Generation of Low Power SSIC Sequences

Bei Cao[1(✉)] and Yongsheng Wang[2]

[1] Electronic Engineering School,
Heilongjiang University, Harbin 150080, China
caobei@hlju.edu.cn
[2] Microelectronic Center, Harbin Institute of Technology, Harbin 150000, China

Abstract. Single input change (SIC) sequence for VLSI testing has been researched because of effectiveness to more test fault models and low power consumption testing. It is the high fault coverage in deterministic built-in self-test (BIST) with low test cost and short test application time. The sequential single input change (SSIC) sequence used in deterministic BIST is presented in this paper for decreasing the dynamic power, reducing test application time and increasing fault coverage. The selection of seed vectors is the significant technique in deterministic BIST. The critical features of SSIC sequence are proposed for selecting seed vectors. The SSIC sequence generator is designed. The simulation results using benchmark circuits show that the SSIC sequences can increase fault coverage and decrease application time than random SIC sequences. SSIC sequence also has low dynamic power consumption.

Keywords: Single input change sequence · BIST · Low power testing
Fault coverage

1 Introduction

BIST technique have been widely improved and applied to reduce the cost and to solve testing problems of complex chips. In BIST, the test methods frequently used are exhaustive testing, pseudorandom testing and deterministic testing. A great number of test vectors are generated to assure high testing fault coverage in the exhaustive and pseudorandom schemes. The patterns based on fault model are generated by automatic test pattern generation (ATPG) software in the deterministic BIST. The high fault coverage and optimized application times make it a more attractive testing strategy.

Test pattern generator (TPG) is a very important hardware in BIST. Some main design objectives should be satisfied, such as improving fault coverage, reducing test time, decreasing power consumption, and so on. The single stuck-at fault is the classic fault model in VLSI testing technique. At present, more defecting types would appear in advanced CMOS technology. And effective fault models and testing methods should be considered to avoid chip failure. The power consumption during the chip testing could be higher than that in normal model [1]. Chips under testing may be damaged because of high test power dissipation. Researching and designing to generate the test sequences with low power consumption, high fault coverage and low hardware cost has become a significant topic to VLSI testing.

X. Gu et al. (Eds.): MLICOM 2017, Part II, LNICST 227, pp. 285–293, 2018.
https://doi.org/10.1007/978-3-319-73447-7_32

The SIC sequences are more effective than the multiple input change (MIC) sequences for some fault types, which is researched in depth and proved in the previous work [2, 3]. The SIC sequences can be used to meet the above testing objectives as an effective BIST scheme [2–7]. Research and application on SIC sequence has become an important topic [3–6]. The SIC sequence generator is significant design in BIST. Design criteria about the sequence generator is proposed by David, which must be satisfied to generate RSIC sequences [5]. SIC generator and seed generating algorithm were presented [6, 7]. Normally, RSIC TPG used in BIST mainly consists of two key parts, such as pseudorandom sequence source circuit and decoder circuit. Linear feedback shift register (LFSR) can generate pseudorandom sequences, which is used to control the variable bits in RSIC generator. RSIC sequences must have enough test length for achieving high fault coverage. How to select seed vectors is also a key technique [6].

SSIC sequence and its properties used in deterministic BIST are proposed to solve problems of RSIC in this paper. Applying the properties of SSIC sequences, a novel seed vector selection algorithm is proposed. The value in the input change bits is changed sequentially in the proposed SSIC sequence. The counter can be used as pseudorandom source circuit instead of the LFSR. The SSIC TPG is easier to implement with low hardware cost. The simulation results used benchmark circuits ISCAS'85 show that the proposed SSIC TPG and algorithm of seed vectors selection are effective than RSIC sequences. The SIC sequences can be used in low dynamic power testing because of low switching activity (SA).

2 Theory Research of SIC Sequence

2.1 Theories of SIC Sequence Generation

Normally, values in some bits are not the same between successive testing vectors, which are called as MIC sequence. Only one bit can be changed for successive test sequence pairs, named SIC test sequences. It is firstly defined as follows. Let S is a test sequence,

$$S = V(1)V(2)\ldots V(i)\ldots V(L) \tag{1}$$

n-bits and L successive test vector $V(i) = \{v_n\ v_{n-1}\ldots v_1\}$ constitute sequences S. For any $i > 1$, $V(i)$ is not the same as $V(i-1)$ in only one bit. The current change bit and the previous change bit are not repeated in the sequence loop. The principle of SIC generation is shown as Fig. 1.

LFSR based on the primitive polynomial can generate M-sequences, which is used commonly as pseudo-random pattern source. The code transition circuit is another important component. At any time t, $R(t) = r_1(t)\ r_2(t)\ldots r_m(t)$ as the m-bit binary code can be generated in LFSR. It is mapped to a decimal value $d(t)$, and $1 \leq d(t) \leq n$. The $R(t)$ is transformed to $d(t)$, which map the $d(t)$ bit in $V(t-1)$ as a changing bit. The value of $d(t)$ bit is transformed in the code transition circuit. The $V(t)$ of the SIC sequences can be obtained from SIC generator.

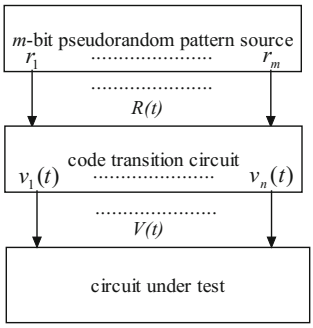

Fig. 1. Generation principle of SIC sequences.

The algebraic models of SIC sequences are shown as Eqs. (2)–(8). Initially, *R(t)* is generated as a binary value in pseudorandom source circuit LFSR. The input changing bit *d(t)* is a decimal value, which is converted from *R(t)* using the code transition circuit.

$$R(t) = r_1(t)r_2(t)\ldots r_m(t) \tag{2}$$

$$d(t) = \sum_{j=1}^{m} 2^{r_j(t)} \tag{3}$$

$$V(t) = v_n(t)v_{n-1}(t)\ldots v_i(t)\ldots v_1(t) \tag{4}$$

For example, if $d(t) = i$,

$$V(t+1) = v_n(t+1)\ldots v_i(t+1)\ldots v_1(t+1) = v_n(t)\ldots\overline{v_i(t)}\ldots v_1(t) \tag{5}$$

The above formula (2) and (5) are expressed as (6).

$$V(t+1) = V(t) \otimes 2^{d(t)-1} \tag{6}$$

The symbol "⊗" is exclusive or operation. Using the above expressions, The SIC sequences can be generated if *V(0)* is given as the seed vector. Obviously, the cycle is *2n* for a SIC sequence set based an *n*-bits seed vector. The length of LFSR *m* can be determined from seed vector bits *n*, which is shown as (7) and (8).

$$2^m \geq n \tag{7}$$

$$m \geq \log_2 n = \lceil \log_2 n \rceil \tag{8}$$

Here, the symbol $\lceil x \rceil$ is ceiling function. Ceiling operation for x means that arbitrary real number x is the smallest integer, and not less than x. The SIC sequences can be described and generated using algebraic model (2)–(8).

The SA in the circuit under test can be reduced effectively using SIC sequences because only one value is changed between successive testing vectors. An experiment about the SA used benchmark circuits ISCAS'85 was performed. It is the comparison of power consumption between MIC sequences and SIC sequences. We obtained the MIC sequences using Atlantic [8]. The SA means the dynamic power consumption. The experimental results are shown as Fig. 2.

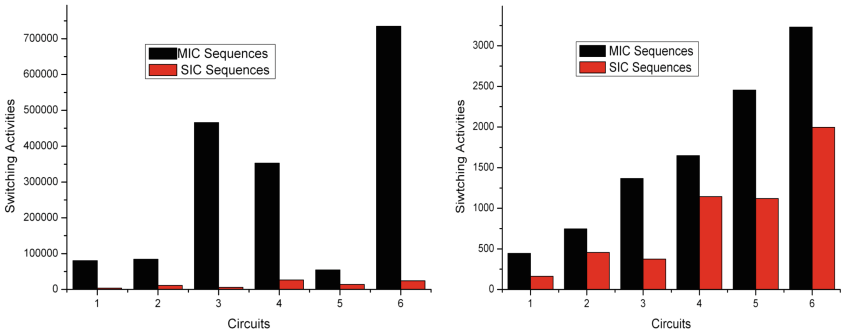

Fig. 2. Comparison of power total SA (a) and peak SA (b) for SIC and MIC sequences.

The SIC sequences are obtained according to the random seed vectors in the experiment. The amounts of SIC and MIC sequences are the same. The simulation results demonstrate that the SIC sequences can significantly reduce SA relative to the MIC sequences. It means the dynamic power consumption is decreased during test using SIC sequences. The X-axis refers to the experimental circuit. The Y-axis indicates the numbers of SA.

2.2 Research of SSIC Generation

The changing bit of SIC sequence is controlled by a pseudo-random source circuit. LFSR based the primitive polynomial is commonly used. The SIC test sequences are applied to deterministic BIST in this paper. The changing bits control is better suited to a counter instead. Here, the sequential single input change (SSIC) sequence is proposed. Suppose $V(0) = \{v_1\ v_2 \ldots v_i \ldots v_n\}$ to be a SSIC vector, n represents the number of bits, where $v_i \in \{0, 1\}$. Each vector bit is inverted sequentially in the given n-bit test vector according to the clock. The SSIC vector is produced sequentially until the original seed vector appears. The definition and instance of SSIC sequence are shown in Fig. 3. The value in the first bit is active. Then the value of the second bit is changed at next clock, in turn. Obviously, only one bit is mapped and changed between the neighboring clocks. The $2n$ SSIC sequences are generated by a seed vector, which are referred to as segment, such as $S = \{v(1) \ldots v(n), v(n+1), \ldots v(2n)\}$. The clock cycle of each SSIC segment is $2n$, also be seen as a testing subset. The seed vectors are obtained using the proposed algorithm.

The size of segment will be $2n \times n$ if it is generated by an n-bit seed vector. SSIC segment can be partitioned into eight sub-segments in Fig. 3(b), which are signed respectively. S_i' and S_i'' are named neighboring sub-segment. A neighboring sub-segment

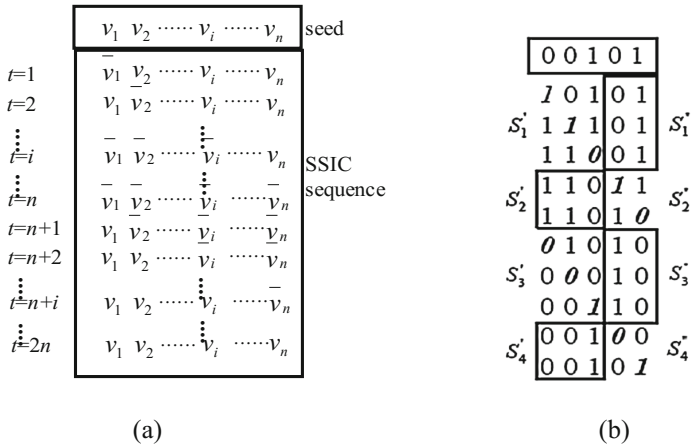

Fig. 3. Definition of the SSIC test sequence (a) and an instance of segment (b)

is considered as a subset of SSIC segment. The size of each sub-segment is (n/2) × (n/2) if *n* is even. The size of sub-segment has several cases while n is odd. Concretely, the size of sub-segment are *[(n + 1)/2) × (n + 1)/2]* for S_1' and S_3', *[(n − 1)/2) × (n + 1)/2]* for S_2' and S_4', *[(n + 1)/2) × (n − 1)/2]* for S_1'' and S_3'', *[(n − 1)/2) × (n−1)/2]* for S_2'' and S_4'', respectively. The parts are called head-segment if they include S_i'. Similarly, the parts including S_i'' are tail-segment. Both of head-segment and tail-segment are n in all. It is defined as a deterministic sub-segment if it has the same sub-vector in a sub-segment. Otherwise, it is named as transformable sub-segment. The deterministic sub-segments are S_1'', S_2', S_3'', S_4' in Fig. 3(b), respectively. The other sub-segments are transformable. Some properties of SSIC sequence are described as follows.

Property 1: SSIC segment *S = {v(1)... v(n), v(n + 1),..... v(2n)}* generated by an *n*-bits seed vector, if *v(i)* and *v(j)* are complementary, here $i, j \in \{1, \ldots n, \ldots 2n\}$, *i* is not equal to *j*, then $|i - j| = n$.

Property 2: Deterministic sub-segment S_1'' and S_3'', S_2'' and S_4' are complementary, respectively.

Property 3: Each vector can be considered two sub-vectors which belong to neighboring sub-segment. One sub-vector is in the deterministic sub-segment, and the other is in the transformable sub-segment.

Property 4: There is a changing bit between successive vectors in deterministic sub-segment or transformable sub-segment.

Property 5: The difference between successive vectors for SSIC sequence can be expressed. $|v(i) - v(i-1)| = 2^{n-i}$ if $i \leq n$, and $|v(i) - v(i-1)| = 2^{2n-i}$ if $n < i \leq 2n$.

The SSIC seed vectors can be obtained from ATPG deterministic test patterns according to seed selection algorithm based on the above definitions and properties. Corresponding SSIC generator can be designed easily.

3 Seed Selection Algorithm and SSIC Generator Design

The previous definitions and properties can be used in seed vectors selection based on correlative characteristic of the test sequence. The testing vectors are correlative if they have appeared in a SSIC segment. They can be defined isolated vectors if the vector is not correlative with any ones in ATPG deterministic test set. A weight value is defined based on correlative characteristic of vectors, which is used to select and optimize the seed vectors.

The proportion of don't care bits "X" in ATPG deterministic test sets are high. In most cases, the don't care bits may be able to exceed 90% [9]. How to map don't care bits is key technique instead of random filling. The number of seed vectors can be optimized if mapping in terms of primary vector is used. The each vector is regarded as a primary vector one by one. "X" of other vectors is substituted by the corresponding value according to the primary vector. The weight based correlative characteristic is calculated between the primary vectors and other ones after "X" filling. An anticipant weight value matrix of two-dimension is built to indicate the correlative characteristic among testing vectors.

The key to seed vectors selection algorithm is to build the anticipant weight value matrix. The SSIC properties are the principle of correlative characteristic in this paper. The anticipant weight value "$ANT_{i,j}$" is defined according to vector i and j:

$$ANT_{i,j} = \lambda * (c_i + c_j) \tag{9}$$

Here, λ is a precedence parameter of correlative characteristic. It has precedence when Property 1 is satisfied for both vector i and vector j. It is $\lambda > 1$, and $\lambda = 1$ for other correlative property. c_i and c_j denote that the numbers of vectors is same for their sub-segment. The more original vectors may be included into an SSIC segment if the high anticipant weight vector is selected as a seed vector. It is good to optimize the seed vectors. Algorithm of anticipant weight value matrix is shown as follows.

```
Algorithm 1: Anticipant Weight Value ANT()
   Initialize the parameter λ;
   Partition the front sub-segment and the tail sub-
   segment in terms of n;
   for (i=1 to m) //m is the number of ATPG sequences;
       Calculating c_i according to mapping principle of
       primary vector;
   end for
   for (i=1 to m-1)
      for (i=i+1 to m)
      Judging the correlative characteristic;
      Computing anticipant weight about i and j;
      end for
    end for
    Analysis and return solutions;
```

The pseudo-code of the SSIC seed vectors selection is depicted in Algorithm 2.

```
Algorithm 2: Seed Vectors Selection
    Loading the deterministic test sequence set;
    Initializing the sign variable flag;
    Partition the front sub-segment and the tail sub-
   segment according to n;
    Calling function ANT();
    while flag=1
    The vector i and j are selected to constitute the
   SSIC segment if ANT_{i,j} is high;
    The seed vector is determined according the current
   SSIC segment;
    Deleting the vectors included in the current SSIC
   segment from test sets;
    Mapping don't care bits in SSIC segment according to
   corresponding vectors;
    Revising signs of anticipant weight value matrix and
   original test sequences;
    Judging the variable flag;
       If flag=1; vectors are not treated; return and
   continue to call ANT();
       Else continue to the next step;
       If there are isolated vectors, return and
   continue to call ISO_V();
       Else end if;
    end while
    Calling function ISO_V( ); //optimize seed vector
   for the isolated vectors;
    Statistic the seed vectors:
    Print the solutions;
```

There is no direct correlation characteristic between the isolated vectors and other vectors. And it may be treated as seed vector. Vector will be first selected as an initial vector if the proportion of don't care bits 'X' is low in this paper.

The hardware structure to generate SSIC sequences is similar to SIC generator. Two pseudorandom source circuits and code transition circuit are designed. However

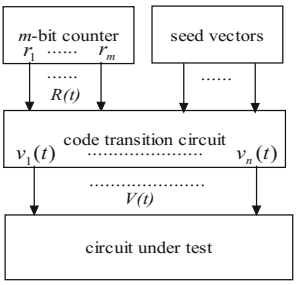

Fig. 4. Generator of SSIC sequences

an m-bit counter is used to generate changing bits. The counter is more suitable for generating SSIC than LFSR. The bits *m* of the counter can be computed according to *n*-bit value. It is shown as function (8) (Fig. 4).

4 Experimental Results and Conclusions

The simulations were performed to verify the proposed SSIC seed vector selection algorithm using benchmark circuits ISCAS'85. The algorithm is designed based on MATLAB. The deterministic test sequences are generated from the Atlantic ATPG [8].

The simulation results are obtained to generate the proposed SSIC sequences in Table 1. The number of input ports and seed vectors are shown in the second and third column respectively. The SA of SSIC sequences are given to analyze the dynamic power consumption. The total power and peak power based the SA are shown in the fourth and fifth columns respectively.

Table 1. Simulation results of SSIC test sequences generation

Benchmark circuits	Input (n)	Number of seeds	Switching activity for SSIC sequences	
			Total SA	Peak SA
c432	36	16	6750	89
c499	41	55	900	8
c1355	60	107	26218	47
c6288	41	49	580987	2492
c1908	33	136	469537	742
c3540	233	153	813214	1415
c5315	50	158	1795938	1763
c880	178	68	88478	301
c2670	32	119	631794	958
c7552	207	249	3643206	3063

The proposed scheme compared to the RSIC generation technique, which is 10-bit RSIC seed selection circuits in the Table 2 [6]. The test length represents the number of SIC vectors. These vectors are generated gradually by the seed vectors. Seed vectors play an important role in determining the length of testing. SFC means the single stuck-at fault coverage. The proposed SSIC sequences generation scheme can obtain high fault coverage. It is the advantage of deterministic BIST. The test length can reduce effectively for fewer input port numbers.

Low power testing in BIST is now becoming focus of both academic and industry communities. The novel SSIC test sequence is proposed in this paper. The algorithm of seed vector selection is presented using SSIC the correlative characteristic. As a result of SA, 80% of the total power consumption in the CMOS circuit is caused by SA. The SSIC test sequences can reduce effectively the SA and decrease test application time. Test costs have also been effectively reduced based on the deterministic BIST.

Table 2. Experimental results comparison

Benchmark circuits	Input (n)	Proposed		Reference [6]	
		Test length	SFC (%)	Test length	SFC (%)
c432	36	1152	99.24	9216	98.70
c499	41	4510	98.95	5248	99.50
c880	60	8160	100.00	15360	92.03
c1355	41	8774	99.49	10496	97.56
c1908	33	8976	99.52	8448	97.15
c2670	233	55454	95.74	28224	75.68
c3540	50	15300	96.00	12800	87.20
c5315	178	40448	98.90	23784	88.55
c6288	32	3136	99.56	8192	98.08
c7552	207	103086	98.27	52992	80.68

References

1. Zorian, Y.: A distributed BIST control scheme for complex VLSI devices. In: Proceedings of 11th IEEE VLSI Test Symposium, Los Alamitos, California, pp. 4–9 (1993)
2. Virazel, A., David, R., Girard, P., Landrault, C., Pravossoudovitch, S.: Delay fault testing: choosing between random SIC and random MIC test sequences. J. Electron. Test.: Theory App. **17**, 233–241 (2001)
3. Li, X., Cheung, Y.S.: High-level BIST synthesis for delay testing. In: International Symposium Defect and Fault Tolerance in VLSI Systems, pp. 2–4. November 1998
4. David, R., Girard, P., Landrault, C., Pravossoudovitch, S., Virazel, A.: On using efficient test sequence for BIST. In: Proceedings of the 20th IEEE VLSI Test Symposium, pp. 145–150 (2002)
5. David, R., Girard, P., Landrault, C., Pravossoudovitch, S., Virazel, A.: On hardware generation of random single input change test sequences. In: Proceedings of European Test Workshop, pp. 117–123 (2001)
6. Lei, S.C., Hou, X.Y., Shao, Z.B., Liang, F.: A class of SIC circuits: theory and application in BIST design. IEEE Trans. Circuits Syst.-II **55**, 161–165 (2008)
7. Voyiatzis, I., Haniotakis, T., Halatsis, C.: Algorithm for the generation of SIC pairs and its implementation in a BIST environment. IEE-Proc.-Circuits Devices Syst. **153**, 427–432 (2006)
8. Lee, H.K., Ha, D.S.: On the generation of test patterns for combinational circuits. Technical report, Department of Electrical Engineering, Virginia Polytechnic Institute and State University, pp. 12–93
9. Lee, J., Touba, N.A.: LFSR reseeding scheme achieving low power dissipation during test. IEEE Trans. Comput.-Aided Des. **26**, 396–401 (2007)