N

If there are many test vectors, the total number of clock cycles of the scan testing without using the pipelining technology is shown in Eq. (2)

$$T_2 = (l_{N1} + 1 + l_{N2}) \times p, \tag{2}$$

where T_2 represents the scan testing time without using the pipelining technology and its unit is the same as T_1, p represents the number of test vectors, l_{N1} epresents the number of clock cycles needed for the scan-in of test stimuli, and l_{N2} represents the number of clock cycles needed for test response output.

The total number of clock cycles of the scan testing using the pipelining technology is shown in Eq. (3)

$$T_3 = (l_{N1} + 1) \times p + l_{N2}, \tag{3}$$

where T_3 represents the scan testing time with using pipelining technology, l_{N2} represents the number of clock cycles needed for the last test response output, p represents the number of test vectors, and l_{N1} represents the number of clock cycles needed for the scan-in of test stimuli.

The scan-in of the test stimuli can cause the state turnover of the FFs and result in the change of dynamic test power. The power consumption for the scan-in of the test stimuli is expressed by Eq. (4)

$$W_{in} = \sum_{i=1}^{p} \left[\sum_{j=1}^{N-1} \left(t_{i,j} \oplus t_{i,j+1} \right) \times j \right], \tag{4}$$

where W_{in} represents the scan-in power, p represents the number of test vectors, N represents the length of the scan chain, $\oplus$ represents the XOR logic operation, and $t_{i,j}$ represents the j^{th} bit of the i^{th} test vector. If $t_{i,j} \oplus t_{i,j+1} = 1$, the values of the j^{th} bit and the $(j+1)^{\text{th}}$ bit of test stimulus are different. The state of the FFs in the scan-in needs turn over, which increases the dynamic power consumption.

The scan-out of the test responses causes the state turnover of the FFs and result in the change of the dynamic test power. The power consumption for the scan-out of the test responses is expressed by Eq. (5)

$$W_{out} = \sum_{i=1}^{p} \left[\sum_{j=1}^{N-1} \left(r_{i,j} \oplus r_{i,j+1} \right) \times \left(N - j \right) \right], \tag{5}$$

where W_{out} represents the scan-out power, and $r_{i,j}$ represents the j^{th} bit of the i^{th} test stimulus. If $r_{i,j} \oplus r_{i,j+1} = 1$, the values of the j^{th} bit and the $(j+1)^{\text{th}}$ bit of the test stimulus are different. During the scan-out, the state of the FFs needs turn over, which increases dynamic power consumption.

3.2 Test Stimulus Segmentation

The test stimuli and test responses in the traditional scan chain test methods are transmitted in a serial manner. A scan chain containing n scan FFs is shown in Fig. 2. In the figure, SE is the scan-enable port. In this case, the test stimuli are serially shifted into the scan FFs of the IC through the SI port, and the test responses are shifted out through the SO port. The above test stimuli or test responses are completed in a serial manner. However, considerable test application time is needed to complete a scan chain test, thus resulting in low transmission efficiency.

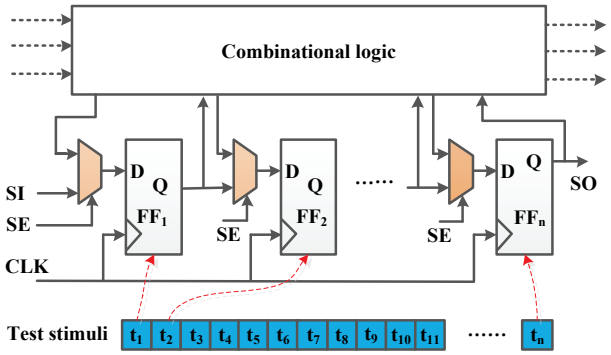

Figure 2 Scan chain containing n scan FFs

In Fig. 2, the length of the test stimuli is the same as that of the scan chain. All test stimuli are transmitted to the location corresponding to the scan chain. In the existing scan testing methods, the test stimuli adopt the serial transmission mode. Test stimuli at the length of n bits require the n clock cycles of the ATE to be transmitted to the scan chain. Thus, data transmission efficiency is low. This mode of data transmission takes up almost 50% of the test time for the scan-in of the test stimuli to complete the test of a test vector, seriously affecting test speed.

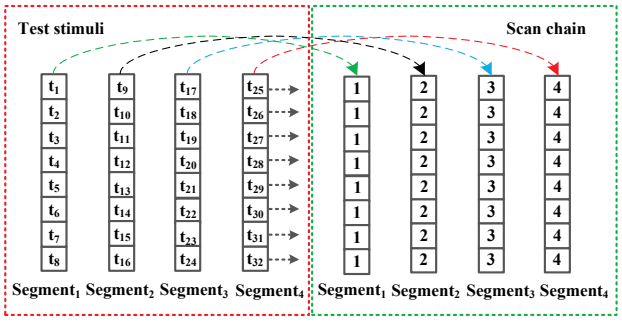

Figure 3 Segmentation of the Test Stimuli with the Width of 8 bits

The segmentation of test stimuli achieves the parallel transmission of test data, thus improving the transmission efficiency of test data. The scan chain and test stimuli are segmented into eight bits according to the data width. In one ATE clock cycle, the eight-bit test stimuli are transmitted in parallel to the corresponding scan FFs. The segmentation of the test stimuli and the scan chain is shown in Fig. 3 when the data widths of the scan chain and the test stimuli are 32 bits. The test stimuli after segmentation are transmitted in parallel to the FFs of the scan chain.

In Fig. 3, the part in the left red dotted line frame represents the test stimuli, and the part in the right green dotted line frame represents the FFs of the scan chains. The test stimuli and scan FFs are segmented equally, and each segment is eight-bit wide. The test stimuli of each segment can be directly transmitted to the corresponding scan FFs in parallel, and 32 bits of the test stimuli can be used as input through four ATE clock cycles, thus indicating high transmission efficiency. However, this direct parallel transmission mode needs to configure eight-chip input ports for each scan chain. If there are several scan chains, the chip ports may be insufficient. The width of each segment is eight bits. In most cases, the data width of the test stimuli and the length of the scan chain are not the integral multiples of 8. Thus, the segment number of the test stimuli is calculated by Eq. (6)

$$k = \text{ceil}\left(\frac{l_{N1}}{8}\right), \tag{6}$$

where k represents the segment number of test stimuli, ceil( ) represents the smallest integer returning to greater than or equal to the specified expression, and l_{N1} represents the width of the test stimuli.

3.3 Test Stimuli Coding Method

The test vectors consist of the test stimuli and test responses. The values of some data in the test stimuli (0 or 1) do not affect the fault coverage of the scan testing. These data is called x bits, and after segmentation of the test stimuli, the data width of each segment is eight bits. The data compatibility of this study is defined as follows: the eight-bit data of the current segment are compared with the eight-bit data of the previous segment in position; if the data in the corresponding two bits are the same or if the data of the current segment are x bits, these data are considered compatible. As shown in Fig. 4, if the length of the test stimuli is 32 bits, the test data that are not compatible between adjacent segments are expressed in blue.

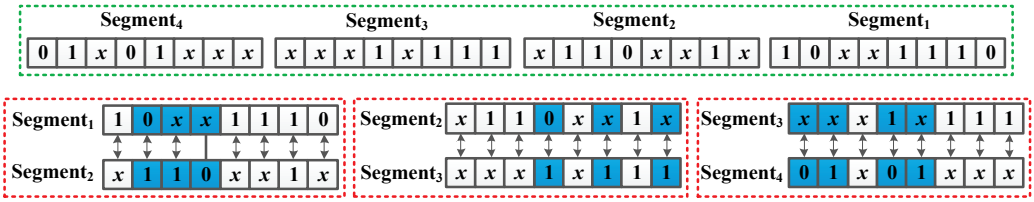

Figure 4 Data Compatibility between Adjacent Segments

The test stimuli can be coded using x bits and data compatibility. The test stimuli data of the next segment can be decoded using those of the previous segment. The x bits and data compatibility of the test stimuli are used after the statistics of the test vectors. All x bits are set to 1 in the test stimuli, which does not affect the fault coverage of the IC test. The data of the next segment can be obtained by inverting part of the test stimuli data of the previous segment.

Next, the test stimuli are segmented with eight-bit wide data. The data width of each segment is eight bits, which can be expressed by five bits after encoding. Data are assumed to be expressed by $t_8t_7t_6t_5t_4t_3t_2t_1$ before coding and $e_5e_4e_3e_2e_1$ after encoding. After the x bit in $t_8t_7t_6t_5t_4t_3t_2t_1$ is set to 1, the data are expressed by $s_8s_7s_6s_5s_4s_3s_2s_1$. They are compared with the test data of the previous decoded segment. The number of data width that needs to be reversed in the current segment is expressed by p, $0 \le p \le 8$.

When p = 1, only one among eight bits of data $t_8t_7t_6t_5t_4t_3t_2t_1$ in the current segment needs to be reversed compared with the data of the previous segment. The number of states is eight. When $e_5e_4e_3$ is encoded to 3'b001 the reversed data take place at the four least significant bits of $t_8t_7t_6t_5t_4t_3t_2t_1$. The position of $t_4t_3t_2t_1$ is binary-encoded using the binary form of e_2e_1. When $e_5e_4e_3$ is encoded by 3'b010, the reversed data take place at the four most significant bits of $t_8t_7t_6t_5t_4t_3t_2t_1$. The position of $t_8t_7t_6t_5$ is binary-encoded using the binary form of e_2e_1. The detailed encoding method is shown in Tab. 1.

Table 1 Encoding truth table when p = 1

$t_8t_7t_6t_5t_4t_3t_2t_1$	$e_5e_4e_3$	e_2e_1	*clock cycle*
t_1	001	00	1
t_2	001	01	1
t_3	001	10	1
t_4	001	11	1
t_5	010	00	1
t_6	010	01	1
t_7	010	10	1
t_8	010	11	1

In Tab. 1, $e_5e_4e_3$ represents the high three bits of the encoding result, e_2e_1 represents the low two bits of the encoding result, and the clock cycle represents the number of clock cycles required for decoding circuit. If p = 0, no bit of data in the current segment needs to be reversed compared with the previous decoded data; here, $e_5e_4e_3$ is encoded to 3'b000. The value of e_2e_1 in this case can be ignored. The decoded data of the previous segment can be directly used similar to those of the current segment.

When p > 1, several bits of data among eight in the current segment need to be reversed compared with the data of the previous segment. To reduce the complexity of

the decoding circuit, two ATE clock cycles are used to complete the transmission of the segmented data; e_5 is set as the flag bit with the fixed coding of 1; $e_4e_3e_2e_1$ is encoded to $s_4s_3s_2s_1$ in the first cycle of ATE; and $s_4s_3s_2s_1$ is encoded to $s_8s_7s_6s_5$ in the second cycle of ATE. The first segment of the test stimuli is encoded according to the situation when $p > 1$. The detailed encoding method is shown in Tab. 2.

Table 2 Encoding method of the test stimuli

p	$e_5e_4e_3e_2e_1$	*clock cycle*
$p = 0$	001e_2e_1	1
$p = 1$	000*xx*	1
	010e_2e_1	
$p > 1$	1$s_4s_3s_2s_1$	2
	1$s_8s_7s_6s_5$	

The encoding algorithm of this study is described as follows:

(1) Set all x bits of the test stimuli to 1 and fix the segmented data to eight bits to calculate the number of segments of the current test stimuli expressed by k.
(2) Encode the data of the first segment according to Step 6.
(3) Calculate the value of p according to the data of the next segment; if $p = 0$, skip to Step (4); if $p = 1$, skip to Step (5); and if $p > 0$, skip to Step (6).
(4) Update $s_8s_7s_6s_5s_4s_3s_2s_1$ and encode $e_5e_4e_3$ to 3'*b*000.
(5) Update $s_8s_7s_6s_5s_4s_3s_2s_1$, encode $e_5e_4e_3$ to 3'*b*001, and encode e_2e_1 according to Tab. 1.
(6) Update $s_8s_7s_6s_5s_4s_3s_2s_1$, encode e_5 to 1, encode $e_4e_3e_2e_1$ to $s_4s_3s_2s_1$ in the first clock cycle, and encode $e_4e_3e_2e_1$ to $s_8s_7s_6s_5$ in the second clock cycle.
(7) Return to Step (3) to repeat the operation; after completing the encoding of the last segment, encode $e_5e_4e_3$ to 3'*b*001.

A detailed test stimuli coding process is shown in Tab. 3 to illustrate the coding method clearly. The total number of segments is 16, and the data of each segment are eight bits.

Table 3 Test stimuli coding with 16 segments

Segment No.	$t_8t_7t_6t_5t_4t_3t_2t_1$	$s_8s_7s_6s_5s_4s_3s_2s_1$	p	$e_5e_4e_3e_2e_1$	*clock cycle*
1	11*x*00*xx*1	11100111	$p > 1$	10111	2
				11110	
2	11*xxxx*01	11111101	$p = 3$	11101	2
				11111	
3	*x*1*xx*11*xx*	11111111	$p = 1$	00101	1
4	*xxx*1*x*1*xx*	11111111	$p = 0$	000*xx*	1
5	*x*0*xxx*1*x*1	10111111	$p = 1$	01010	1
6	*x*0*x*0*xxx*1	10101111	$p = 1$	01000	1
7	1*x*0*xx*1*xx*	11011111	$p = 3$	11111	2
				11101	
8	*xx*0*xx*0*xx*	11011011	$p = 1$	00110	1
9	*xxx*11*xxx*	11111111	$p = 2$	11111	2
				11111	
10	*xxxxx*0*x*0	11111010	$p = 2$	11010	2
				11111	
11	011*xxxxx*	01111111	$p = 3$	11111	2
				10111	
12	*xxx*11*xx*1	11111111	$p = 1$	01011	1
13	*x*1*xx*1*xxx*	11111111	$p = 0$	000*xx*	1
14	*xx*111*xxx*	11111111	$p = 0$	000*xx*	1
15	0*xxxx*0*x*0	01111010	$p = 3$	11010	2
				10111	
16	00*x*11001	00111001	$p = 3$	11001	2
				10011	

The test stimuli in Tab. 3 have a total of 16 segments at eight-bit wide. The data of the first segment are 8'*b*11*x*00*xx*1. All x bits in the segment are set to 1. Thus, the data become 8'*b*11100111. According to the encoding method when $p > 1$, it needs two ATE clock cycles. In the first cycle, e_5 is encoded to 1, and $e_4e_3e_2e_1$ is encoded to the four least significant bits of 8'*b*11100111. In the second cycle, e_5 is encoded to 1, and $e_4e_3e_2e_1$ is encoded to the four most significant bits of 8'*b*11100111. The data of the second segment are 8'*b*11*xxxx*01. All x bits in the segment are set to 1. Thus, the data become 8'*b*11111101. A comparison of the data of the second segment with those of the first segment bit by bit shows that the three bits of data need to be reversed. Thus, $p = 3$, which belongs to the encoding situation of $p > 0$. Thus, the encoding method in Tab. 2 is adopted. The data of the third segment are 8'*bx*1*xx*11*xx*. All x bits in the segment are set to 1. Thus, the value of $s_8s_7s_6s_5s_4s_3s_2s_1$ becomes 8'*b*11111111. A comparison of the data with those of the previous segment shows that only s_2 requires a reversal in order to complete the test data transmission, which belongs to the encoding situation of $p = 1$. According to the encoding methods in Tabs. 2 and 1, $e_5e_4e_3$ is encoded to 3'*b*001 and e_2e_1 is encoded to 2'*b*01, respectively. The data of the fourth segment are 8'*bxxx*1x1*xx*. All x bits in the segment are set to 1. Thus, the value of $s_8s_7s_6s_5s_4s_3s_2s_1$ becomes 8'*b*11111111. The data of the current segment are consistent with those of the previous segment. Thus, the bits do not need to be reversed. This result belongs to the situation of $p = 0$. According to the encoding method in Tab. 2, $e_5e_4e_3e_2e_1$ is encoded to 5'*b*000*xx*. The test stimuli in Table 3 have a total of 16 segments. The data of the

subsequent segments are encoded by the cases above. The data of each segment are eight-bit wide, resulting in a total of 128 bits of test data. Thus, a total of 128 ATE clock cycles are required according to the serial transmission mode. However, only 24 ATE clock cycles are needed in the encoding method used in this study, thus saving 81.3% transmission time.

3.4 Structure of the Decoding Circuit

The data bits of the test stimuli before coding are consistent with the number of scan FFs of scan chains. During the scan testing, all test stimuli must be transmitted to the scan FFs. The test stimuli and scan chains are segmented according to the data width of eight bits. The number of segments is k, and the data of each segment is encoded to the encoded data of five bits. ATE transmits the data of five bits into the CUT. The test stimuli must first be decoded before the coded data are transmitted to the scan FFs. The decoding circuit diagram of the test stimuli is shown in Fig. 5.

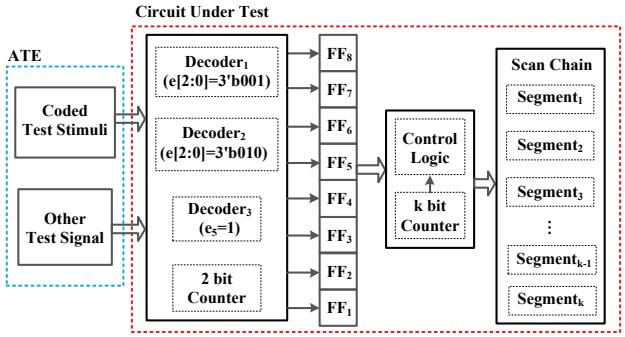

Figure 5 Decoding circuit diagram of the test stimuli

The test stimuli are divided into several segments with eight bits each. The eight-bit data of each segment are encoded according to the method in Section 3.3. The data are stored in ATE after coding. Under the control of the scan testing signal, the coded test stimuli are transmitted in parallel through the five input ports to the decoding circuit inside the chip. The decoding circuit decodes according to the characteristics of the coded data, stores the decoded data to the eight-bit FFs, and then transmits them to the corresponding scan FFs to complete the rapid input of the test stimuli.

In Fig. 5, the coded test stimuli are stored in the ATE, which represents the coded test stimuli. Other test signals represent other control signals of the scan testing. FF_1 to FF_8 are the FFs of the decoding circuit, which are used to store the decoded segmented data. In case $e_5e_4e_3$ is encoded to 3'b001, the data must be reversed at the four least significant bits of $t_8t_7t_6t_5t_4t_3t_2t_1$. The position of $t_4t_3t_2t_1$ is binary-encoded in the binary form of e_2e_1. $Decoder_1$ is used to decode e[4:0] data in this case, and the decoded data are stored in the FFs. In case the $e_5e_4e_3$ is encoded to 3'b010, the data must be reversed at the four most significant bits of $t_8t_7t_6t_5t_4t_3t_2t_1$. The position of $t_8t_7t_6t_5$ is binary-encoded in the binary form of e_2e_1. $Decoder_2$ is used to decode the e[4:0] data in this case. The decoded data are stored in the FFs; $e_5 = 1$ represents more than two bits of data that need to be reversed in the current segment compared with those of the previous segment. In this case, the decoding circuit needs two ATE clock cycles. In the first cycle, $e_4e_3e_2e_1$ are directly assigned to the four least significant bits of the FFs. In the second cycle, $e_4e_3e_2e_1$ are directly assigned to the four most significant bits of the FFs. The decoding process requires a two-bit counter to control the assignment objects of $e_4e_3e_2e_1$ that are considered four least significant bits or the four most significant bits of the FFs. The segmentation method of the FFs of the scan chains is consistent with that of the test stimuli. $Segment_1$ to $Segment_k$ represent the segments of the scan chains. The data of each segment are eight bits wide. The decoded test stimuli are stored in eight FFs of the decoding circuit. The eight-bit test stimuli must be accurately transmitted to the corresponding FFs of the scan chains after decoding. The k-bit counter and control logic module are used to control the accurate transmission of the decoded test stimuli to the corresponding scan chain segments.

At least one ATE clock cycle is needed to complete the data decoding of a segment. Two ATE clock cycles are needed at the most to complete the data decoding of all segments. The average ATE clock cycle for data decoding is represented by q, $1 < q \leq 2$. Therefore, the calculation of the number of clock cycles required for the parallel input of a test stimulus according to the above coding and decoding methods is shown in Eq. (7)

$$l_C = k \times q \quad \left(1 < q \leq 2, \frac{l_{N1}}{8} < l_C \leq \frac{l_{N1}}{4}\right), \tag{7}$$

The calculation for the test of a test vector is shown in Eq. (8)

$$T_C = (l_C + 1 + l_{N2}), \tag{8}$$

where l_C represents the number of clock cycles required to complete the parallel input of a test stimulus, k represents the number of segments of the test stimuli, and q is the coding coefficient, $1 < q \leq 2$. The value range of l_C is shown as Eq. (7). In addition, l_{N2} represents the time of serial output of test responses. Here, T_C represents the time to complete the testing of a test vector after using the encoding and decoding methods proposed in the study.

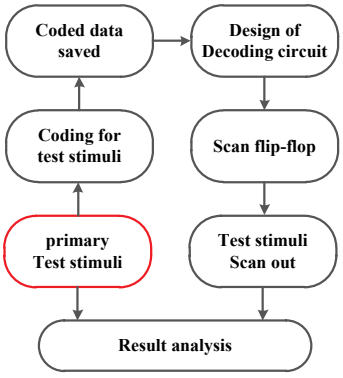

Figure 6 Verification process

4 RESULT ANALYSIS AND DISCUSSION

An encoding and decoding method of the test stimuli is proposed in this study. In one or two clock cycles of the

ATE, the eight-bit test stimuli can be rapidly scanned in using five input ports of the chip. To validate the accuracy of the proposed encoding and decoding method, seven circuits of the largest size in the ISCAS'89 benchmark circuit are used. The verification process is shown in Fig. 6.

In Fig. 6, the red box represents the un-encoded test stimuli. The test stimuli are encoded using VC++6.0. The decoding circuit is implemented using the Verilog hardware description language. The logic circuit simulation, logical synthesis, scan chain insertion, and test vector generation are realized by using the electronic design automation tools of Synopsys Company. The concrete validation steps are described below.

(1) Implement logic synthesis, scan chain insertion, and test vector generation on the benchmark circuit; extract the test stimuli from the test vectors and store them.
(2) Encode the test stimuli by using the encoding method introduced in Section 3.3 to encode the eight-bit segmented data into five bits data. Store the encoded data.
(3) Add the decoding circuit in the benchmark circuit by using the decoding method introduced in Section 3.4.
(4) Assign the decoded test stimuli to eight scan FFs of the decoder according to the segmentation method of test stimuli. Then, transmit the state values of eight FFs to the corresponding segments of the scan chains.
(5) Serially output the data saved in the scan FFs. Compare the saved data with the test stimuli in Step 1.

After using the coding method described above, the output power of the test responses is not changed according to Eq. (5). The input power of test stimuli can be expressed by Eq. (9)

$$W_{in}' = \sum_{j=1}^{p}\left[\sum_{i=1}^{k}\left(s_{i,j} \oplus s_{i+1,j}\right)\right], \tag{9}$$

where W_{in}' represents the input power of test stimuli after using the above coding method, p represents the number of test vectors, k represents the number of groups of test stimuli, $s_{i,j} = (s_8s_7s_6s_5s_4s_3s_2s_1)_{i,j}$ represents the segmented data after the x bit is set to 1; the subscript j represents the j^{th} test vector, and the subscript i represents the i^{th} segment of the current test stimuli. In addition, $s_{i,j} \oplus s_{i,j+1}$ represents the xor calculation of the data of the current segment and the next segment. The test stimuli are directly sent to the predetermined scan FFs in parallel using the above coding method, thus saving the dynamic power consumption generated by the serial shift of the test vectors. The transmission time of the test stimuli is improved after using the above methods. The number of the corresponding ATE clock cycles and dynamic power consumption are also optimized. Therefore, using the above coding and decoding methods will not increase the test power consumption.

This study proposes the test stimuli coding and decoding methods to reduce the transmission time of test stimuli. The test stimuli are evenly divided into multiple segments. The data in each segment are eight bits wide. The eight-bit data are encoded to the five-bit data. The corresponding decoding circuit is designed to reduce the input time of the test stimuli and compress the test stimuli data. The lengths of the data before and after the coding of the test stimuli are shown in Tab. 4 based on the test stimuli data of the seven benchmark circuits at a large scale.

Table 4 Data compression rate after encoding the test stimuli

Circuits	Inputs	Outputs	l_{N1} (s)	k	Coded Data	CR%
s5378	35	49	179	23	115	35.8%
s9234	19	22	228	29	145	36.4%
s13207	31	121	669	84	420	37.2%
s15850	14	87	597	75	375	37.2%
s35932	35	320	1728	216	1080	37.5%
s38417	28	106	1636	205	1025	37.3%
s38584	12	278	1452	182	910	37.3%
Avg	-	-	-	-	-	37%

In the table, "Circuits" represents the names of the benchmark circuits, "Inputs" represents the number of input ports of the benchmark circuits, "Outputs" represents the number of output ports of the benchmark circuits, l_{N1} represents the length of the test stimuli data in a test vector, k represents the number of segments of the test stimuli, "Coded Data" represents the data length after the encoding of test stimuli, and "CR" represents the data compression rate. "CR" can be calculated by using the equation

$$CR = \frac{(T_D - T_C)}{T_D} \times 100\%, \tag{10}$$

where T_D represents the length of the test stimulus data before encoding, and T_C represents the length of the test stimulus data after encoding.

From the data in Tab. 4, the test stimuli of Circuit s35932 are 1,728 bits long, which is simply the integral multiples of 8. Thus, the data compression rate after encoding of the test stimuli is 37.5%. The lengths of the test stimuli of the remaining circuits are not the integral multiples of 8. Thus, compression rate is slightly lower than 37.5%. Therefore, the test vector coding method proposed in the paper has a certain data compression effect.

A novel method of test stimuli coding and decoding is proposed to reduce the number of clock cycles for the scan-in of test stimuli and reduce the transmission time. The ISCAS'89 benchmark circuit is used to carry out the experiments. The experimental results of the above coding method are compared with those of the serial input method. The test vectors are generated by using the automatic test pattern generation (ATPG) tool of Synopsys Company. The test stimuli data of 20 test vectors are extracted, encoded, and decoded by using the test stimuli coding and

decoding method proposed in the paper. The corresponding experimental data are shown in Tab. 5.

In the table, "Circuits" represents the name of the benchmark test circuit, l_{N1} represents the time required for the serial input of test stimuli in a test vector, k represents the number of segments of the test stimuli, and "Coded Data" represents the data length of the test stimuli after encoding. The test stimuli can be quickly transmitted to the scan chains within the IC using the method proposed in this study. The number of clock cycles required is shown in Eq. (7). To facilitate analysis, an ATE clock cycle is assumed to be 1 s. In addition, l_C represents the time required for completing the transmission of a test stimulus in seconds, and red_1 represents the percentage of the time reduction for test stimuli transmission, which can be calculated by using the Eq. (11)

$$red_1 = \frac{(l_{N1} - l_C)}{l_{N1}} \times 100\%, \tag{11}$$

Table 5 Time optimization of the test stimuli

Circuits	l_{N1} (s)	Patterns	k	Coded Data (bits)	l_C (s)	red_1 %
s5378	179	20	23	115	35.6	80%
s9234	228	20	29	145	46.4	79.6%
s13207	669	20	84	420	126.8	81%
s15850	597	20	75	375	111.6	81.3%
s35932	1728	20	216	1080	259.2	85%
s38417	1636	20	205	1025	287.4	82.4%
s38584	1452	20	182	910	251.2	82.7%
Avg	-	-	-	-	-	81.7%

The data in Tab. 5 show that, in the experimental data of randomly extracted 20 test stimuli, the proposed method has a good optimization effect on the transmission time of the test stimuli. This method can save an average transmission time of 81.7% for the test stimuli. Among them, s35932 has the best optimization effect on the transmission time of test stimuli, and s9234 has the worst optimization effect. These can be attributed to the fact that the proportion of x bits of test stimuli on s35932 is higher than that on s9234.

To further demonstrate the feasibility of the proposed method, the test vectors generated by seven ISCAS'89 benchmark circuits with the largest number of scan FFs by Mintest ATPG tool are gathered. Fifty of the test stimuli are selected for the experiments. The experimental data are shown in Tab. 6.

Table 6 Experimental data by using the mintest ATPG tool

Circuits	l_{N1} (s)	T_1 (s)	Patterns	k	Coded Data (bits)	T_C (s)	red_2 %
s5378	179	359	50	23	115	213.6	40.5%
s9234	228	457	50	29	145	276.4	39.5%
s13207	669	1339	50	84	420	783.8	41.5%
s15850	597	1195	50	75	375	702.6	41.2%
s35932	1728	3457	50	216	1080	1958.2	43.4%
s38417	1636	3273	50	205	1025	1904.4	41.8%
s38584	1452	2905	50	182	910	1724.2	40.6%
Avg	-	-	-	-	-	-	41.2%

In the table, "Circuits" represents the name of the benchmark test circuit, l_{N1} represents the time required for the serial input of test stimuli in a test vector, k represents the number of segments of the test stimuli, T_1 represents the input time of the test stimuli without using the method proposed in the paper, k represents the number of segments of the test stimuli, and "Coded Data" represents the data length of the test stimuli after encoding. The time required for completing the test of a test vector using the proposed encoding and decoding method is expressed by T_C and is calculated by Eq. (8). red_2 represents the percentage of the time reduction for test stimuli transmission, which can be calculated by Eq. (11).

Fifty of the test vectors generated by the Mintest ATPG tool are randomly selected for the experiments. According to the data in Tab. 6, the proposed method can save an average of 41.2% of the testing time for the test vectors compared with the results generated by using the method of serial input of test stimuli.

5 CONCLUSIONS

A novel method of test stimuli coding and decoding was proposed in this study to increase the speed of scan testing and reduce the number of input ports required for the parallel input of the test stimuli. First, the segments of the test stimuli were fixed at eight bits wide. The test stimuli and scan chains were evenly segmented. Second, a novel method of test stimuli coding and decoding was proposed. This method can encode the eight-bit data of each segment into the five-bit data, thereby realizing the rapid input of the test stimuli. This method was verified by using the ISCAS'89 benchmark circuits. The following conclusions are drawn.

(1) This study proposes a method for the fast scan-in of the test stimuli. The experimental results verify the effectiveness of the proposed method. In this study, the test stimuli are segmented into eight bits wide. The parallel transmission of the eight-bit test stimuli can be achieved by using the five input ports of the IC through the above coding and decoding method, thus realizing the rapid input of the test stimuli.

(2) The proposed test stimuli coding and decoding method can effectively reduce the number of input ports required for parallel input of test stimuli. In the case that IC contains several scan chains but has a small number of input ports, the parallel input of the segmented test stimuli requires multiple input ports. However, only five input ports are used to realize the parallel transmission of the eight-bit test stimuli when the test stimuli coding and decoding method proposed in this study is used.

This study proposes a novel method of test stimuli coding and decoding based on the traditional scan testing technology and test stimuli segmentation. This method can encode the eight-bit data of each segment to five-bit data, thus reducing the requirement of the parallel input of the test stimuli for the number of input ports of IC. The data width of each segment of the test stimuli is fixed at eight bits. The ISCAS'89 benchmark circuit is used for verification. Thus, some limitations are observed in the segment width and the experimental object.

By using the test stimuli segmentation and coding methods, this study reveals that the occurring sequence of the segmented data has a significant influence on the coding result. This method can be scheduled by the reasonable intelligent optimization algorithm, and on such basis, future studies may discuss the coding scheduling problem of the segmented data of the test stimuli.

Acknowledgements

This study was supported by Fujian provincial leading project under Grant No. 2017H0029 and Electronic Information and Control of Fujian University Engineering Research Center under Grant No. EIC1707.

6 REFERENCES

[1] Sismanoglou, P. & Nikolos, D. (2013). Input test data compression based on the reuse of parts of dictionary entries: static and dynamic approaches. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 32*(11), 1762-1775. https://doi.org/10.1109/TCAD.2013.2270433

[2] Chen, C., Lin, B., & Michelle, Z. (2018). Verification method for area optimization of mixed-polarity Reed-Muller logic circuits. *Journal of Engineering Science and Technology Review, 11*(1), 28-34. https://doi.org/10.25103/jestr.111.04

[3] Yuan, H., Ju, Z., Sun, X., Guo, K., & Wang, X. (2016). Test data compression for system-on-chip using flexible runs-aware PRL coding. *Journal of Electronic Testing, 32*(5), 1-9. https://doi.org/10.1007/s10836-016-5595-z

[4] Zhan, W., Liang, H., Cheng, Y., Wu, H., & Zhu, S. (2016). Test Data Compression Coding-based Scheme Storing Integers Represented for Irrational Numbers. *Journal of Computer-Aided Design & Computer Graphics, 28*(9), 1605-1612.

[5] Yuan, H., Guo, K., Sun, X., & Ju, Z. (2016). A power efficient test data compression method for SoC using alternating statistical run-length coding. *Journal of Electronic Testing, 32*(1), 59-68. https://doi.org/10.1007/s10836-016-5562-8

[6] Cai, S., Zhou, Y., Liu, P., Yu, F., & Wang, W. (2017). A novel test data compression approach based on bit reversion. *IEICE Electronics Express,14*(13), 1-11. https://doi.org/10.1587/elex.14.20170502

[7] Chen, C., Wei, R., Luo, H., & Chen, Y. (2018). Improved test stimulus scan-in method for integrated circuits. *Dyna, 93*(4), 391-397. https://doi.org/10.6036/8788

[8] Lee, L. J., Tseng, W. D., & Liu, Y. Y. (2015). A hybrid method for segmented test data compression. *Journal of the Chinese Institute of Engineers, 38*(2), 169-180. https://doi.org/10.1080/02533839.2014.955971

[9] Wu, T., Liu, H., & Zhang, B. (2014). A novel test data compression scheme for SoCs based on block merging and compatibility. *IEICE Transactions on Fundamentals of Electronics Communications & Computer Sciences, E97.A*(7), 1452-1460. https://doi.org/10.1587/transfun.E97.A.1452

[10] Sismanoglou, P. & Nikolos, D. (2013). Input test data compression based on the reuse of parts of dictionary entries: static and dynamic approaches. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 32*(11), 1762-1775. https://doi.org/10.1109/TCAD.2013.2270433

[11] Yuan, H., Mei, J., Song, H., & Guo, K. (2014). Test data compression for system-on-a-chip using count compatible pattern run-length coding. *Journal of Electronic Testing, 30*(2), 237-242. https://doi.org/10.1007/s10836-014-5441-0

[12] Wu, D., Liu, Y., Zhu, H., Wang, D., & Hao, C. (2013). A novel pattern run-length coding method for test data compression. *IEICE Transactions on Electronics, 96*(9), 1201-1204. https://doi.org/10.1587/transele.E96.C.1201

[13] Mehta, U. S., Dasgupta, K. S., & Devashrayee, N. M. (2010). Modified selective huffman coding for optimization of test data compression, test application time and area overhead. *Journal of Electronic Testing, 26*(6), 679-688. https://doi.org/10.1007/s10836-010-5183-6

[14] Sarkar, S. K., Jingjing, C., & Wenge, G. (2013). Multiple testing in a two-stage adaptive design with combination tests controlling FDR. *Journal of the American Statistical Association, 108*(504), 1385-1401. https://doi.org/10.1080/01621459.2013.835662

[15] You, Z. Q. & Hu, N. (2015). A high test quality test compression method based on fdr codes. *Journal of Hunan University, 42*(2), 109-113.

[16] Vohra, H. & Singh, A. (2016). Optimal selective count compatible runlength encoding for SoC test data compression. *Journal of Electronic Testing, 32*(6), 1-13. https://doi.org/10.1007/s10836-016-5617-x

[17] Wu, T., Liu, H., & Zhang, B. (2014). A novel test data compression scheme for SoCs based on block merging and compatibility. *IEICE Transactions on Fundamentals of Electronics Communications & Computer Sciences, E97.A*(7), 1452-1460. https://doi.org/10.1587/transfun.E97.A.1452

[18] Chandra, A. & Chakrabarty, K. (2003). Test data compression and test resource partitioning for system-on-a-chip using frequency-directed run-length (FDR) codes. *IEEE transactions on computers, 52*(8), 1076-1088. https://doi.org/10.1109/TC.2003.1223641

[19] Kuang, J., Zhou, Y., Cai, S., & Pi, X. (2015). Improved EFDR code method for test data compression. *Journal of Electronic Measurement & Instrumentation, 29*(10), 1464-1471.

[20] Sivanandam, L., Oorkavalan, U. M., & Periyasamy, S. (2017). Test data compression for digital circuits using tetrad state skip scheme. *Design Automation for Embedded Systems, 21*(8), 1-15. https://doi.org/10.1007/s10617-017-9196-6

[21] Papameletis, C., Keller, B., Chickermane, V., & Hamdioui, S. (2015). A dft architecture and tool flow for 3-d sics with test data compression, embedded cores, and multiple towers. *IEEE Design & Test, 32*(4), 40-48. https://doi.org/10.1109/MDAT.2015.2424422

[22] Czysz, D., Mrugalski, G., Mukherjee, N., Rajski, J., & Tyszer, J. (2013). On deploying scan chains for data storage in test compression environment. *IEEE Design & Test, 30*(1), 68-76. https://doi.org/10.1109/MDT.2012.2184072
[23] Cheng, Y. F. & Zhan, W. F. (2014). Test data compression method of dual run length alternating coding. *Computer Science, 41*(11), 22-24.
[24] Liu, J., Liang, H. G., Yi, M. X., & Zhao, F. Y. (2012). Test data compression for multiple scan chains with dynamic vector adjustment. *Acta Electronica Sinica, 40*(2), 287-292.
[25] Wu, T., Liu, H., & Zhang, B. (2014). A novel test data compression scheme for SoCs based on block merging and compatibility. *IEICE Transactions on Fundamentals of Electronics Communications & Computer Sciences, E97.A*(7), 1452-1460. https://doi.org/10.1587/transfun.E97.A.1452
[26] Seo, S., Yong, L., & Kang, S. (2016). Tri-state coding using reconfiguration of twisted ring counter for test data compression. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 35*(2), 274-284. https://doi.org/10.1109/TCAD.2015.2413416
[27] Yuan, H., Mei, J., Song, H., & Guo, K. (2014). Test data compression for system-on-a-chip using count compatible pattern run-length coding. *Journal of Electronic Testing, 30*(2), 237-242. https://doi.org/10.1007/s10836-014-5441-0
[28] Kim, D., Ansari, M. A., Jung, J., & Park, S. (2016). Low power scan chain reordering method with limited routing congestion for code-based test data compression. *Journal of Semiconductor Technology Andence, 16*(5), 583-594. https://doi.org/10.5573/JSTS.2016.16.5.582
[29] Zhan, W., & El-Maleh, A. (2012). A new scheme of test data compression based on equal-run-length coding (ERLC). *Integration the Vlsi Journal, 45*(1), 91-98. https://doi.org/10.1016/j.vlsi.2011.05.001
[30] Manjurathi, B. & Rajaguru, R. H. (2016). Analysis of test data compression and power reduction using multiple encoding for opto electronic circuits. *Journal of Optoelectronics & Advanced Materials, 18*(1-2), 112-117.

Contact information:

Haibo LUO, PhD
(Corresponding author)
Electronic Information and Control of Fujian,
University Engineering Research Center, Minjiang University,
200 Xi Yuan Gong Road, University Town, Fuzhou, Fujian Province, China
E-mail: 50209525@qq.com

Chuandong CHEN
Department of Microelectronics, Fuzhou University,
2 Xue Yuan Road, University Town, Fuzhou, Fujian Province, China
E-mail: cdchen@fzu.edu.cn

Yuxiang CHEN
Division of Physical Sciences, The University of Chicago,
5640 South Ellis Avenue, Chicago, IL, United States
E-mail: yuxiangc@uchicago.edu