Deterministic built-in self-test using split linear feedback shift register reseeding for low-power testing

M.-H. Yang, Y. Kim, Y. Park, D. Lee and S. Kang

Abstract: A new low-power testing methodology to reduce the excessive power dissipation associated with scan-based designs in the deterministic test pattern generated by linear feedback shift registers (LFSRs) in built-in self-test is proposed. This new method utilises two split LFSRs to reduce the amount of the switching activity. The original test cubes are partitioned into zero-set and one-set cubes according to specified bits in the test cubes, and the split LFSR generates a zero-set or one-set cube in the given test cube. In cases where the current scan shifting value is a do not care bit accounting for the output values of the LFSRs, the last value shifted into the scan chain is repeatedly shifted into the scan chain and no transition is produced. Experimental results for the largest ISCAS'89 benchmark circuits show that the proposed scheme can reduce the switching activity by 50% with little hardware overhead compared with previous schemes.

1 Introduction

Highly developed deep sub-micron technology has enabled the implementation of a large system-on-a-chip (SoC). The large SoC design includes several intellectual property (IP) cores such as processor cores, embedded memories and other peripheral cores. Traditional test methods using the automatic test equipment (ATE) have become unsuitable for testing of these large SoCs. This is because the more IP cores are used in one SoC, the larger test data volumes are required. Therefore in order to apply this large volume of test patterns to the SoC, the ATE requires large memory and this increases the test cost of the SoC. Also, since the number of external I/O pins of the SoC and the number of ATE channels are limited, the SoC testing is very time consuming. Built-in self-test (BIST) is widely known as a good solution for testing the individual IP cores [1–6]. As a test pattern generator of BIST, an LFSR (linear feedback shift register) is widely adopted to generate a pseudo-random test pattern. However, in cases that produce many random pattern resistant (RPR) faults in the circuit under test, it is very difficult to get high fault coverage with a pseudo-random test pattern.

Several attractive solutions for this problem have been proposed [7–21]. The weighted random test was presented to reduce the test set size and to guarantee high fault coverage [7–9]. The weighted random test adds the additional hardware necessary to change the probability of the logic value 1 or 0 occurring at each primary input. In [10–12], the deterministic BIST schemes using 'bit-flipping' or 'bit-fixing' were proposed. In these schemes, fixing or flipping some bits in a pseudo-random sequence generates the deterministic test patterns.

doi:10.1049/iet-cdt:20060114

Paper first received 29th July 2006 and in revised form 30th January 2007

The authors are with the Department of Electrical and Electronic Engineering, Yonsei University 134, ShinChon-Dong, SeoDaeMun-Gu, Seoul, South Korea

E-mail: mhyang@soc.yonsei.ac.kr

Another solution is the LFSR reseeding technique. The LFSR reseeding methodology was proposed in [13–21]. BIST loads a seed into an LFSR and the LFSR generates a deterministic test pattern and fills a scan chain set with the pattern. A seed can be computed for each test cube by solving a system of linear equations based on the feedback polynomial of the LFSR. It was determined in [13] that in order to keep the probability of not finding a solution for the system of linear equations less than 10^{-6}, the LFSR degree should be larger than $S_{max} + 20$, where S_{max} is the largest number of specified bits in any test cube in the test set. In order to enhance the LFSR reseeding technique, many methods have been proposed [14–21]. In [14, 15], a reseeding scheme using multiple polynomial linear feedback shift registers (MP-LFSR) was introduced. In this method, the LFSR degree should be larger than S_{max}, to keep the probability of not finding a solution less than 10^{-6}. In [16, 17], a variable-length reseeding methodology was proposed to optimise the memory allocation for seeds. In [18, 19], the reseeding approach was modified into a partial reseeding scheme using smaller size seeds than the LFSR degree. The study in [20, 21] proposed an LFSR reseeding scheme with variable-length multiple polynomial linear feedback shift registers (VLMP-LFSR). In this method, test cubes with a larger number of specified bits are encoded with LFSR polynomials of higher degree whereas test cubes with a smaller number of specified bits are encoded with LFSR polynomial of lower degree. While the LFSR reseeding scheme is a good solution for RPR faults, it causes excessive power dissipation. In the LFSR reseeding scheme, the don't care bits in the test cubes are filled with pseudo-random bits generated by the LFSR and unnecessary switching activity is produced.

Several techniques for reducing switching activity in deterministic pattern generation have been developed [22–24]. A low power scheme using dual LFSR reseeding was proposed in [22]. The main LFSR generates the test cube using conventional LFSR reseeding, and an additional LFSR generates the mask pattern. The outputs of the two LFSRs are ORed or ANDed. Therefore the transition

probability is reduced, but the test data storage and the hardware overhead to generate the mask pattern are greatly increased. In [23], a low power scheme uses hold cubes. In this scheme, each test cube is divided into several blocks and each block has a hold flag. If the hold flag for a block is 1, then the scan input data in the block are simply held constant from the last data bit in the previous block. However, additional test storage for the hold cubes and additional hardware for the hold flag shift registers are required. In [24], a low power scheme based on scan slice overlapping has been introduced. In this scheme, the pattern is partitioned into several overlapping slice sets, and no transition is produced in the overlapping block. However, the variance in the reduction of the switching activity is high, because the number of overlapping blocks is different according to the circuits and test patterns.

This paper proposes a new split LFSR reseeding methodology to reduce the number of transitions in the scan chain by 50%. The proposed scheme uses two LFSRs to generate low power deterministic test patterns. One LFSR generates a zero-set cube of a given test cube, and the other LFSR generates a one-set cube of a given test cube. Bit 1s generated by the LFSR for zero-set cube and bit 0s generated by the LFSR for one-set cube are don't care bits. Therefore the number of transitions can be reduced by filling the don't care bits with the last value shifted into the scan chain. The experimental results for the largest ISCAS'89 benchmark circuits show that the proposed scheme can reduce the switching activity by about 50% with little hardware overhead compared to previous schemes.

2 Low power testing methodology

In the conventional LFSR reseeding methodology, the corresponding seeds can be computed for each test cube by solving a system of linear equations based on the feedback polynomial of the LFSR. Therefore the LFSR can generate all specified bits 0 and 1 in the test cube by using the computed seeds. In this procedure, don't care bits are filled with a pseudo-random pattern generated by the LFSR. These bits filled by the LFSR cause unnecessary transitions in the scan chain. Therefore the don't care bits must be filled with the proper values in order to minimise the number of transitions in the test pattern.

Given a logic signal S, the signal probabilities $P_0(S)$ and $P_1(S)$ represent the average fractions of clock cycles in which signal S is 0 or 1. Assuming S is a binary random signal generated by an LFSR, $P_0(S) + P_1(S) = 1$ and $P_0(S) = P_1(S) = 0.5$. The transition probability of signal S, $P_{tr}(S)$, represents the average fraction of clock cycles when the current value of S is different from its previous value. $P_{tr}(S)$ can be computed as $P_{tr}(S) = P_0(S) \times P_1(S) + P_1(S) \times P_0(S) = 0.5 \times 0.5 + 0.5 \times 0.5 = 0.5$ [22]. In order to reduce the number of transitions in the test pattern, the transition probability should be lowered. AND and OR compositions of two mutually independent random signals can be used to lower the transition probability. In AND and OR compositions, $P_1(S_{AND}) = P_0(S_{OR}) = 0.25$ and $P_0(S_{AND}) = P_1(S_{OR}) = 0.75$. Therefore $P_{tr}(S_{AND}) = P_{tr}(S_{OR}) = 2 \times 0.25 \times 0.75 = 0.375$ and the transition probabilities are 25% lower than those of the original signals. If all don't care bits can be filled with the last specified bit shifted into the scan chain, the transition probability will be lowered to the minimum value. As an example, let the ratio of the number of the specified bits in test cubes be 5% and the ratio of bit 0 or 1 of the specified bits 50%. Since the transitions can be produced by the specified bits and the transition probability in each specified bit is 50%, the transition probability of the test pattern can be lowered to 2.5% ideally. However, the process is so difficult that all don't care bits are filled with the last specified bits shifted into the scan chain in the LFSR reseeding methodology. But, if as many don't care bits as possible can be filled with the previous bits, the transition probability will be lowered and the switching activity can be significantly reduced.

Let $C = (c_0, \ldots, c_{m-1}) \in \{0, 1, X\}^m$ be a deterministic test cube and $S(C) = \{i \mid c_i \neq X\}$ the set of specified bits of C. $S(C)$ can be partitioned into a zero-set cube and one-set cube of specified bits. Let $S_0(C) = \{i \mid c_i = 0\}$ be a zero-set cube and $S_1(C) = \{i \mid c_i = 1\}$ a one-set cube. The initial seed to generate test cube C can be computed by the system of linear equations using the reseeding scheme proposed in [13]. The LFSR with the initial seed can generate a test sequence covering all specified bits in the given test cube. However, the specified bits generated by the LFSR cannot be distinguished from the don't care bits filled with the pseudo-random bits generated by the LFSR. The initial seeds to generate zero-set and one-set cubes can be computed in a similar manner with the set of specified bits. Considering the initial seeds for the zero-set and one-set cubes, all 1s generated by the LFSR for the zero-set cube and all 0s generated by the LFSR for the one-set cube are don't care bits for each cube. Therefore a new test pattern including all specified bits in the test cube can be generated by combining the patterns from zero-set cube and one-set cube. Table 1 shows the pattern value in each case of the patterns from zero-set and one-set cubes and the corresponding values for the original test cube. The first and second columns, respectively, show pattern values from the LFSRs for the zero-set and one-set cubes. The last column represents the value of the original test cube for each case. 0_x in the first and second columns represents a don't care bit filled with bit 0 by the LFSR for zero-set or one-set cube. 0_s represents a specified bit 0 generated by the LFSR for zero-set or one-set cube. 1_x and 1_s can be defined in a similar manner with 0_x and 0_s. 1_s cannot be generated by the LFSR for zero-set cube, and 0_s cannot be generated by the LFSR for one-set cube. There is no case where both LFSRs generate the specified bits (0_s and 1_s).

In cases where both patterns from the zero-set and one-set cubes are 0s, a value of the original test cube can be 0 or 'X' and the bit 0 from the one-set cube must be a don't care bit. Therefore if bit 0 is used for the scan input value, the value of the original test cube can be covered. Similarly, bit 1 can be used for the scan input value in cases where both patterns from the zero-set and one-set cubes are 1s. When the pattern from the zero-set cube is 1 and the pattern from the one-set

Table 1: Pattern values from LFSR for zero-set and one-set cubes

Pattern from zero-set cube	Pattern from one-set cube	Original test cube
0_x	0_x	X
0_s	0_x	0
0_x	1_s	1
0_x	1_x	X
0_s	1_x	0
1_x	0_x	X
1_x	1_x	X
1_x	1_s	1

cube is 0, the scan input value can be a don't care bit. In order to reduce the number of transitions in the scan chain shifting, the don't care bit can be filled with the previous scan input value instead of a random value from an LFSR. If a pattern from the zero-set cube is 0 and a pattern from the one-set cube is 1, the scan input value can be 0 or 1. When a bit 0 from zero-set cube is a specified bit (0_s), the scan input value should be 0. Similarly, in cases where 1 of one-set cube is a specified bit (1_s), the scan input value should be 1. In these cases, if a fixed value or the previous scan input value is used for the scan input value, the original test cube is not covered. Therefore in order to reduce the number of transitions, a previous scan input is used for a pattern of this type, and a pattern that differs from the original test cube can be corrected by using additional correction information. Since the density of don't care bits in a test cube is high, the number of bits to be corrected is not large. If this bit correction is ignored, transition probability is lowered to 25%, because the transition can be produced in cases where both patterns from the zero-set and one-set cubes are 0s or 1s. Therefore the scan input pattern generated by this approach has 50% lower transition probability than the original signals.

Fig. 1 shows experimental results for the ratios of the patterns from the zero-set and one-set cubes for the ISCAS'89 benchmark circuits. The ratios of the patterns to be corrected are lower than 2% in all cases. For large benchmark circuits (s13207, s15850, s38417 and s38584), the percentages of the patterns to be corrected became lower than approximately 0.5%. Therefore the size of the additional correction information is very small relative to the size of the initial LFSR seeds data. The ratios of the patterns which can be replaced with the previous scan input values are approximately 50% in all circuits. In other words, these patterns do not produce any transition in either the scan chains or the other half of the patterns related with the transitions. With the split LFSR reseeding methodology, about 50% of the transitions could be reduced so that the switching activity could be significantly lowered.

3 Encoding algorithm

This section explains the proposed method in order to reduce the number of transitions in the test pattern. The key idea of the proposed encoding scheme is to reduce the number of transitions in the don't care bits filled with pseudo-random bits by the LFSR. In order to accomplish this goal, the don't care bits must be distinguished from the specified bits in the generated patterns by the LFSR. If these distinguished don't care bits are substituted with the last values shifted into the scan chain, the transitions generated by the don't care bits can be removed easily. This approach reduces the number of transitions in the scan chain and requires an adequate size of the encoded data dependent on the number of specified bits of the test cube as compared with conventional LFSR reseeding scheme.

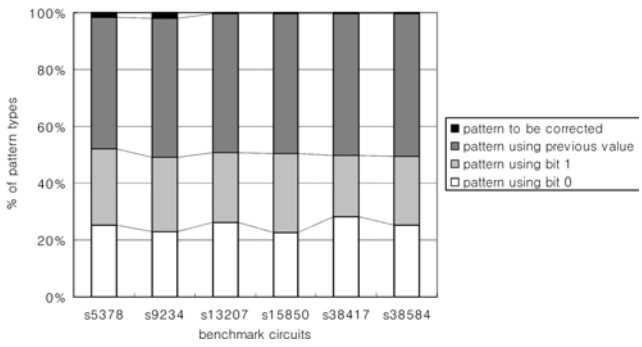

Fig. 1 *Ratios of the patterns from zero-set and one-set cubes*

Fig. 2 shows a pseudo-code for the process explained in the previous section. C in Fig. 2 means an original test cube generated by automatic test pattern generation (ATPG). The notation of $C[i]$ means the ith value of the original test cube C. C_0 and C_1 represent a zero-set cube and a one-set cube into which the original test cube is partitioned. P_0 and P_1 represent the test pattern which generated by the LFSRs for the zero-set and one-set cubes. The notations $P_0[i]$ and $P_1[i]$ stand for the i-th bits of patterns P_0 and P_1.

In order to remove the easy-to-detect faults, pseudo-random test patterns are applied. Then the deterministic test cubes are generated for the remaining hard-to-detect faults. Each test cube of the test set generated by ATPG is partitioned into the zero-set and one-set cubes (C_0 and C_1). The initial seed for the zero-set or one-set cube can be computed by any LFSR reseeding scheme. This study uses the VLMP-LFSR proposed in [18]. The low power test pattern sequence can be generated by using the patterns from the zero-set and one-set cubes. If both of the test pattern values from the zero-set and one-set cubes have the same value, this value can be used for the current pattern value. If the test pattern values from the zero-set and one-set cubes have a different value, the previous pattern value can be used for the current pattern value. In cases where the pattern value from the zero-set cube is equal to that from the one-set cube, the original test cube can be covered completely. However, if the pattern values from the partitioned cubes are different, not all specified bits in the original test cube can be covered as explained in the previous section. In this case, the corresponding bit in the correction pattern is set to bit 1 so that the corrected value is then used for the current scan input value. A correction operation is executed by inverting the corresponding correction bit and the previous scan input value. If the correction pattern is 0, the previous scan input value is used for the current scan input. Otherwise, the inverted value of the previous scan input value is used instead. The hardware implementation requirements for this process will be described in the next section. The following example illustrates the procedure for generatinga low power deterministic test pattern for a given test cube.

Example: Consider test cube $C = (X, X, 0, X, 1, X, X, 1, X, X, X)$ and a four-stage LFSR with feedback polynomial given by $h(x) = x^4 + x^3 + 1$. Solving the system of linear equations based on the feedback polynomial and the test cube will produce the initial seed (0, 1, 0, 1). The original test pattern generated by the given LFSR loaded by the computed initial seed will be $P = (0, 1, \underline{0}, 1, \underline{1}, 0, 0, \underline{1}, 0, 0, 0)$, where the underlined bits are the specified bits from the original test cube C. The generated pattern P using the computed initial seed contains five transitions. According to the process described in Fig. 2, the procedure to generate initial seeds for split LFSR and a correction pattern consists of the following steps.

(1) Test cube partitioning

The original test cube can be partitioned into the zero-set cube C_0 and the one-set cube C_1

$$C_0 = (X, X, 0, X, X, X, X, X, X, X, X)$$
$$C_1 = (X, X, X, X, 1, X, X, 1, X, X, X)$$

(2) Computation of the initial seeds

```
LP_PATTERN_GEN {
  do pseudo_random_fault_simulation ;
  do deterministic_ATPG ;                          /* generate the original test cube C */
  foreach ( C ) {
    do partition C into C0 and C1 ;                /* test cube partitioning */
    do compute the initial seeds from C0 ;         /* reseeding for the zero-set cube */
    do generate the zero-set cube pattern P0 ;
    do compute the initial seeds from C1 ;         /* reseeding for the zero-set cube */
    do generate the one-set cube pattern P1 ;
    PL = 0;                                        /* initialize the low power pattern */
    PC = 0;                                        /* initialize the correction pattern */
    for (i = 1 to pattern_size ) {
      if ( P0 [i] = P1 [i] )
        then PL [i] = P0 [i] = P1 [i];
      else–
        then
          if ( C [i] ≠ 'X' and PL [i-1] ≠ C [i] )  /* a pattern to be corrected */
            PL [i] = invert (PL [i-1]) ;           /* pattern correction */
            PC [i] = bit 1 ;                       /* update the correction pattern */
          else
            then PL [i] = PL [i-1] ;               /* the previous pattern value */
    }
  }
}
```

Fig. 2 *Pseudocode for a low power pattern and a correction pattern generation*

Consider a LFSR with feedback polynomial given by $h(x) = x^3 + x^2 + 1$. Solving the system of linear equations by the same manner with the original test cube, the initial seeds for zero-set cube and one-set cube can be computed. The initial seeds for each cube are (0, 1, 0) and (1, 1, 1).

(3) Test pattern for partitioned cubes generation

The zero-set pattern and one-set pattern generated by the given LFSR loaded by the computed initial seeds can be generated, where underlined bits are the specified bits from the zero-set cube and one-set cube

$$P_0 = (0, 1, \underline{0}, 0, 1, 1, 1, 0, 1, 0, 0)$$
$$P_1 = (1, 1, 1, 0, \underline{1}, 0, 0, \underline{1}, 1, 1, 0)$$

(4) Correction pattern generation

The temporary scan input pattern resulting from the method described in Section 2 is $P_T = (1, 1, \underline{1}, 0, \underline{1}, 1, 1, \underline{1}, 1, 1, 0)$. The boldfaced and underlined bit 1 in P_T is different from the specified bit of the original test cube. Therefore the correction pattern is $P_C = (0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0)$.

By generating the new low power pattern with P_T and P_C, the generated pattern is $P_L = (1, 1, \underline{0}, 0, \underline{1}, 1, 1, \underline{1}, 1, 1, 0)$. The pattern P from the original test cube has five transitions, whereas the pattern P_L generated by the proposed method for the same test cube contains three transitions.

In the proposed method, a test cube is encoded with two initial seeds for each split LFSR. Each seed can be computed by the same manner with the conventional LFSR reseeding methodology. In cases where the same bits are generated in the two LFSRs, those bits are shifted into the scan chain. If the two LFSRs generate the identical pattern, the last value shifted into the scan chain is repeatedly shifted into the scan chain and no transition is produced.

4 Proposed hardware architecture

This paper proposes that the reseeding method in [20, [21] be used for the LFSR reseeding scheme. Since the size of a seed is reduced according to the number of specified bits of a test cube in [20, 21], the test storage can be reduced. Fig. 3 shows the block diagram of the VLMP-LFSR reseeding scheme. An additional MUX in Fig. 3 is used to select whether the LFSR operates in the 'normal' mode or 'load seed' mode. Using polynomials of different lengths involves variable-length seeds.

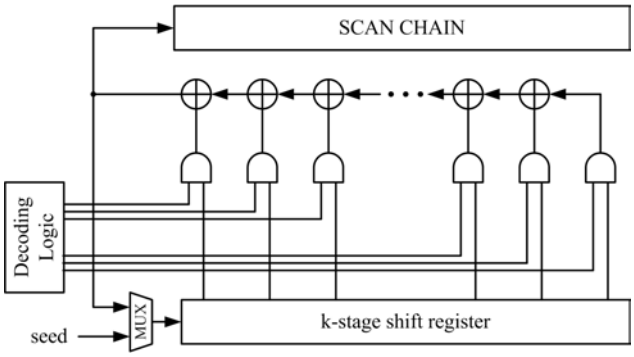

Fig. 3 *Variable-length multiple polynomial LFSR*

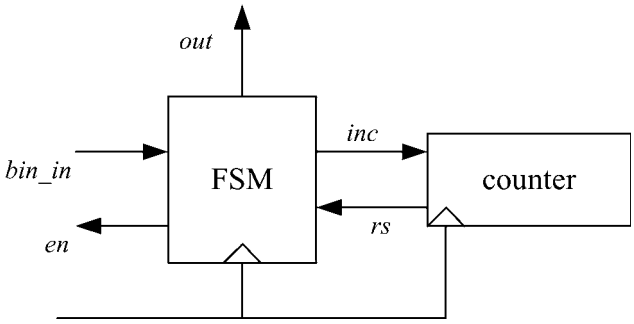

Fig. 5 *Block diagram of the correction decoder*

Fig. 4 shows the proposed hardware architecture. Compared to the conventional reseeding architecture, it has additional hardware consisting of two exclusive OR gates, one 2-to-1 MUXs, and one correction pattern decoder. In the proposed scheme, the LFSR is split into two sub blocks, namely a zero-set LFSR and a one-set LFSR. The output values of the two split LFSRs can be compared using the exclusive OR gate. If the two output values are equal, the output value of the zero-set of one-set LFSR can be shifted into the scan chain. In the proposed architecture, the output value of the zero-set LFSR is used for the shifted value. In cases where the two outputs have different values, the last value shifted into the scan chain is repeatedly shifted into the scan chain. As described in the previous sections, when the value stored in the first scan element is not covered the original test cube, the output of the correction decoder should be a bit 1 so that the inverted value is shifted into the scan chain by using an exclusive OR gate.

In order to reduce the test storage size, the correction patterns should be compressed. The correction pattern value is bit 0 except for the case where a current pattern bit should be inverted to cover the original test cube. As described in Section 2, the density of bit 1 in the correction pattern is much too low. For the large benchmark circuits, the ratio of bit 0 in the correction pattern is up to approximately 98%. In other words, only 2% of total test patterns should be inverted. For the largest circuits (s13207, s15850, s38417 and s38584), the ratios of the patterns to be corrected become lower than 0.5%. The correction pattern was compressed using run-length encoding and the compression ratio (CR) was found to be higher than 95%. The decoder decompresses this compressed correction pattern. The correction decoder can be efficiently implemented by a simple finite-state machine (FSM) and a small counter [25]. The block diagram of the correction decoder is shown in Fig. 5. The *bit_in* is the input to the FSM and an *en* signal is used to input the bit whenever the correction decoder is ready. The *inc* is used to increment the counter and *rs* indicates that the counter has finished counting. The *out* is the correction decoder output signal. The exclusive OR gate is used to invert the feedback value from the scan chain, when a particular bit of the correction pattern from the decoder is 1. When the last value shifted into the scan chain is shifted into the scan chain, no transition resulted. The correction pattern can be decompressed by the correction pattern decoder, which is a simple run-length decoder using a counter.

In the LFSR reseeding scheme, the length of the LFSR is dependent on S_{max}. Similarly, the length of the LFSR for zero-set cube or the LFSR for one-set cube is dependent on the largest number of specified bit 0s or 1s. Since these values are less than S_{max}, the length of the LFSR for zero-set cube or one-set cube is smaller than the length of the LFSR for the original test cube. In the proposed architecture, if the initial seed and the polynomial for the zero-set cube are swapped with the seed and the polynomial for the one-set cube, the generated pattern is not changed. In order to optimise the length of the two split LFSR, the numbers of the specified bits in the zero-set and one-set cubes should be accounted for. One LFSR degree, k_0 or k_1 in Fig. 4 should be $S0_{max}$ or $S1_{max}$, $S0_{max}$ or $S1_{max}$ is the largest number of the specified bits in any zero-set or

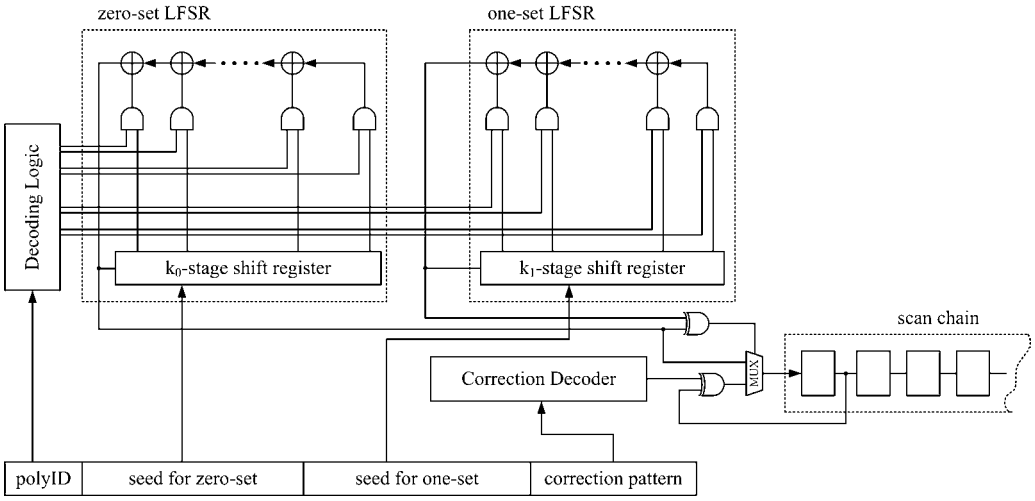

Fig. 4 *Proposed hardware architecture for deterministic BIST*

	the number of specified bits		
test cube	zero-set cube	one-set cube	origianl cube
C_0	**19**	(13)	**32**
C_1	(8)	16	24
C_2	15	(7)	22
C_3	(9)	12	21
C_4	(14)	17	31
C_5	(9)	13	22

Fig. 6 *Example of decision of the LFSR length*

one-set cube in the test set. The other LFSR degree should be the largest number of the specified bits in any zero-set or one-set cube with the smaller number of the specified bits in each cube pair. An example of this is shown in Fig. 6. The value for S_{max} is 32 for the original test cubes for the example in Figs. 4 and 6. In order to keep the probability of not finding a solution for the initial seeds in the MP-LFSR method proposed in [14, [15], the LFSR degree should be $S_{max} = 32$. In this example, $S0_{max}$ is 19, and $S1_{max}$ is 17. Therefore the length of the zero-set LFSR, k_0 should be 19. The length of the one-set LFSR, k_1 should be not 17 ($S1_{max}$) but 14, the maximum number in the circled numbers, the smaller number of the specified bits in each cube pair. Since the numbers of the specified bits in the one-set cube of C_1 and C_4 are larger than the length of the one-set LFSR, the initial seeds for the one-set cubes of C_1 and C_4 should be swapped with the initial seeds for the corresponding zero-set cubes. The length of the split LFSR is $k_0 + k_1 = 33$. Since the length of the split LFSR is similar to the LFSR degree for the original test cubes, the split LFSR can be implemented with little additional hardware overhead compared to the conventional LFSR. Table 2 compares the length of the split LFSR for the partitioned cubes with the normal LFSR for the original cubes. The length of the split LFSR is similar to the length of the normal LFSR. In cases of several benchmark circuits (S 9234, S3 8417 and S3 8584), the split LFSR length is shorter than the normal LFSR length.

5 Experimental results

The experiments were performed on the largest ISCAS'89 benchmark circuits. For each circuit, ATPG was performed to generate the deterministic test cubes for all faults targeting at 100% fault coverage. Each test cube was encoded into a corresponding seed for the zero-set and one-set cubes. The proposed method can be used for a mixed-mode BIST. In this case, different low power schemes such as the LT-RTPG proposed in [26, 27] should be used in order to reduce the transitions in the pseudo-random patterns.

Table 3 presents the experimental results in the test power. The second and the third columns represent the number of transitions in the original test patterns and the number of transitions in the proposed scheme, respectively. The last column represents the rate of reduction in the number of transitions. The rate of reduction in the number of transitions for each circuit is larger than 40% in all benchmark circuits. For large benchmark circuits, the rate of reduction in transitions becomes approximately 50%. As described in Section 2, the number of the transitions in the scan chain can be reduced by 50%, because no transition is produced in almost all cases where the values from the split LFSR are different in the proposed scheme.

Table 4 presents the experimental results of test storage. The second column shows the number of test cubes for the deterministic pattern generated by ATPG targeting at 100% fault coverage. The third column represents the total number of specified bits in each test cube of the test set. The fourth and the fifth columns represent the number of bits to be corrected in the patterns from the split LFSR and the amount of the correction patterns for the correction information, respectively. As described in Section 4, the correction patterns are generated by compressing the bit sequence to be corrected using the run-length encoding. The sixth column represents the amount of the total test storage required including the correction patterns. Since the test storage is heavily dependent on the size of the original test cube set, only the test storage is not sufficient for

Table 2: Comparison of the length of the split LFSR with the normal LFSR

Circuits	S_{max}	LFSR length for original cubes	$S0_{max}$	$S1_{max}$	LFSR length for zero-set cubes	LFSR length for one-set cubes
S5378	23	19	15	15	12	8
S9234	52	50	31	24	27	20
s13207	30	27	21	21	19	9
s15850	45	39	33	15	28	13
s38417	92	108	43	53	48	42
s38584	54	51	52	18	47	10

Table 3: Experimental results in the test power for the proposed method

Circuit	Original number of transitions	Number of transitions	Transition reduction, %
S5378	413 3891	236 5298	42.78
S9234	955 5186	499 0198	47.77
s13207	976 010 22	523 046 57	46.41
s15850	566 911 02	294 382 73	48.07
s38417	101 444 3748	510 308 038	49.70
s38584	575 745 133	307 177 381	46.65
Ave.	293 028 347	151 097 308	46.90

Table 4: Experimental results of test storage for the proposed method

Circuit	Num. test cubes	Num. specified bits	Num. corrected bits	Correction pattern storage	Total test storage	Compression Ratio (%)
S5378	305	3945	1000	2444	6258	90.41
S9234	488	10547	2276	3929	13134	89.10
s13207	593	8909	1986	5947	14992	96.39
s15850	447	8142	1704	4478	12393	95.46
s38417	1202	44139	7051	13283	51987	97.40
s38584	898	12208	3255	9929	13184	98.99
Ave.	656	14648	2879	6668	18568	94.63

Table 5: Comparison of the proposed scheme with previous schemes

Circuit	Dual-LFSR reseeding [22]			Hold flag reseeding [23]			Scan slice overlapping [24]			Proposed scheme		
	TSto	CR, %	Tred, %	TSto	CR, %	Tred, %	TSto	CR, %	Tred, %	TSto	CR, %	Tred, %
s5378	8756	79.44	25.04	NA	NA	NA	NA	70.38	NA	6258	90.41	42.78
s9234	19 440	67.61	24.35	10302	79	53	12 273	71.04	54.66	13 134	89.10	47.77
s13207	11 803	94.71	25.26	10 484	94	53	15837	90.42	83.06	14 992	96.39	46.41
s15850	14 518	90.02	25.14	11 411	93	52	16 517	81.02	71.17	12 393	95.46	48.07
s38417	66 234	92.01	24.90	32 152	95	52	48 527	70.13	61.77	51 987	97.40	49.70
s38584	23 835	95.36	24.70	31 152	93	40	44 896	79.92	70.78	13 184	98.99	46.65
Ave.	24 098	86.53	24.90	19 100	90.8	50	27 610	77.15	68.29	18 568	94.63	46.90

evaluating various reseeding schemes. The last column represents the CR. The CR can be calculated by dividing the total amount of storage required to explicitly store the deterministic test patterns (the product of the length of the scan chain and the number of test cubes) by the amount of the encoded test data. The CR for each benchmark circuit is higher than 90% in all cases. For large benchmark circuits (s13207, s15850, s38417 and s38584), the CRs became higher than approximately 95%.

Table 5 compares the results for the proposed scheme with previous schemes. For each scheme, the first columns and the second columns represent the test storage and the CRs, respectively. The last columns for each scheme represent the reduction of the number of transitions. It can be seen that the proposed scheme requires much smaller test storage and fewer transitions than the dual-LFSR reseeding in [22]. The CR of the proposed scheme for each benchmark circuits is higher than that of [22] in all most cases. When compared to the hold flag reseeding scheme in [23], test storage requirements of the proposed method are larger than those found for [23], but the proposed scheme requires achieves higher CR than those found for [23] for all benchmark circuits while retaining similar reduction of the number of transitions. Therefore if the proposed scheme will be applied in the same test cubes, test storage requirements can be smaller than those of the hold flag reseeding scheme. When compared to the scan slicing overlapping scheme in [24], the reduction of the number of transitions is not as much as the method of [24], but test storage requirements are much less than those found for [24] in all cases, except for s9234 and s38417. In all the circuits, the CR is much better than those found for that earlier study. As have been noted above, it can be expected that test storage requirements can be much smaller than those of the scan slicing overlapping scheme for the same test cubes. Also, the variance in the reduction of transitions of the proposed scheme is much lower than that of [24].

6 Conclusion

BIST is a good solution for testing a large SoC. The LFSR reseeding scheme is an attractive approach for achieving high fault coverage and compressing test data, but causes excessive power dissipation. This paper has proposed a new low power deterministic BIST methodology based on the split LFSR reseeding. The original test cubes are partitioned into zero-set and one-set cubes according to specified bits in the test cubes, and the split LFSR generates a zero-set cube or a one-set cube of a given test cube. Using the patterns generated from the zero-set cube and the one-set cube, many don't care bits in the original test cube can be replaced with the last values shifted into the scan chain, and no transition is produced. The test pattern generated by this approach covers the original test cube and reduces the number of transitions. By using the proposed scheme, the switching activity can be reduced by 50% for the largest ISCAS'89 benchmark circuits, while preserving the fault coverage with a conventional LFSR reseeding scheme and achieving high CR. Finally, the additional hardware requirements for the proposed scheme are very simple and small. Therefore the scheme presented greatly improves upon earlier efforts.

7 Acknowledgments

This work was supported by the Korea Science and Engineering Foundation Grant funded by the Korea government (MOST) (No. R01-2006-000-11038-0).

8 References

1 Abramovici, M., Breuer, M.A., and Friedman, A.D.: 'Digital systems testing and testable design' (IEEE Press, New York, 1990)
2 Bushnell, M.L., and Agrawal, V.D.: 'Essential of electronic testing for digital, memory and mixed-signal VLSI circuits' (Kluwer Academic Publishers, Boston, MA, 2000)
3 Jha, N., and Gupta, S.: 'Testing of digital systems' (Cambridge University Press, Cambridge, UK, 2003)
4 Bardell, P.H., McAnney, W., and Savir, J.: 'Built-in test for VLSI: pseudo-random techniques' (John Wiley and Sons, New York, 1987)
5 Agrawal, V.D., Kime, C.R., and Saluja, K.K.: 'A tutorial on built-in self-test, Part 1: Principles', *IEEE Design Test Comput.*, 1993, **10**, (1), pp. 73–82
6 Agrawal, V.D., Kime, C.R., and Saluja, K.K.: 'A tutorial on built-in self-test, Part 2: Principles', *IEEE Design Test Comput.*, 1993, **10**, (2), pp. 69–77
7 Brglez, F., Gloster, C., and Kedem, G.: 'Hardware-based weighted random pattern generation for boundary scan'. Proc. Design Automation Conf. (DAC), Las Vegas, NV, June 1989, pp. 264–274
8 Wunderlich, H.-J.: 'Multiple distributions for biased random test patterns', *IEEE Trans. Computer-Aided Design*, 1990, **9**, (6), pp. 584–593
9 Kim, H.-S., Lee, J.-K., and Kang, S.: 'A new multiple weight set calculation algorithm'. Proc. IEEE Int. Test Conf. (ITC), Baltimore, MD, October–November 2001, pp. 878–894
10 Touba, N.A., and McCluskey, E.J.: 'Altering a pseudo-random bit sequence for scan-based BIST'. Proc. IEEE Int. Test Conf. (ITC), Washington, DC, October 1996, pp. 167–175
11 Wunderlich, H.-J., and Kiefer, G.: 'Bit-flipping BIST'. Proc. IEEE Int. Conf. on Computer Aided Design (ICCAD), Austin, TX, October 1996, pp. 337–343
12 Kiefer, G., Vranken, H., Marinissen, E.J., and Wunderlich, H.-J.: 'Application of deterministic logic BIST on industrial circuits'. Proc. IEEE Int. Test Conf. (ITC), Atlantic City, NJ, October 2000, pp. 105–114
13 Koenemann, B.: 'LFSR-coded test pattern for scan designs'. Proc. European Test Conf., Munich, Germany, April 1991, pp. 237–242
14 Hellebrand, S., Reeb, B., Tarnick, S., and Wunderlich, H.-J.: 'Pattern generation for a deterministic BIST scheme'. Proc. of Int. Conf. on Computer-Aided Design (ICCAD), San Jose, CA, November 1995, pp. 88–94
15 Hellebrand, S., Rajski, J., Tarnick, S., Venkataraman, S., and Coutois, B.: 'Built-in test for circuits with scan based on reseeding of multiple-polynomial linear feedback shift registers', *IEEE Trans. Comput.*, 1995, **44**, (2), pp. 223–233
16 Zacharia, N., Rajski, J., Tyszer, J., and Waicukauski, J.A.: 'Two-dimensional test decompressor for multiple scan designs'. Proc. of Int. Test Conf. (ITC), Washington, DC, October 1996, pp. 186–194
17 Rajski, J., Tyszer, J., and Zacharia, N.: 'Test data decompression for multiple scan designs with boundary scan', *IEEE Trans. Comput.*, 1998, **47**, (11), pp. 1188–1200
18 Krishna, C.V., Jas, A., and Touba, N.A.: 'Test vector encoding using partial LFSR reseeding'. Proc. of Int. Test Conf. (ITC), Baltimore, MD, October–November 2001, pp. 885–893
19 Krishna, C.V., Jas, A., and Touba, N.A.: 'Achieving high encoding efficiency with partial dynamic LFSR reseeding', *ACM Trans. Design Automation Electron. Syst.*, 2004, **9**, (4), pp. 500–516
20 Kim, H.-S., Kim, Y.J., and Kang, S.: 'Test-decompression mechanism using a variable-length multiple-polynomial LFSR', *IEEE Trans. Very Large Scale Integration Syst.*, 2003, **11**, (4), pp. 687–690
21 Kim, H.-S., and Kang, S.: 'Increasing encoding efficiency of LFSR reseeding-based test compression', *IEEE Trans. Computer-Aided Design of Integrated Circuits Syst.*, 2006, **25**, (5), pp. 913–917
22 Rosinger, P.M., Al-Hashimi, B.M., and Nicolici, N.: 'Dual multiple-polynomial LFSR for low-power mixed-mode BIST', *IEE Proc., Comput. Digital Tech.*, 2003, **150**, (4), pp. 209–217
23 Lee, J., and Touba, N.A.: 'Low power test data compression based on LFSR reseeding'. Proc. IEEE Int. Conf. on Computer Design (ICCD), San Jose, CA, October 2004, pp. 180–185
24 Li, J., Han, Y., and Li, X.: 'Deterministic and low power BIST based on scan slice overlapping'. IEEE Int. Symp. on Circuits and Systems, Kobe, Japan, May 2005, pp. 5670–5673
25 Chandra, A., and Chakrabarty, K.: 'System-on-a-chip test-data compression and decomposition architectures based on Golomb codes', *IEEE Trans. Computer-Aided Design Integrated Circuits Syst.*, 2001, **20**, (3), pp. 355–368
26 Wang, S., and Gupta, S.K.: 'LT-RTPG: a new test-per-scan BIST TPG for low switching activity', *IEEE Trans. Computer-Aided Design Integrated Circuits Syst.*, 2006, **25**, (8), pp. 1565–1574
27 Kim, Y., Yang, M.-H., Lee, Y., and Kang, S.: 'A new low power test pattern generator using a transition monitoring window based on BIST architecture'. Proc. Asian Test Symp., Calcutta, India, December 2005, pp. 230–235