Broadcast Scan Compression Based on Deterministic Pattern Generation Algorithm

Hyeonchan Lim¹, Sungyoul Seo¹, Soyeon Kang¹ and Sungho Kang¹
¹Departments of Electrical & Electronic Engineering Yonsei University, Seoul, Korea
E-mail: lhcy92@yonsei.ac.kr

Abstract
As advances in technology make integrating more transistors on a single integrated circuit (IC) feasible, test data volume becomes one of major factors of testing system-on-chips (SoCs). The large volume of test data leads to increasing test application time and needs more expensive Automatic test equipment (ATE) with high memory. In this paper, we present broadcast scan compression based on deterministic pattern generation algorithm to reduce the volume of test data. The proposed method further improves compression ratio of the volume of test data by exploiting advances of both broadcast scan compression and pattern generation using linear feedback shift register (LFSR). The volume of test data can be reduced by feeding multiple scan chains from a few LFSRs and by compressing the data using LFSRs. ISCAS’89 benchmark circuits verify the proposed method and the experimental results show that the compression ratio is up to 10X which means test application time also is reduced extremely.

Keywords
Test data compression, Scan-based test, broadcast scan compression, compatibility, LFSR

1. Introduction
As the complexity of VLSI circuits continues to increase, test cost is becoming an important factor in manufacturing the complex integrated circuit. While a test based on scan architecture is the most widely used solution in order to achieve high fault coverage, the test doesn’t consider the problems of large volume of test data which is recognized as one of major contributors to the test cost. Automatic test equipment (ATE) with high memory which can deal with increasing volume of test data is expensive. Thus, compression of test data has been on the rise in the situation of the increasing cost of IC manufacturing by reducing memory requirements on the ATE [1]. Many compression schemes have been proposed to reduce the volume of test data needed to be stored in ATE by exploiting the don’t care bits [2, 3, 4, 5]. Although a decompressor circuit should be designed to decompress the compressed data into original test patterns, test data compression can reduce the test data volume and test application time without degrading performance of the circuit. Typically, the existing test data compression schemes can be classified in to three types [2]: code-based schemes, linear-decompression-based schemes, and broadcast scan-based schemes. Code-based schemes encode the test patterns into code words by partitioning the original test patterns into symbols and replacing each symbol with a code word. Decompressor simply works by converting each code word into the corresponding symbol [6]. Linear-decompression-based schemes encode the test patterns by solving the system of linear equations. Generally, these schemes utilize a linear decompressor which is based on combinational linear XOR networks [3] or sequential linear logic such as linear feedback shift registers (LFSRs) [4, 5]. Broadcast scan-based schemes distribute the same value to multiple scan chains using a single test channel. These schemes are actually a special case of linear decompression-based schemes where the decompressor consists of only fan-out wires. Illinois scan for independent scan chains and scan forest for broadcasting the outputs of scan chain are included in these schemes [7, 8]. Since linear decompressors and combinational broadcasting logic can be simply implemented, linear decompression-based schemes and broadcast scan-based schemes needs lower hardware overhead and with commercial automatic test pattern generator (ATPG), these two schemes can obtain higher compression ratio. As the complexity of VLSI circuits increases more and more, however, the compression ratio of previously proposed compression schemes isn’t enough to relieve the memory burden of the ATE. In order to solve this problem, a new broadcast scan compression based on deterministic generation algorithm is proposed. It utilizes the properties of broadcast scan compression which feeds multiple scan chains with the same values and the LFSR which can compress the test data into seeds whose volume is much smaller than the volume of the original test data.

2. Background and motivation

2.1. Linear feedback shift register
A LFSR is generally used to generate test patterns. Figure 1 shows a conventional LFSR in multiple scan chains architecture. Scan chains are connected to the LFSR (or phase shifter) and the test patterns are fed into the scan chains in parallel. Using primitive characteristic polynomial, an LFSR generates all of states, except all 0s. A n-degree LFSR which uses primitive polynomial generates 2n−1 states. The LFSR architecture easily generates pseudo random patterns. However, Since the LFSR feeds not a scan chain but a scan slice and scan chains which are adjacent each other have just 1-bit shifted values, it is complex for the LFSR to generate certain deterministic patterns.

2.2. Serial LFSR
Figure 2 shows a serial LFSR in multiple scan chains architecture. Unlike the conventional LFSR architecture, scan chains are connected to the last register of the LFSR, which means that every scan chain is fed with the same values (or inverted one). In this architecture, it is easy to
predict which pattern will be generated by solving characteristic polynomials of the LFSR. However, since scan chains having the same values may degrade fault coverage, it is important to combine similar scan chains and broadcast properly.

2.3. Deterministic pattern generation

The computation algorithm of characteristic polynomial of LFSR for generating deterministic patterns is proposed in [9]. LFSR synthesized by the algorithm can generate all deterministic patterns with a single polynomial and corresponding seeds. The maximum degree of computed characteristic polynomial is determined by $S_{\text{max}}$ which is the maximum number of specified bits in the test patterns [9]. Mostly, $S_{\text{max}}$ is under half of the length of a scan chain. With a polynomial of degree $L$, first $L$ bits of the test patterns become a seed. Although the LFSR can generate all deterministic test patterns, the algorithm is restricted to a single scan chain architecture.

3. Proposed method

Although the algorithm proposed in [9] can synthesize LFSR which can generate all deterministic patterns, it is restricted to a single scan chain architecture. To expand into multiple scan chains architecture, multiple LFSR may become one of solutions, however, it increases hardware area overhead besides the volume of test data. To solve this problem, we introduce the concept of broadcast-based scan compression. At first, we’ll find compatible scan chains and combine them into a set. In turn, using the small number of serial LFSRs synthesized by the algorithm in [9], deterministic patterns are broadcasted into the sets of scan chains. Detailed of the proposed method is following.

3.1. Finding compatible scan sets

In serial LFSR architecture, scan chains should be combined properly in order to retain fault coverage in desirable level. A compatible scan set is a set of compatible scan chains which will be broadcasted from the same LFSR. Two bits are compatible if they have the same value or at least one of those is don’t care bit. Two test patterns are compatible if all bits in the patterns are compatible element-by-element. Two scan chains may be considered compatible if all patterns in the scan chains are compatible in sequence. In spite of many don’t card bits existing in the test patterns, however, it is almost impossible that all patterns in the scan chains are compatible. Thus, compatible-bound $CB$, lower bound of ratio of compatible patterns in two scan chains, is introduced to determine whether the scan chains are compatible. Two scan chains are considered compatible if $CB*100$ percent of test patterns in the scan chains are compatible respectively and then, the two scan chains are combined into a compatible scan set. $CB$ should be set properly, since low $CB$ makes two scan chains having different property broadcasted together, which means degrade of fault coverage. Table I shows an example of a compatible scan set with $CB = 0.6$. Column 1 shows the test pattern number. Column 2 and 3 show test pattern in each scan chain. Column 4 shows a compatible scan set and its transformed patterns. Chain 1 and Chain 2 have 4 compatible patterns of total 6 patterns. 4 over 6 is larger than $CB=0.6$. Thus, Chain 1 can be considered compatible with Chain 2. In turn, they are combined into a compatible scan set. For patterns in a compatible scan set, if they are compatible with each other, they are transformed into compatible one by filling don’t care bits properly but if they are not compatible, specified bits having mismatch are arbitrarily selected. Detailed of finding compatible scan sets is explained in procedure 1.

Line 1 initializes $k=0$. In line 3~7, new compatible scan set which takes arbitrary scan chain $c$ and its test patterns from $C$ is created. Line 8~9 determines the compatibility between the compatible scan set and all scan chain which isn’t included in any compatible scan set. Subroutine ‘Compatible_Ratio’ compares all test patterns in sequence exhaustively and returns their compatible ratio. In line 10~12, if they are compatible, the compatible scan set will take the scan chain and subroutine ‘Trans_Pattern’ will calculate corresponding compatible test patterns and return them. After finding a compatible scan set completely, next compatible scan set will be found by repeating processes above until $C$ become empty.
Table 1: An example of a compatible scan set with CB=0.6

<table>
<thead>
<tr>
<th>#</th>
<th>Chain 1</th>
<th>Chain 2</th>
<th>Compatible Scan Set</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>XX1X1X</td>
<td>1XX0XX</td>
<td>Compatible : 1X101X</td>
</tr>
<tr>
<td>2</td>
<td>10XX0X</td>
<td>X1XX00</td>
<td>Not : 10XX00</td>
</tr>
<tr>
<td>3</td>
<td>XXX10X</td>
<td>1X0X0X</td>
<td>Compatible : 1X010X</td>
</tr>
<tr>
<td>4</td>
<td>01XX0X</td>
<td>0XX1XX</td>
<td>Compatible : 01X1X0</td>
</tr>
<tr>
<td>5</td>
<td>0XXX1X</td>
<td>XXXX11</td>
<td>Compatible : 0XXX11</td>
</tr>
<tr>
<td>6</td>
<td>XX101X</td>
<td>1X1X0X</td>
<td>Not : 1X101X</td>
</tr>
</tbody>
</table>

Procedure 1. Finding_Compatible_Scan_Sets(C, CB)

**Input:** Test pattern set $T$, set of scan chains $C$, compatible-bound $CB$

**Output:** Set of compatible scan sets $S$, transformed compatible test pattern set $TS$, $|S| = |TS| = k$

1: $k = 0$
2: while $C \neq \phi$ do
3: choose arbitrary $c$ in $C$
4: $k = k + 1$
5: $S[k] = \{c\}$
6: $TS[k] = T[c]$
7: $C = C - c$
8: for $\forall c$ in $C$
9: if $Compatible_Ratio(TS[k], T[c]) \geq CB$ then
10: $S[k] = S[k] \cup c$
11: $TS[k] = Trans_Pattern(TS[k], T[c])$
12: $C = C - c$
13: end if
14: end for
15: end while
16: return $S$, $TS$, $C$

4. Overall scheme

Figure 3 shows overall test scheme of the proposed method. Compatibility of all scan chains is checked step-by-step, until every scan chain is included in compatible scan set. The number of compatible scan sets should be smaller than the number of scan chains and be restricted to use the concept of broadcast. K compatible scan sets means that K serial LFSRs are needed. Scan chains in a compatible scan set are fed from the last stage of the same LFSR. If there are scan chains which are counter-compatible like Scan chain 4, they can be fed with inverted value of the last stage of the LFSR. Assumed that the length of LFSR is L and the length of scan chain is $n$ where $L \ll n$, first $L$ clocks initialize the state of LFSRs by feeding seeds. During next $n - L$ clocks, LFSRs generate deterministic patterns using seeds and broadcast them into the compatible scan sets, respectively. At this time, the remain L bits of generated test patterns are stored in LFSR. Thus, during next L clocks, ATE feeds the seeds into the LFSRs and at the same time, remaining L bits are shifted into the scan chains.

3.2. Synthesizing serial LFSR

To solve the problem that the algorithm in [9] is restricted to single scan chain architecture, we proposed the concept of compatible scan set and will apply the algorithm in [9] on the compatible scan set. It is important to determine proper $CB$, since it is closed related with fault coverage. The entire procedure is outlined in Procedure 2.

Line 1–2 initializes minimum and maximum value of $CB$. $CB$ is determined like line 4. Line 5 calls a subroutine ‘Finding_Compatible_Scan_Sets’ described in procedure 1, which returns compatible scan sets and corresponding test patterns. Line 5 calls a subroutine ‘Char_Polynomial’ which is proposed in [9]. It synthesizes a LFSR which can generate test pattern set $TS$ and returns characteristic polynomial of the LFSR. In line 7, a subroutine ‘Fault_Sim’ processes fault simulation and determine the fault coverage. Since low $CB$ degrades fault coverage and high $CB$ leads to excessive hardware area overhead due to LFSR, $CB_{max}$ and $CB_{min}$ are changed until we get desirable $CB$ value by iteration in line 3, 8 and 10. Finally, line 13 returns compatible scan sets and characteristic polynomials of LFSR which broadcasts each compatible scan set.

Procedure 2. Synthesize_Serial_LFSR(C, CB)

**Input:** Test pattern set $T$, set of scan chains $C$, desirable fault coverage $FC$

**Output:** Set of compatible scan sets $S$, transformed compatible test pattern set $TS$, characteristic polynomial set $P$, $|S| = |TS| = |P| = k$

1: $CB_{min} = 0.5$
2: $CB_{max} = 1$
3: while $CB_{max} - CB_{min} > 0.5$ do
4: $CB = (CB_{max} + CB_{min}) / 2$
5: $[S, TS, k] = Finding_Compatible_Scan_Sets(C, CB)$
6: $P = Char_Polynomial(TS, k)$
7: if Fault_Sim($TS, P$) $\geq FC$ then
8: $CB_{max} = CB_{max} - 0.05$
9: else
10: $CB_{min} = CB_{min} + 0.05$
11: end if
12: end while
13: return $S$, $TS$, $P$, $k$
5. Experimental results

Commercial ATPG tool-TETRAMAX was used to generate deterministic test set. Table II shows the experimental results on ISCAS’89 benchmark circuits [11]. Column Circuit shows name of the circuit. Column Test set shows size of whole test set. Column #S.C. shows the number of scan chains used in the proposed method. [9] is restricted in single scan chain architecture. Column Deg. shows corresponding LFSRs’ degree. Column F.C. shows fault coverage. Finally, Column C.R. shows the comparison of compression ratio between [9] and the proposed method. For S38584 circuit, [9] shows average compression ratio of 6.30X. Using the proposed test scheme, 12 scan chains can be merged into 2 compatible scan sets. By this step, compression ratio can reach 6X because 12 scan chains are broadcasted by 2 inputs. For further reduction, LFSR whose length is much shorter than the length of a scan chain is introduced. The test cubes of 1464 bits are compressed into the seeds of 134 bits. Thus, the compression ratio is increased up to 10.93X. In the same way, the compression ratio of the other circuits is computed. A loss of the fault coverage comes from the number of test patterns. [9] unmerges test pattern from ATPG to exploits lots of X-bits, making generation of the test patterns easier. In the proposed method, 731 patterns are used for S38417 circuit but [9] uses 1449 patterns for the same circuit.

6. Conclusion

In this paper, a new broadcast scan compression based on deterministic pattern generation using serial LFSR is proposed. It exploits the advances of both broadcast scan-based compression and deterministic pattern generation using serial LFSR. Using the proposed method, the volume of test data can be reduced dramatically. Since this method utilizes LFSR with short length compared to scan chain length and doesn’t need to modify circuit under test (CUT), it is expected that the hardware overhead and router overhead is similar to the traditional broadcast scan-based compression architecture. This method utilizes already proposed algorithm in [9] for deterministic pattern generation, however, any other algorithm can be applied.

7. References

