

Modified PRPG for Test Data Reduction Using BAST Structure

Zheng Hong Cai, Hiroyuki Yotsuyanagi and Masaki Hashizume

Department of Electrical and Electronic Engineering, Tokushima University
 2-1 Minami-Josanjima, Tokushima 770-8506, Japan
 E-mail: {cai, yanagi4, tume}@ee.tokushima-u.ac.jp

Abstract

In order to reduce the volume of test data, built-in self test (BIST) and BIST-aided scan test (BAST) techniques have been proposed. To provide the test pattern generated by an automatic test pattern generator (ATPG) using BAST, we enhanced the structure of a pseudorandom pattern generator (PRPG) by inserting MUXes and NOT gates in the linear feedback shift register (LFSR) based on correlations of ATPG patterns. The procedures can achieve about 15 to 56% reduction in the volume of test data for BAST.

1. Introduction

For testing highly integrated circuits, the built-in self-test (BIST) [1], [2] technique is used as one of the design-for-testability (DFT) schemes to reduce test data volume. In BIST, test patterns are generated by a pseudorandom pattern generator (PRPG) such as a linear feedback shift register (LFSR) inside a chip and are provided to the scan chains in a circuit under test (CUT). To enhance the test pattern generated by a PRPG, the BIST-aided scan test (BAST) technique, which combines an automatic test pattern generator (ATPG) and BIST, has been proposed [3], [4]. In BAST, to provide the ATPG pattern for scan chains in a CUT, the conflicting bits in the PRPG pattern are flipped by the inverter block such that the PRPG pattern matches the given ATPG pattern. To reduce the volume of test data, some methods of controlling a PRPG have been proposed as stated in [5]. In this paper, we propose a modified PRPG for effective LFSR reseeding.

2. Preliminaries

In the BAST architecture, pseudorandom patterns generated by an LFSR are converted to deterministic ATPG patterns [6], [7], [8]. Figure 1 shows the BAST architecture. The BAST architecture consists of PRPG, inverter block, multiple-input signature register (MISR), decoder block, and X-masking block. Through the inverter block, we can flip some bits in a PRPG pattern to match a given ATPG pattern and provide it to a CUT. In [7], don't care identification was applied to a deterministic pattern set to reduce the number of

BAST codes required for bit flipping. In [9], a scan chain re-ordering method was proposed to reduce the number of bits in PRPG patterns that conflict with the corresponding bits in ATPG patterns.

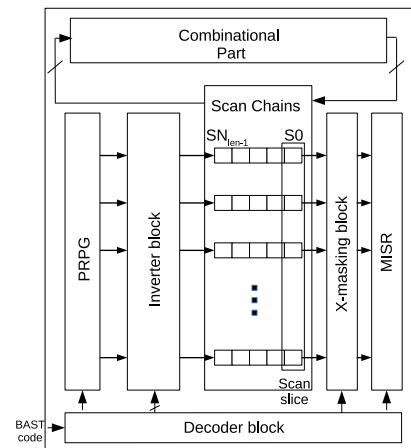


Figure 1: BAST architecture

The control signal for the inverter block is called a BAST code. BAST codes are provided from the tester to the decoder block and indicate which bit is to be flipped.

A BAST code consists of a mode part and an address part. The mode part is used to determine the operation of the LFSR, the inverter block, and the scan chain. The address part is used to indicate which bit to be flipped. The internal state of the flip-flops in the inverter block is called the inverter code. For each scan slice, BAST codes are applied to the decoder block to determine which bits in an inverter code are set to be flipped. The bit of a PRPG pattern is inverted and provided to the scan slice if the corresponding bit in the inverter code has value 1. In the BAST architecture, deterministic ATPG patterns can be provided by using the PRPG and the inverter block.

Figure 2 shows the internal structure of the inverter block consisting of XOR gates and FFs. The inverter block stores the inverter code in the FFs. If the inverter code is 1, the output value of the XOR gate is the inverse of the value of the PRPG signal and is applied to the scan chain. Conversely, if

the inverter code is 0, the output value of the XOR gate is the same as the value of the PRPG signal. The decoder block supplies an inverter code and a reset signal to the inverter block. Hence, the inputs of the XOR gate are fed from the PRPG signal and the internal FF signal.

Figure 3 shows an example of BAST codes for three scan slices. In Fig. 3, we define two operational modes, which are the reset mode R and the invert mode I . In the reset mode, LFSR outputs are provided through the inverter block to scan chains by performing a scan shift, and after that, all FFs in the inverter block are reset. Moreover, in the invert mode, the FF selected by the address is set to 1.

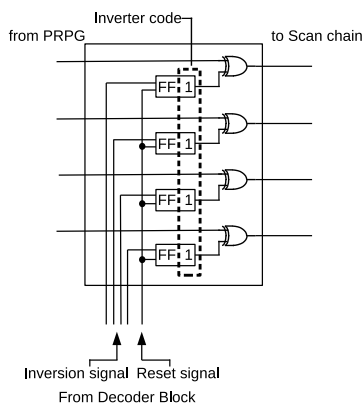


Figure 2: Inverter block

Scan slice		3	2	1
Inverter code	Address0	1 1 0	1 0	0
	Address1	1 0 0	0 0	0
	Address2	0 0 0	0 0	0
	Address3	0 0 0	0 0	0
Operation		I I R	I R	R
BAST code	Operation part	1 1;0; 0 0;0;	1;0; 0;0;	0; 0;0;
	Address part	1 0;0; 0 0;0;	0;0; 0;0;	0; 0;0;

R:Reset, I:Invert
Scan shift code

Figure 3: Example of BAST codes

3. LFSR-reseeding Control Circuit based on Correlation Analysis of ATPG Pattern

BAST can reduce the volume of test data if a small number of bits are to be flipped through the inverter block. Thus, we have managed to improve the LFSR circuit in BAST to reduce the number of BAST codes required for inverter codes.

To provide an ATPG pattern for the scan chains in the CUT, the inverter block flips the conflicting bits such that the PRPG pattern matches the given ATPG pattern. Our proposed method is to insert additional MUXes to the LFSR as shown in Fig. 4.

Table 1 shows the three kinds of reset modes used in the proposed PRPG and the corresponding address part of the BAST codes. We embedded new control signals for the reset mode in the address part of the BAST code. The decoder block outputs two additional signals M and L_0 depending on the reset mode to control the feedback loop of the PRPG.

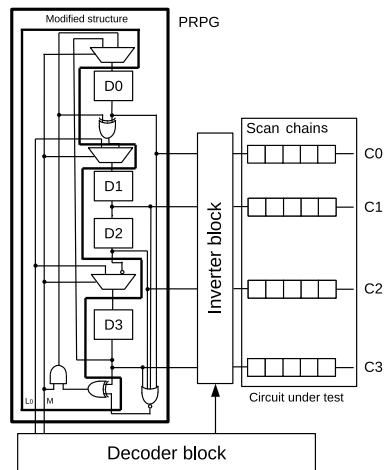


Figure 4: Example of proposed LFSR-reseeding circuit

Table 1: Reset modes for controlling feedback

mode	control signal	action		
Reset	cnt	$\overline{M}L_0$	LFSR	inverter block
00	N	00	normal	Reset all invcodes
	'0'	10	input '0'	Reset all invcodes
	'1'	11	input '1'	Reset all invcodes

In accordance with the ATPG pattern slice, MUXes can switch among normal LFSR pattern modes and set the 1 (or 0) mode using the L_0 and M signals. The FFs with the inserted MUX can be set to 1 (or 0) using the 1 (or 0) mode, therefore, the number of required BAST codes for the inverter block can be reduced.

The procedure to select where to add MUXes is as follows. For each pair of scan chains, the number of ATPG slices with the same logic value is counted for all test patterns. The number of ATPG slices with the other logic value is also calculated.

3.1 Method for making correlation tables

We count the same, conflicting, and don't care logic values in ATPG slices for each pair of scan chains, then form a two-dimensional 16×16 matrix. The compared values are classified as 'same', 'conflict', or 'don't care', and are accumulated in the corresponding two-dimensional matrix. The matrices are called correlation tables of ATPG slices.

For example, Fig. 5 shows the steps to generate the correlation tables based on an ATPG pattern. It can be seen that

by comparing the C0 value (0,1,1,1,0,0) with the C1 value (0,1,1,X,0,0), five of the logic values are the same. Hence, we record 5 in the correlation table ‘same’. Since there is no conflicting logic value, we record 0 in the correlation table ‘conflict’.

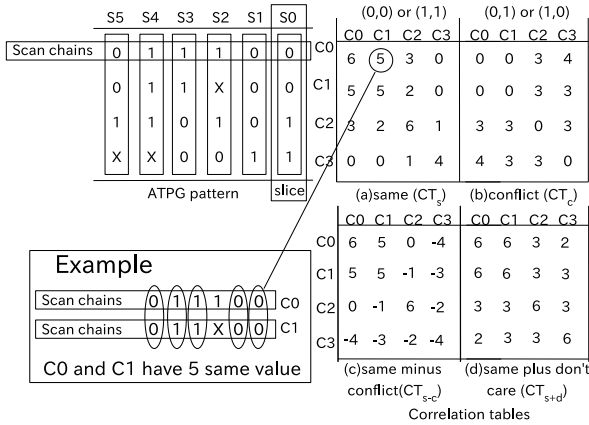


Figure 5: Correlation tables of ATPG pattern slices

3.2 Decision of the position of MUX and NOT gate

In our method, logic values for scan chains can be set to L_0 if a MUX is added to the scan chain. If there are many matching logic values in a pair of scan chains, the method is likely to reduce the number of BAST codes required to set the inverter code. Thus, the row with the largest values in the correlation table ‘same’ is used to determine the position of the MUX. If there are many conflicts in a pair of scan chains, then setting the same logic value for the pair will increase the number of BAST codes required for the inverter code. In this case, the method inserts a NOT gate in the corresponding MUX’s input. The method is described as follows. Since the correlation tables are symmetric, there are two or more rows which have the largest value. Among them, the row with the second largest value is selected. Depending on the values in the row, MUXes are inserted into the LFSR-reseeding circuit in descending order of the selected row as shown in Fig. 6(A). Let C_x be the selected scan chain. If the value of C_x in the correlation table ‘conflict’ is more than its value in the correlation table ‘same’, as shown in Fig. 6(B), then the method adds a NOT gate to the corresponding MUX’s input.

3.3 Selection of correlation table

If the values in the correlation tables are too small, it is difficult to determine the positions of the MUXes and NOT gates that are effective to reduce the number of BAST codes. Thus, the method decides which correlation table is to be used for inserting MUXes depending on the ratio of don’t care in ATPG patterns. If the ratio of don’t care is less than 80% and greater than 60%, the method utilizes the correlation table

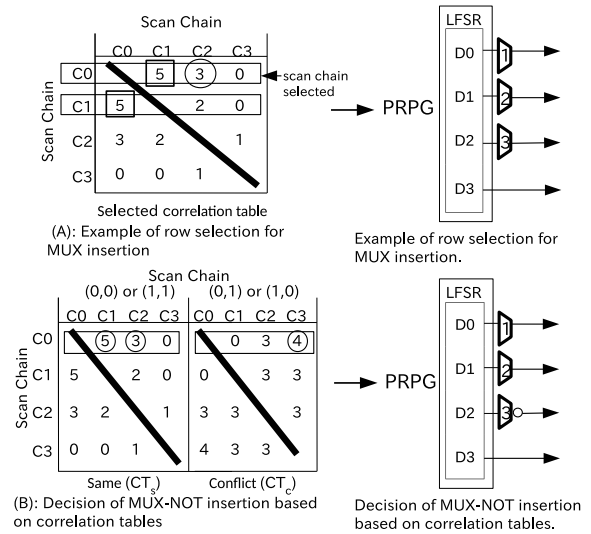


Figure 6: Decision of the positions of MUXes and NOT gates

‘same minus conflict’ to determine the positions of MUXes. If the ratio of don’t care is greater than 80%, the method selects the correlation table ‘same plus don’t care’ to determine the position of MUX. If the ratio of don’t care is less than 60%, the method selects the correlation table ‘same’ to determine the positions of MUXes and also adds a NOT gate to the corresponding MUX’s input as shown in Fig. 5.

4. Evaluation of the Proposed Method

In our experiments, a 16-bit LFSR is used with 16 scan chains and at most 13 MUXes are inserted.

Table 2 shows the experimental circuit data and the ratio of don’t care in the ATPG patterns. The columns *circuit*, *PI*, *FF*, and N_{len} denote the circuit name, the number of external inputs, the number of flip-flops, and the length of the scan chain, respectively.

Table 2: Circuit information and the ratio of don’t care

circuit	PI	FF	N_{len}	Ratio of don’t care
b14	32	245	18	75.6%
b15	36	449	31	87.5%
b17	37	141	91	88.2%
b20	32	490	33	71.3%
b21	32	490	33	71.8%
b22	32	735	486	73.7%
s5378	35	179	14	41.8%
s9234.1	36	211	16	52.2%
s13207.1	62	638	44	60.0%
s15850.1	77	534	39	73.0%
s35932	35	1728	111	48.0%
s38417	28	1636	104	38.2%
s38584.1	38	1426	92	79.6%

The test data required for the BAST codes, TD, are calculated by Eq. (1).

$$TD = (N_{vect} \times N_{len} + N_{inv}) \times (2 + \lceil \log_2 N_{ch} \rceil) \quad (1)$$

Table 3: Experimental results of proposed method

Circuit	Nvect	Ninv	TD	TDR[%]	Number of NOT gates	Selected correlation table
b14	437	7475	92046	15.96	Not used	CT_{s-c}
b15	439	4291	107400	20.71	Not used	CT_{s+d}
b17	452	13678	328860	23.80	Not used	CT_{s+d}
b20	424	15868	179160	23.84	Not used	CT_{s-c}
b21	417	15301	174372	23.51	Not used	CT_{s-c}
b22	440	21538	255948	25.49	Not used	CT_{s-c}
s5378	109	3941	32802	28.27	3	CT_s
s9234.1	143	4545	40998	28.68	1	CT_s
s13207.1	262	19655	187098	32.12	1	CT_s
s15850.1	138	4586	59808	36.06	Not used	CT_{s-c}
s35932	18	2177	25050	56.55	0	CT_s
s38417	106	32393	260502	31.22	1	CT_s
s38584.1	139	11042	142980	23.45	Not used	CT_{s-c}

CT_{s-c} = Correlation table 'same minus conflict'.

CT_{s+d} = Correlation table 'same plus don't care'.

CT_s = Correlation table 'same'.

where $(2 + \lceil \log_2 N_{ch} \rceil)$ is the number of bits required for one BAST code, and N_{ch} is the number of scan chains.

Table 3 shows the experimental results and the selected method for the 13 benchmark circuits. Columns N_{vect} , N_{inv} , TD , and TDR are the number of vectors, the number of inversion bits, the test data volume calculated by Eq. (1), and the reduction rate of the test data volume compared with that in [9], respectively. BAST requires a small volume of test data if there are many don't care bits in the test vectors. Moreover, the proposed method can markedly reduce the number of BAST codes required for inverter codes if the care bits in the test vectors are correlated with the scan chains and a large number of conflicts are found in the test vectors.

As a result, the proposed procedure can achieve a reduction of about 15 to 56% in the volume of test data for BAST. The reduction rate obtained by the proposed method varies among the circuits since the correlations among the test vectors differ among the circuits. The area overhead required for the proposed method is at most 13 MUXes and three NOT gates in this experiment, which is small compared with that of the original BAST circuit.

5. Conclusions

In this paper, an LFSR-reseeding circuit with additional MUXes and NOT gates inserted based on the correlation tables of an ATPG pattern is proposed. The volume of test data can be reduced by about 15-56% compared with the previous method, with a small area overhead. In future work, the reduction of the number of BAST codes by reordering ATPG patterns for the modified PRPG will be pursued to find better matching between PRPG and ATPG patterns.

Acknowledgment

The authors would like to thank Mr. M. Usui for his contribution in the experiments.

References

- [1] V. D. Agrawal, C. R. Kime and K. K. Saluja: A tutorial on built-in self-test: Part 1 Principles, IEEE Design and Test Magazine, Vol. 10, No. 1, pp. 73-82, 1993.
- [2] J. Rajski, N. Tamarapalli and J. Tyszer: Automated synthesis of phase shifters for built-in self-test applications, IEEE Trans. Computer-Aided-Design of Integrated Circuits and Systems, Vol. 19, No. 10, pp. 1175-1188, 2000.
- [3] T. Hiraide, K. O. Boateng, H. Konishi, K. Itaya, M. Emori, H. Yamanaka and T. Mochiyama: BIST-aided scan test – a new method for test cost reduction, Proc. IEEE VLSI Test Symp., pp. 359-364, 2003.
- [4] Y. Okada, H. Yotsuyanagi and M. Hashizume: The test vector compaction considering compatible flip-flops for BIST-aided scan test, Proc. 2012 RISP International Workshop on Nonlinear Circuits, Communications and Signal Processing, pp. 449-452, 2012.
- [5] C.-C. Fang, H. Yotsuyanagi and M. Hashizume: A test pattern matching method on BAST architecture for test data reduction by controlling scan shift, IEEE Workshop on RTL and High Level Testing, 2014.
- [6] M. Yamamoto, H. Yotsuyanagi and M. Hashizume: Scan chain configuration for BIST-aided scan test using compatible scan flip-flops, Workshop on RTL and High Level Testing, pp. 99-104, 2008.
- [7] T. Hosokawa, Y. Chen, L. Wan, M. Wakazono and M. Yoshimura: A test pattern matching method on BAST architecture using don't care identification for random pattern resistant faults, Proc. International Test Conference, pp. 738-743, 2010.
- [8] R. Mori, H. Yotsuyanagi and M. Hashizume: Test data reduction method for BIST-aided scan test by controlling scan shift and partial reset of inverter code, IEICE Tech. Rep., Vol. 113, No. 430, pp. 55 - 60, 2014 (in Japanese).
- [9] H. Yotsuyanagi, M. Yamamoto and M. Hashizume: Scan chain ordering to reduce test data for BIST-aided scan test using compatible scan flip-flops, IEICE Trans. Inf. Syst., Vol. E93-D, No. 1, pp. 10-16, 2010.