Journal of Electronic Testing (2018) 34:685–695
https://doi.org/10.1007/s10836-018-5756-3

LFSR Reseeding-Oriented Low-Power Test-Compression Architecture for Scan Designs

Haiying Yuan [1] · Changshi Zhou [1] · Xun Sun [2] · Kai Zhang [1] · Tong Zheng [1] · Chang Liu [1] · Xiuyu Wang [1]

Received: 3 July 2018 / Accepted: 9 October 2018 / Published online: 29 October 2018
© Springer Science+Business Media, LLC, part of Springer Nature 2018

Abstract
Massive test data volume and excessive test power consumption have become two strict challenges for very large scale integrated circuit testing. In BIST architecture, the unspecified bits are randomly filled by LFSR reseeding-based test compression scheme, which produces enormous switching activities during circuit testing, thereby causing high test power consumption for scan design. To solve the above thorny problem, LFSR reseeding-oriented low-power test-compression architecture is developed, and an optimized encoding algorithm is involved in conjunction with any LFSR-reseeding scheme to effectively reduce test storage and power consumption, it includes test cube-based block processing, dividing into hold partition sets and updating hold partition sets. The main contributions is to decrease logic transitions in scan chains and reduce specified bit in test cubes generated via LFSR reseeding. Experimental results demonstrate that the proposed scheme achieves a high test compression efficiency than the existing methods while significantly reduces test power consumption with acceptable area overhead for most Benchmark circuits.

Keywords LFSR reseeding · Low power test · Test data compression · Scan design · Built-in self-test

1 Introduction

With the development of semiconductor technology to ultra-deep submicron and nano-scale, the integration and complexity of very large scale integrated circuit increase rapidly. It faces huge challenges in high test power consumption and growing test data volume [11], more memory occupancy and prolonged test application time results in high test cost, which seriously impacts on the integrated circuits manufacturing. Traditional test scheme using automatic test equipment (ATE) is an inefficient method for advanced system-on-a-chip (SoC) with a large amount of test data volume. The testing of integrated circuits with random test patterns will result in high power consumption in test mode [14]which is hazardous to functionality of integrated chip [10]and may cause instant circuit damage, decreased system reliability and an undesired yield loss [15, 22].

Test set are randomly generated by Automatic Test Pattern Generation (ATPG) tool in Built-In Self-Test (BIST) architecture, which is performed on LFSR (linear-feedback shift-register) reseeding scheme with seed vectors, it significantly reduces test-data storage and transmission bandwidth. For given a deterministic test cube, the corresponding seed vectors can be achieved by solving a set of linear equations (one equation for each specified bit) based on the feedback polynomial of the LFSR circuit. Only 1%–5% specified bits in a test set is applied to solve a seed vector. It was determined in [7] that the probability of unsolvable is less than 10^{-6}, the LFSR size must be larger than S_{max} +20, where S_{max} is the largest number of specified bits in all test cubes. Furthermore, multiple-polynomial LFSRs [5] or variable-length multiple polynomial LFSRs [6] are used to reduce the total bits of seeds. An LFSR with S_{max} size is sufficient to encode all deterministic test cubes for circuits testing [4, 20].

Responsible Editor: N. A. Touba

✉ Haiying Yuan
yhyingcn@gmail.com

[1] Faculty of Information Technology, Beijing University of Technology, Beijing 100124, People's Republic of China

[2] Department of Electronic Engineering, Tsinghua University, Beijing 100084, People's Republic of China

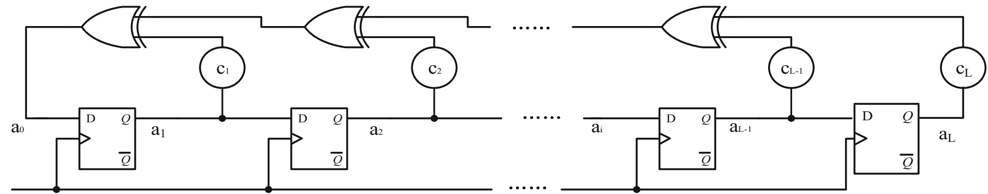

Fig. 1 LFSR architecture-based XOR external-connection

The unspecified bits in test cubes are filled with pseudo-random number though LFSR reseeding, so excessive switching activity occur when they are shifted into a scan chain, which consumes a lot of test power. Hence, the switching activities in deterministic pattern generation are effectively reduced in [21], and the low transition random test pattern generator reduces switching activity of CUT by reducing transitions during scan testing [17]. A low power scheme adopts dual LFSR reseeding [13] and scan slice overlapping [23], respectively. Each pattern is partitioned into several overlapping slice sets, no transition is produced and no specified bits need to be generated via LFSR reseeding in the overlapping block [23]. A low-power linear-decompression scheme which operates X-specified and X-filling successively enhance seed encoding efficiency and reduce shift and capture power consumption [16]. And hold cubes-based low power test scheme are presented, while additional test storage and hardware overhead are required for the hold cubes [9]. A low power scan architecture is used to generate tests for stuck-at faults [12]. Above-mentioned, this paper develops a LFSR reseeding-oriented low-power test-compression scheme for scan testing.

2 LFSR Reseeding Procedure

LFSR reseeding generates pseudo-random sequences for scan testing in a typical test compression scheme. External LFSR circuit shown in Fig. 1. $a_k \in \{0, 1\}$ denotes output sequence of flip-flop, $c_i \in \{0, 1\}$ denotes a constant coefficient, $\mathbf{A}(0)$ indicates the initial seed vector. The states of all registers are represented by L-dimensional vectors shown as follows:

$$\mathbf{A}(t) = a_1(t)a_2(t)a_3(t)\cdots a_i(t)\cdots a_L(t) \tag{1}$$

The state transfer equation is summarized as follows:

$$\mathbf{A}(t+k) = \mathbf{V}^k \cdot \mathbf{A}(t) \tag{2}$$

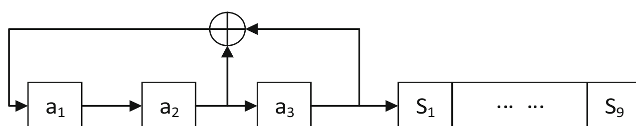

Fig. 2 A three-order LFSR connected to scan chain

For a scan design circuit, most faults can be detected through pseudo-random test vectors, the remains can be detected through deterministic test vectors. The latter vectors are coded as a short seed set and then stored in ROM. During LFSR reseeding, the seed vector is automatically loaded to scan chain for a certain clock period until all seed vectors are expanded into target test pattern for scan testing.

Supposing a given LFSR with L order-characteristic polynomial and test cube contains s specified bits, which means that the linear system of equations contains L unknowns and s equations. Each specified bit corresponds to one state equation, so LFSR seed vector is solved from all equations by Gaussian-Jordan elimination method. Since the compression ratio of LFSR reseeding scheme largely depends on the number of specified bits, the number of specified bits is reduced to decrease the number of equations, which results in a short length of seed vector and small size of LFSR.

A reseeding example is illustrated by three-order LFSR in Fig. 2. $S_i \in \{0, 1\}$denotes the state of scan chain. Supposing test vector 1X1010X1X loaded into scan chain, LFSR characteristic polynomial is $f(x) = 1 + x^2 + x^3$, the state transition matrix is $\mathbf{V} = \left|\begin{matrix} 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{matrix}\right|$.

Only the specified bits is applied to solve the seed vector, the state equations are simplified in Eq. 3:

$$\begin{bmatrix} S_2(9) \\ S_4(9) \\ S_5(9) \\ S_6(9) \\ S_7(9) \\ S_9(9) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 1 \end{bmatrix} \tag{3}$$

Therefore, the equations have a unique solution: $A(0) = a_1a_2a_3 = 011$, and test cube 1X1010X1X is encoded as seed vector 011, it is loaded into LFSR circuit as initial state for scan

Table 1 Block encoding rule

Block types of original test data	Encoded test data	
	Hold flag	Block data
0 compatible block	**0**	0XXXXXXX
1 compatible block	**0**	1XXXXXXX
Incompatible block	**1**	Original data
Unspecified block	**X**	XXXXXXXX

Hold flag is marked bold to highlight it's significance

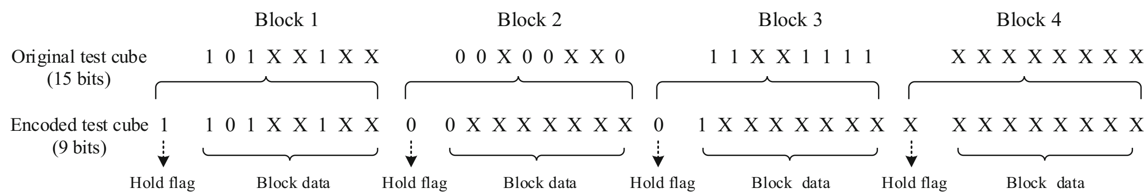

Fig. 3 Test data encoding illustration

testing. After 9 clock cycles, the test pattern 1,110,100,011 compatible with the original test cube is generated.

3 LFSR Reseeding-Based Test Compression Scheme

3.1 Block Encoding Rule

To reduce the number of specified bit for high compression ratio and enhance the consistency between adjacent bits for low test power consumption, test cube is divided into 4 types of data blocks with equal length, namely: 0 compatible, 1 compatible, incompatible and the unspecified blocks. Different types of blocks are coded with different block markers. The block encoding rule is shown in Table 1. Here, 0 compatible block only includes specified bits 0, 1 compatible block only includes specified bits 1, incompatible block includes two kinds of specified bits (1 and 0), unspecified blocks include no specified bit. A hold flag denotes block types, it indicates whether this data block is directly generated by LFSR circuit or not. In addition, the first bit in 0 and 1 compatible block are set to 0 and 1, respectively. Only the first bit in compatible block is generated by LFSR circuit, while the other bits are automatically consistent with the first bit. Obviously, the encoding rule effectively reduces the number of specified bits, and the optimized filling scheme is involved in reducing the shift power consumption.

A test cube is called to illustrate the block encoding rule in Fig. 3. Original test cube with 15 specified bits is divided into four type's blocks, while the encoded test cube only contains 9 specified bits (3 for hold flags and 6 for specified bits). Thereby, a high compression ratio can be achieved in this way. Moreover, no logic transitions occur in blocks 2 and 3 because all data bits in block keep unchanged during LFSR reseeding, which is beneficial for test power consumption reduction.

3.2 Test Cube Optimized Encoding Procedure

The optimized encoding procedure is demonstrated as follows. As shown in Table 2, there are 67 specified bits in a 180 bit test set, it consists of five 32-bit test cubes, and each test cube is divided into four 8-bit data blocks.

According to block encoding rule shown in Table 1, each block is encoded as a 1-bit hold flag and 8-bit test data in Table 3. The encoded test cube only contains 47 specified bits (15 for hold flag and 32 for specified bit). Compared to the original test cube, specified bits are reduced 20 bit through optimized block encoding. Hold flag taken as 0 is regarded as compatible block, which devotes to test power consumption reduction to a large extent.

Above all, the number of specified bits in test set determines the test storage for LFSR reseeding, and the additional hold flag will increase the specified bits. Hence, there is a trade-off between power consumption and test storage.

3.3 Dividing into Hold Flag Partition Sets

Hold flag cube shown in Table 4 (a) is constructed by all hold flags of each block in Table 3, which includes 15 specified bits. It is noticed that lots of hold flags are compatible with each other. Namely, there is no any conflict among all specified bits in blocks. If a hold flag could be shared by multiple test cubes, it only need to be loaded once for each hold flag-compatible set during scan testing. In Table 4 (b), all test cubes are reordered and separated into a compatible set with the compatible hold flag. In Table 4(c), hold partition set is further

Table 2 Original test cube

Original test cube	Block 1	Block 2	Block 3	Block 4
1	00X00XX0	101XX1XX	1X00XX0X	XXXXXXXX
2	0XX1XXX0	X111X11X	XXXXXXXX	X00X00X0
3	X000XXX0	01X0X0XX	X01X00XX	XX1111X1
4	XXX101X0	XXXXXXXX	X00X00X0	1XX111X1
5	XXXXXXXX	11XX1111	XX00X00X	XXXXXXXX

Table 3 Encoded test cube

Encoded test cube	Block 1		Block 2		Block 3		Block 4	
	Hold flag	Test data	Hold flag	Test data	Hold flag	Test data	Hold flag	Test data
1	**0**	0XXXXXXX	**1**	101XX1XX	**1**	1X00XX0X	**X**	XXXXXXXX
2	**1**	0XX1XXX0	**0**	1XXXXXXX	**X**	XXXXXXXX	**0**	0XXXXXXX
3	**0**	0XXXXXXX	**1**	01X0X0XX	**1**	X01X00XX	**0**	1XXXXXXX
4	**1**	XXX101X0	**X**	XXXXXXXX	**0**	0XXXXXXX	**0**	1XXXXXXX
5	**X**	XXXXXXXX	**0**	1XXXXXXX	**0**	1XXXXXXX	**X**	XXXXXXXX

Hold flag is marked bold to distinguish it from test data

Table 4 Updating hold partition sets

(a) Hold cube construction		(b) Hold partition sets		(c) Updated hold partition sets		
Encoded test cube	Hold flag cube	Reordered test cube	Hold flag cube	Reordered test cube	Updated Hold flag	Hold flag cube
1	**011X**	2	**10X0**	2	***1***	**1000**
2	**10X0**	4	**1X00**	4	***0***	**XXXX**
3	**0110**	5	**X00X**	5	***0***	**XXXX**
4	**1X00**	1	**011X**	1	***1***	**0110**
5	**X00X**	3	**0110**	3	***0***	**XXXX**

In order to highlight the difference, hold flag cube is marked bold and updated hold flag is marked bold italic

updated by extra update hold flag, which indicates whether hold flag for the current test cube needs to be updated or not.

Noteworthy, five test cubes are grouped into two hold flag-compatible sets. The first set contains test cubes 2, 4 and 5, whose update flag bit and hold flag cube are 100 and 1000.The second set contains test cubes 1 and 3, whose update flag bit and hold flag cube are 10 and 0110.

A test set is called to illustrate the optimized encoding scheme in Table 5. Obviously, hold flag cubes are only loaded twice during scan testing, which significantly reduces test application time.

Compared to the encoded test cube shown in Table 3, the number of specified bits is further reduced 2 bit through updating hold flag partition. Actually, the scale of test set in industrial circuit is much larger than that of test set shown in Table 5. Thus, the number of specified bits are extremely reduced in this way, which overwhelmingly reduced the number of equations and the computational complexity to solve the seeds.

4 Implementation of Optimized Encoding Algorithm

To implement the above optimized encoding algorithm, the proposed LFSR reseeding-based test compression scheme is illustrated in Fig. 4.

To summarize, block size B, LFSR size L and L_{min} are initialized, test cubes are divided into equal blocks with size B. The optimized encoding algorithm is applied to these blocks until the minimum seed size L_{min} is solved from an appropriate block for LFSR reseeding. B dynamically changes from 1 to 100 by a greedy algorithm for efficient test compression. L_{min} solving process is implemented in Table 6.

5 Hardware Implementation of Decoder Architecture

5.1 Decoder Architecture

LFSR reseeding circuit is demonstrated in Fig. 5, which consists of ATE, LFSR, phase shifter, finite state machine (FSM), decoder architecture and circuit under test (CUT). Here, FSM controller is composed of a bit counter and some small combinational logic. 1-bit Update Hold Flag-shifter (UF-SR) and Hold Flag-shifter (HF-SR) are used to store update hold flag and hold flag, respectively. Sign-SR (Sign-SR) is used to store the first sign bit of each block. Obviously, hardware overhead is mainly consumed on a small FSM controller, two 2-to-1 MUX, one HF-SR and HF-SR per scan chain, two 2-to-1 MUX and one Sign-SR per block.

Obviously, the proposed hardware architecture requires additional circuitry such as LFSR, phase shifter and decoder architecture, which would change the timing analysis of BIST architecture, so a timing analysis is further conducted to provide more timing information in Fig. 6. During scan

Table 5 Optimized test set

Reordered test cube	Updated hold flag	Hold flag cube	Block 1 Test data	Block 2 Test data	Block 3 Test data	Block 4 Test data
2	**1**	1000	0XX1XXX0	0XXXXXXX	XXXXXXXX	0XXXXXXX
4	**0**	XXXX	XXX101X0	XXXXXXXX	0XXXXXXX	1XXXXXXX
5	**0**	XXXX	XXXXXXXX	1XXXXXXX	1XXXXXXX	XXXXXXXX
1	**1**	0110	0XXXXXXX	101XX1XX	1X00XX0X	XXXXXXXX
3	**0**	XXXX	0XXXXXXX	01X0X0XX	X01X00XX	1XXXXXXX

Updated hold flag is marked bold italic for highlight it

testing, seed vectors pre-stored in ROM are loaded into LFSR reseeding circuit under the control of *LFSR_clk*, these seeds are decompressed into test patterns in decoder architecture under the control signal of the FSM. These test patterns are scanned into multiple scan chains for circuit under testing in parallel under the control of *Scan_clk*. Test response is captured and then compressed by a compactor for comparing with golden seed vectors in ATE for test results.

5.2 Decoding Procedure

Each scan chain is divided into one or more blocks. Supposing size B is the number of blocks per scan chain. Each scan chain has a HF-SR whose size is equal to B. LFSR reseeding is used to generate all test data for each test cube, which consists of three components, namely: update flag, hold flag and test data with a sign flag. The format for test data coming out of LFSR reseeding for each test cube is given in Fig. 7.

A small finite-state machine (FSM) controller controls whether the data coming out from the LFSR is stored or not. In the first clock cycle, LFSR circuit generates a single update hold flag bit and stores it in UF-SR. If the update hold flag is 1, LFSR circuit in next B clock cycles generates hold flags for each scan chains and loads them into HF-SRs. If update hold flag is 0, the HF-SRs are not loaded. Supposing the length of each scan chain be M. LFSR circuit generates test data for the

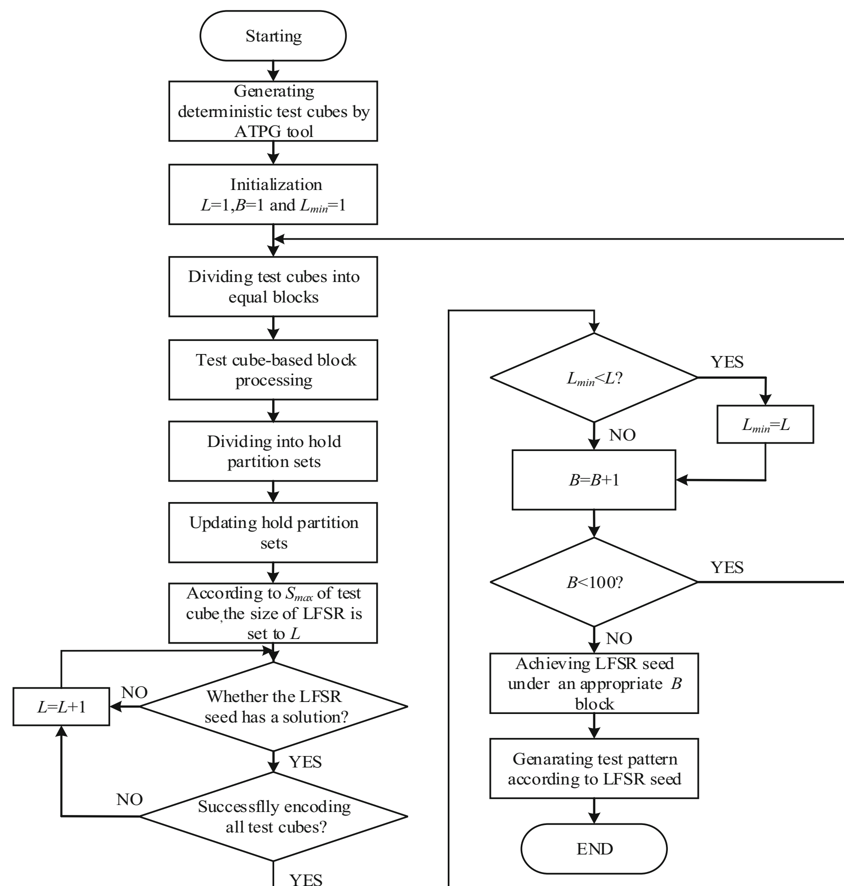

Fig. 4 Proposed LFSR reseeding-based test compression scheme

Table 6 $L_{\min}$ solving process

Algorithm: LFSR reseeding-based test compression algorithm

Input: Test cubes

Output: The size L_{min} of LFSR seed and the size B of blocks

```
0.  Block-based test cube processing
1.  Initialize block size B=0, the size of LFSR L=1, the smallest size L_min =1
2.  According to S_max of test cube, the size of LFSR is set to L to compute the LFSR seeds;
3.  If (equation has a solution)
4.      If (all test cubes are successfully encoded)
5.          If (L_min<L) let L_min=L,B=B+1;
            Else B=B+1;
6.              If (B<100) go to step 2;
                Else solving L_min and LFSR seeds under an appropriate B block;
                    generate the test pattern according to seed;
                End if
            End if
        Else L=L+1; Then go to step 5;
        End if
    Else L=L+1; Then go to step 4;
    End if
```

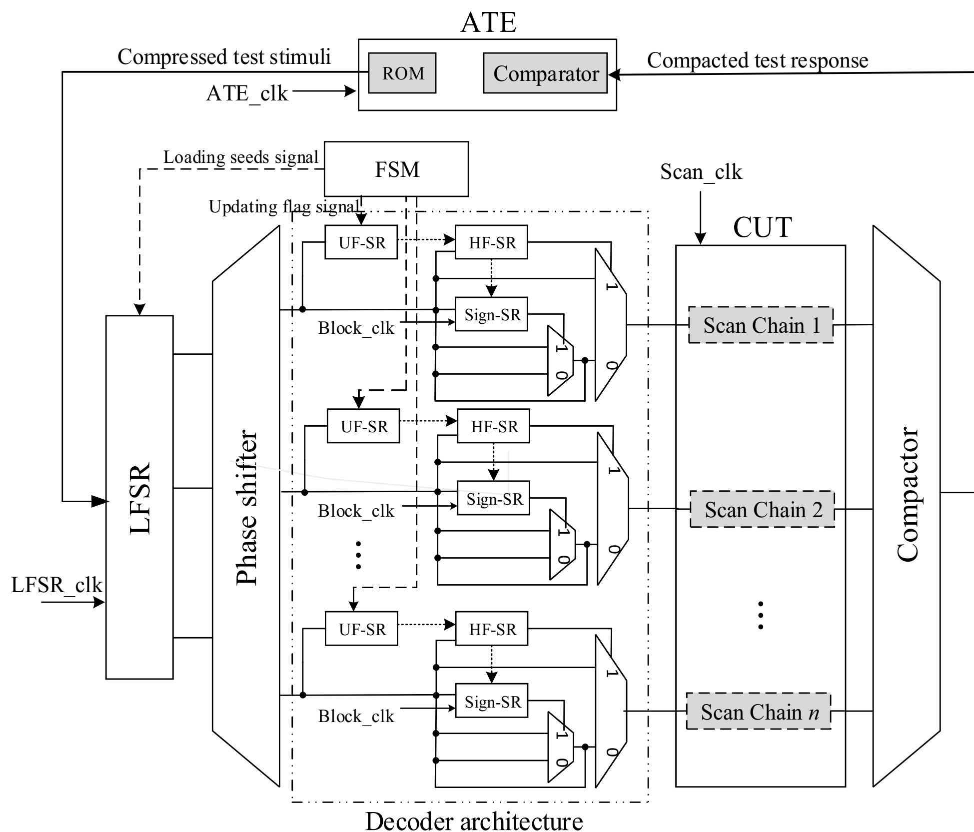

Fig. 5 Hardware architecture of LFSR reseeding circuit

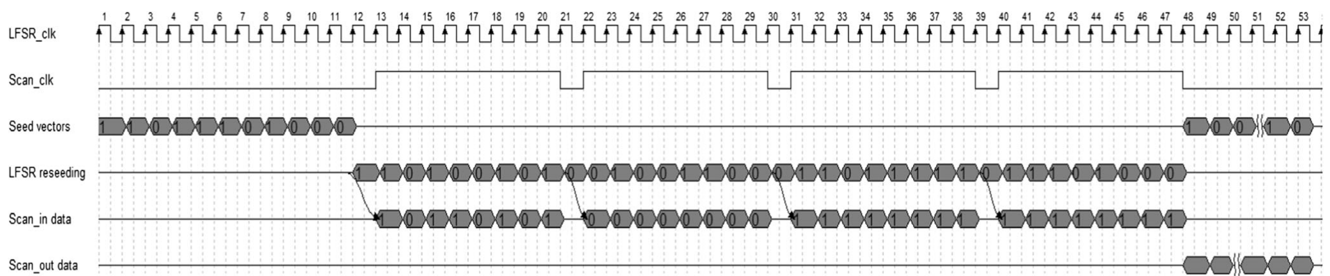

Fig. 6 Timing diagram of test data decompression

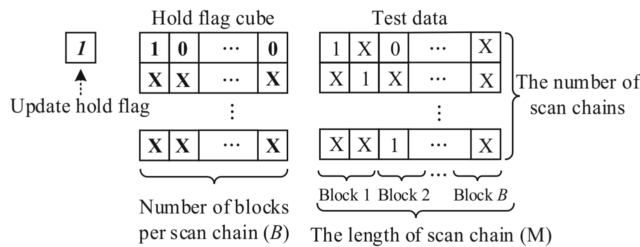

Fig. 7 Data format generated by LFSR-reseeding

next M clock cycles. During each M/B clock cycle, if the corresponding hold flag is 1, the scan chain is loaded from the LFSR, If the corresponding hold flag is a 0, the LFSR generates a single bit, which is the sign flag, its' value is repeatedly shifted into the scan chain for M/B clock cycle, and test data from the LFSR is ignored.

After each M/B clock cycle, the HF-SR is shifted so that the next hold flag becomes active for its corresponding block and used as the control signal of a multiplexer. Once the scan chains is filled, test pattern is applied to circuit under test, and the test response is captured from scan chain. The above processing is repeated to generate the next test pattern. Noteworthy, the size of the HF-SR and Sign-SR dominantly determines the resource occupation, which depends on the number of scan chains and the total number of blocks.

5.3 Decoding Timing Analysis

As shown in Table 7, in order to verify the designed decode architecture, four test cubes with four data blocks are taken as an encoding example to solve the seed vector.

Four seed vectors are expanded to four test cubes by LFSR reseeding in Table 8. Obviously, all decoded test cubes are completely compatible with the original test cubes.

Moreover, the functional verification and timing analysis of decoder is performed on Mentor's EDA software Modelsim tool. The simulation results of Fig. 8 are in perfect agreement with the timing analysis.

6 Circuit Simulation and Results Analysis

The experiments for the six largest ISCAS Benchmark circuits are performed on CPU with the Intel core 3.2 GHz and 4G RAM, and test sets are generated by Atalanta ATPG tool with dynamic compaction. Basic information of Benchmark circuit is shown in Table 9. PI and PO represent the numbers of input and output port, respectively. INV, LG and FA indicates the total numbers of inverter, logic gate and stuck-at-fault.

Pseudo random test set is applied to detect easily faults. Furthermore, LFSR reseeding scheme is used to test faults that are failure to be detected after 10,000 pseudo random test patterns, whose test set is generated by using ATPG tool (Lee & Ha, 1993). Fault simulation is performed on HOPE (Lee& Ha, 1992) simulator.

6.1 Test Compression Ratio Analysis

As shown in Table 10, the proposed scheme are compared with previous encoding works in the compression ratios, S_{max} is the largest number of specified bits in all test cubes. Compression ratio is defined as $CR\ \% = [(|T_D| - |T_E|)/T_D] \times 100\%$, where $|T_D|$ is the size of original test set, $|T_E|$ is the size of seed vectors.

Table 7 Test sets after optimization of encoding and grouping

NO.	Test cubes	Test cubes with an extra hold flag	Solved seeds
1	101XX1XX 00X00XX0 11XX1111 XXXXXXXX	**1**101XX1XX **0**0XXXXXXX **0**1XXXXXXX **X**XXXXXXXX	11011101000
2	1X11X1X1 XXXXXXXX 0X00X00X 01X0X1X	**0**1XXXXXXX **X**XXXXXXXX **0**0XXXXXXX **1**01X0X1XX	00101000100
3	XXXXXXXX 11XXX1XX 0XX0XXX0 0XX10XXX	**X**XXXXXXXX **0**1XXXXXXX **0**0XXXXXXX **1**0XX10XXX	00100010001
4	0XXX00XX 0XX0X1XX XXXXXXXX 1X1X1XXX	**0**0XXXXXXX **1**0XX0X1XX **X**XXXXXXXX **0**1XXXXXXX	11010010100

Hold flag is marked bold for highlight it

Table 8 An illustration of LFSR reseeding

NO.	Test cubes	Seed vectors	Test data generated by LFSR reseeding	Load into scan chain	Compatibility
1	101XX1XX 00X00XX0 11XX1111 XXXXXXXX	11011101000	1101001010010011000110111110111101000	10100101 00000000 11111111 11111111	100%
2	1X11X1X1 XXXXXXXX 0X00X00X 01X0X1XX	00101000100	0110111010111010100010100001010000100	11111111 11111111 00000000 01000100	100%
3	XXXXXXXX 11XXX1XX 0XX0XXX0 0XX10XXX	00100010001	0100111010101110100000101001000100001	11111111 11111111 00000000 00010001	100%
4	0XXX00XX 0XX0XX1XX XXXXXXXX 1X1X1XXX	11010010100	0001011011001001101111011110100101000	00000000 00100110 11101111 11111111	100%

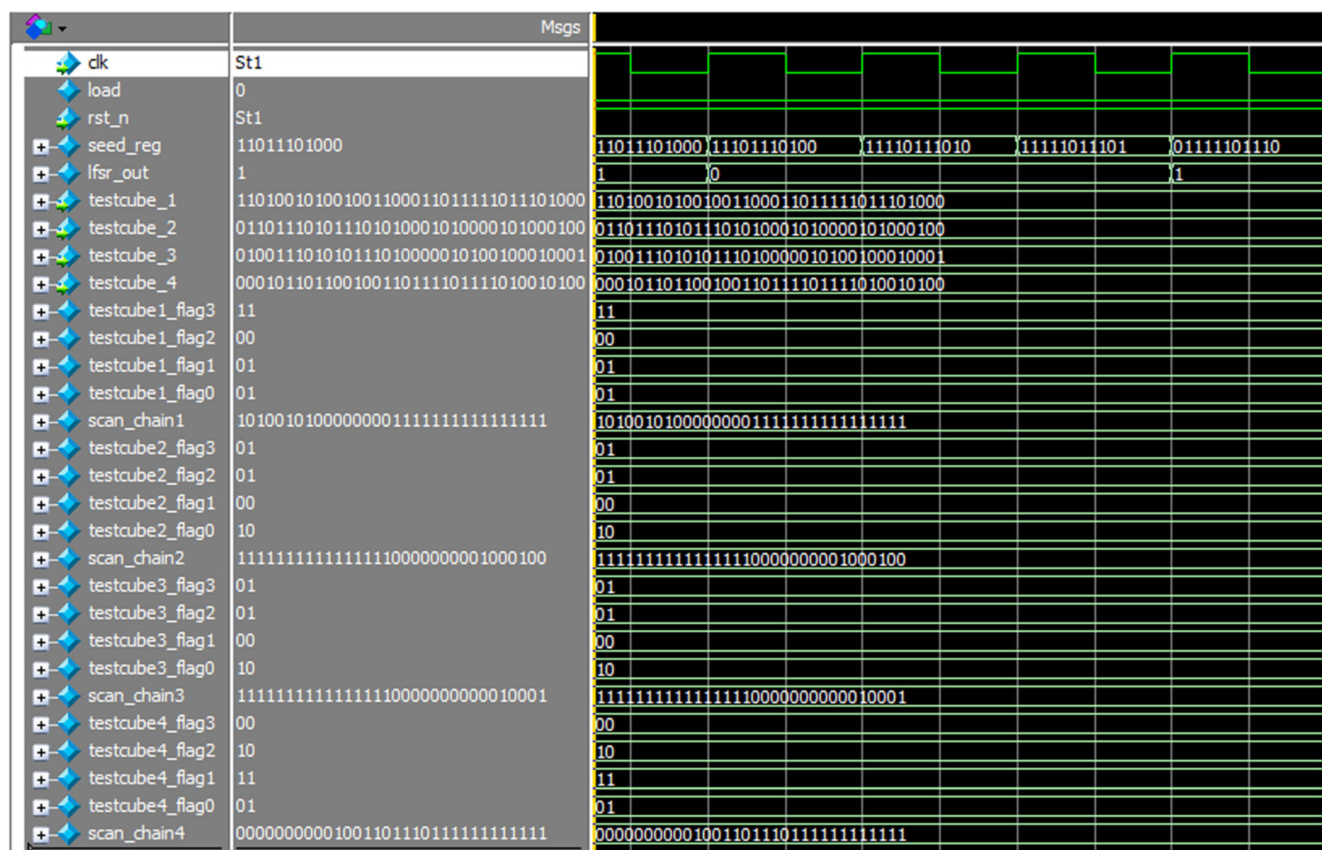

Fig. 8 Simulation results of decompression flow

Table 9 Basic information of Benchmark circuit

Circuits	PI	PO	INV	LG	FA
S5378	214	213	1775	1004	4551
S9234	247	250	3570	2027	6927
S13207	700	790	5378	2573	9815
S15850	611	684	6324	3448	11725
S38417	1664	1472	13470	8709	31180
S38584	1464	1730	7805	11448	36303

Obviously, compared with EFDR [3] and Hybrid coding [19] scheme, the proposed optimized encoding algorithm achieves a better compression effect for overwhelming majority circuits, and the average compression ratio achieves up to 92.9%. For circuits S15850 and circuits S38417, the compression ratio is slightly lower than that of hybrid coding scheme. However, the decoding structure of hybrid coding scheme is more complex, thus it causing long test application time and high hardware overhead.

Table 10 Compression ratio comparison with previous encoding scheme

Circuit	EFDR [3]			Hybrid coding [19]				The proposed scheme				
	T_D	T_E	CR%	T_D	T_E	S_{max}	CR(%)	T_D	T_E	S_{max}	LFSR size	CR(%)
S5378	20,758	9739	53.08	5992	897	18	85	6032	551	17	19	91.2
S9234	25,939	15,057	41.94	73,112	13,927	49	81	221,082	37,584	51	54	83.2
S13207	163,100	32,144	80.29	220,500	8574	22	96.1	459,900	15,768	22	24	96.7
S15850	57,434	24,263	57.75	163,748	9478	48	**94.2**	380,042	26,124	41	42	93.1
S38417	113,152	51,357	54.61	2,201,472	73,841	66	**96.7**	3,675,776	172,302	86	78	95.3
S38584	199,104	71,121	64.28	456,768	10,787	54	97.6	717,360	16,660	54	34	**97.7**
Average	96,581	33,947	58.7	520,265	–	–	91.8	875,170	44,832	–	–	**92.9**

The best compression ratio is marked bold

Table 11 Compression ratio compared with the previous schemes

Benchmark circuit	DUAL-LFSR–reseeding scheme [13]			Modified scan-slice-overlapping scheme [23]			Low-power LFSR Reseeding scheme [9]			The proposed scheme		
	T_D	T_E	CR%	T_D	T_E	CR%	T_D	T_E	CR%	T_D	T_E	CR%
S9234	60,750	19,440	68%	39,273	10,867	72.33%	49,057	10,302	79%	221,082	37,584	83.2%
S13207	196,717	11,803	94%	165,200	15,579	90.57%	174,733	10,484	94%	459,900	15,768	96.7%
S15850	145,180	14,518	90%	76,986	14,203	81.55%	163,014	11,411	93%	380,042	26,124	93.1%
S38417	827,925	66,234	92%	164,736	46,966	71.49%	643,040	32,152	95%	3,675,776	172,302	95.3%
S38584	397,250	23,835	94%	199,104	38,805	80.51%	445,029	31,152	93%	717,360	16,660	97.7%
Average	–	–	87.6%	–	–	79.3%	–	–	90.8%	–	–	93.2%

Overall, the proposed LFSR reseeding-based test compression scheme outperformed in test data compression and decoder hardware implementation

Table 12 Area overhead for decoder circuit

Circuit	Partial LFSR reseeding [8]			The proposed scheme					
	Decomp. (bit)	Num. Scan Chains	Decomp. Area	Sign-SR/HF-SR	Number Mux + OR gate	Number Scan Chains	Decomp. (bit)	Decomp. area	Increased Area ratio
S9234	–	–	–	10	10	10	54	13,624.6	–
S13207	64	20	21,467.9	20	20	20	24	23,277.3	8.4%
S15850	66	20	23,595.2	20	20	20	42	25,139.3	6.5%
S38417	120	40	58,045.7	40	40	40	78	63,057.6	8.6%
S38584	120	40	56,671.4	40	40	40	34	61,083.1	7.8%

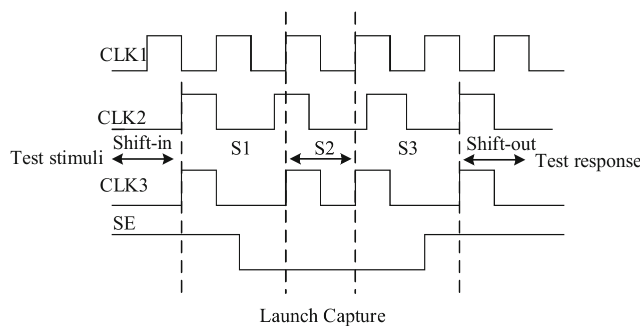

Fig. 9 Timing diagram for test vectors shift-in and shift-out

In addition, the compression ratios compared with other LFSR reseeding schemes are presented in Table 11. Obviously, the proposed LFSR reseeding-based test compression scheme achieves a high compression ratio for all circuits than Dual-LFSR [13], Modified scan-slice [23] and low-power LFSR [9], the average compression ratio achieves up to 93.2%.

6.2 Area Overhead for Decoder Circuit

The area overhead for decompression circuit is shown in Table 12. The standard library for TSMC

Table 13 Shift power consumption evaluation of the proposed scheme

Benchmark circuit	Shift-in		Shift-out	
	Ave power	Peak power	Ave power	Peak power
S9234	4321	11,493	7107	13,469
S13207	25,218	62,729	41,389	71,123
S15850	30,794	58,804	42,276	75,396
S38417	190,143	399,805	261,392	487,426
S38584	135,627	321,001	217,500	406,048

0.18 μm process [1] is applied to calculate the area overhead.

As shown in Table 12, the proposed scheme consumes a little more area overhead than that of Partial LFSR reseeding. Nevertheless, it increases compression ratio on average owing to block-based coding during Scan-in. In addition, area overhead of the decompression architecture is generally acceptable compared with the prior schemes.

6.3 Test Power Consumption Analysis

Test power includes shift power and capture power in BIST scheme. Actually, shift mode including shift-in mode and shift-out mode occupies most test time during chip testing, so the shift power determines the average power of circuit under testing [18]. The timing diagram of shift-in and shift out in test per-scan architecture is presented in Fig. 9. Here, CLK1 is the at-speed clock in CUT to be applied in capture phase, CLK2 is the shift clock signal to control test vectors are shifted-in/out of the scan chains, and CLK3 is the clock signal that the sequential elements on the scan chain will receive. It obviously noted that the shift-in power is associated with test vectors and shift-out power is associated with test responses.

To estimate how shift powers associates with test patterns, they are calculated by the weighted transition metric (*WTM*) [2], whose value reflects the inversion states of circuit under test, so it is used to evaluate the scan shift power caused by the corresponding test patterns.

Assuming that test set and test response are $T=\{T_1, T_2, T_3, \cdots, T_m\}$ and $R=\{R_1, R_2, R_3, \cdots, R_m\}$, respectively. The length of each test cube is represented as $T_i=\{t_{i1}, t_{i2}, t_{i3}, \cdots, t_{in}\}$, $1\le i\le m$, $1\le j\le n$, t_{in} denotes the *n*-th bit in the *i*-th test pattern. Hence, *WTM* of shift-in and shift-out are estimated as follows:

$$WTM_{i-\mathrm{in}} = \sum_{i=1}^{n-1} (n-j)*\left(t_{i,j}\oplus t_{i,j+1}\right) \quad (4)$$

$$WTM_{i-\mathrm{out}} = \sum_{i=1}^{n-1} j*\left(R_{i,j}\oplus R_{i,j+1}\right) \quad (5)$$

For N test patterns, the average shift-in/out power consumption $P_{shift-in}$ and $P_{shift-out}$ are respectively estimated as follows:

$$P_{shift-in} = \frac{1}{N}\sum_{i=1}^{N} WTM_{i-\mathrm{in}} = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{n-1}(n-j)*\left(t_{i,j}\oplus t_{i,j+1}\right) \quad (6)$$

$$P_{shift-out} = \frac{1}{N}\sum_{i=1}^{N} WTM_{i-\mathrm{out}} = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{n-1} j*\left(R_{i,j}\oplus R_{i,j+1}\right) \quad (7)$$

Peak scan-in power consumption P_{peak} are estimated as follows:

$$P_{peak} = \max_{1\le i\le m} WTM_i \quad (8)$$

Consequently, the average and peak power consumption of the proposed scheme for shift-in and shift-out mode are evaluated in Table 13.

The compression ratio and shift-in power consumption are compared with other schemes in Table 14. Compared with the former three schemes, the proposed scheme has obvious advantages in test compression ratio for each circuit. In addition, it significantly outperformed in shift-in power reduction than the first and third schemes and slightly underperformed in shift-in power reduction than the second scheme.

Overall, the proposed LFSR reseeding-based test compression scheme shows a trade-off between the shift-in power consumption reduction and test compression ratio.

Table 14 Comparison of compression ratio and shift-in power consumption with other schemes

Benchmark circuit	DUAL-LFSR reseeding scheme [13]		Modified scan slice overlapping [23]		Low-power LFSR reseeding scheme [9]		The proposed scheme		
	Compression ratio	Power reduction	Compression ratio	Power reduction	Compression ratio	Power reduction	Compression ratio	Power reduction	Peak power
S9234	68%	24%	72.33%	70.89%	79%	53%	83.2%	62.3%	11,493
S13207	94%	25%	90.57%	92.9%	94%	53%	96.7%	79.4%	62,729
S15850	90%	25%	81.55%	78.84%	93%	52%	93.1%	76.1%	58,804
S38417	92%	25%	71.49%	72.13%	95%	52%	95.3%	73.5%	399,805
S38584	94%	25%	80.51%	76.35%	93%	40%	97.7%	70.8%	321,001
Average	87.6%	24.8%	79.3%	78.2%	90.8%	50%	**93.2%**	**72.4%**	–

The average compression ratio and power reduction are marked bold for highlight it

7 Conclusion

The LFSR reseeding-oriented deterministic BIST scheme is an efficient technology for compressing test data. The proposed low-power test-compression scheme combines clustering scan-block with updating hold flag, which provides a valid way to significantly reduce test storage and test power consumption for scan design. In addition, the optimized encoding algorithm can be applied to either a BIST environment or test compression schemes based on LFSR reseeding to satisfy certain test power constraints, so it is practical to test large-scale industrial circuit.

Acknowledgments This research work was supported by National Natural Science Foundation of China (61001049), Beijing Natural Science Foundation (4172010) and Research Fund from Beijing Innovation Center for Future Chips (KYJJ2018009).

References

1. Artisan Components (2001) TSMC 0.18μm Process 1.8-volt SAGE-X standard cell library databook. DB-SX-TSM003–3.1/18
2. Czysz D, Kassab M, Lin X, Mrugalski G, Rajski J, Tyszer J (2009) Low-power scan operation in test compression environment. IEEE Tran CAD 28(11):1742–1753
3. EI Maleh AH (2008) Test data compression for systern-on-chip using extended frequency-directed run-length code. IET Comput Dig Tech 2(3):155–163
4. Filipek M et al (2015) Low-power programmable PRPG with test compression capabilities. IEEE Trans Very Large Scale Integr (VLSI) Syst 23(6):1063–1076
5. Hellebrand S, Rajski J, Tarnick S (1995) Built-in test for circuits with scan based on reseeding of multiple-polynomial linear feedback shift registers. IEEE Trans Comput 44(2):223
6. Kim HS, Kim YJ, Kang S (2003) Test-decompression mechanism using a variable-length multiple-polynomial LFSR. IEEE Trans Very Large Scale Integration Syst 11(4):687
7. Koenemann B (1991) LFSR-coded test pattern for scan designs. Proc. Eur. Test Conf, Munich, pp 237–242
8. Krishna CV, Jas A, Touba NA (2002) Reducing Test Data Volume Using LFSR Reseeding with Seed Compression. Proc. Int'l Test Conf. (ITC 02), IEEE CS Press, 321–330
9. Lee J, Touba NA (2007) LFSR-reseeding scheme achieving low-power dissipation during test. IEEE Trans Comput Aided Design Integr Circuits Syst 26(2):396
10. PrasadaRao RV, Varaprasad NA, Babu GS, Mohan CM (2013) Power optimization of linear feedback shift register (LFSR) for low power BIST implemented in HDL. Int J Modern Eng Res (IJMER) 3(3):1523–1528
11. Rajski J, Tyszer J, Kassab M, Mukherjee N (2008) Embedded deterministic test. IEEE Trans CAD 23:776–792
12. Rajski J, Moghaddam EK, Reddy SM (2011) Low Power Compression Utilizing Clock-Gating. in Proc. Intl. Test Conf., Paper 7.1
13. Rosinger PM, Al-Hashimi BM, Nicolici N (2003) Dual multiple-polynomial LFSR for low-power mixed-mode BIST. IEE Proc Comput Digital Tech 150(4):209–217
14. Sakthivel PA, Kumar N, Mayilsamy T (2012) Low transition test pattern generator architecture for built in self-test. Am J Appl Sci 9(9):1396–1406
15. Saxena J, Butler KM, Jayaram VB (2003) A case study of irdrop in structured at-speed testing. Proc IEEE Int Test Conf:1098
16. Tian CHEN, Dandan SHEN, Xin YI, Huaguo LIANG, Xiaoqing WEN, Wei WANG (2016) Reseeding-oriented test power reduction for linear-decompression-based test compression architectures. IEICE Trans Inf Syst 99(11):2672–2681
17. Varatharajan R, Lekha R. (2012). A Low Power BIST TPG for High Fault Coverage.Information Engineering and Electronic Business, pp. 19–24
18. Wang S, Sato Y, Miyase K, Kajihara S (2012) Low power BIST for scan-shift and capture power. Proc IEEE Asian Test Symp:173–178
19. Würtenberger A, Tautermann CS, Hellebrand S (2003) .A hybrid coding strategy for optimized test data compression.Proceeding of International Test Conference (ITC) 451–459
20. Xiang D, Zhao Y, Chakrabarty K, Fujiwara H (2008) A reconfigurable scan architecture with weighted scan-enable signals for deterministic BIST. IEEE Trans CAD 27(6):999–1012
21. Yuan H, Guo K, Sun J, Zijian (2016) A power efficient test data compression method for SoC using alternating statistical run-length coding. J Electron Testing-theory Appl 32(1):59–68
22. Zhang C, Yu W, Wang Q, Shi Y (2015) Fast random walk based capacitance extraction for the 3-D IC structures with cylindrical intertier-via. IEEE Trans Comput-Aided Des Integr Circuits Syst 34(12):1977–1990
23. Zhou B, Ye YZ, Wu XC et al. (2009) Reduction of test power and data volume in BIST scheme based on scan slice overlapping. IEEE Int Symp Circ Syst 2737

Haiying Yuan received her Ph.D., M.S. and B.S. from University of Electronic Science and Technology of China. She is an associate professor in Beijing University of Technology. Her research interests include VLSI and System-on-Chip design, verification and fault diagnosis, signal detection and information processing.

Changshi Zhou is pursuing his M.S. degree at Beijing University of Technology. His research interests include integrated circuit design, testing, design for testability and fault diagnosis.

Xun Sun is a Ph.D. researcher at Tsinghua University. His research interests include satellite navigation signal processing and system simulation, communication and information system.

Kai Zhang is pursuing his M.S. degree at Beijing University of Technology. His research interests include integrated circuit design, testing, design for testability and time triggered Ethernet.

Tong Zheng is pursuing his M.S. degree at Beijing University of Technology. His research interests include integrated circuit design, testing, design for testability and time triggered Ethernet.

Chang Liu is pursuing her M.S. degree at Beijing University of Technology. Her research interests include signal detection, information processing and pattern recognition.

Xiuyu Wang is pursuing her M.S. degree at Beijing University of Technology. Her research interests include signal detection, information processing and fault diagnosis.