Weight Set Optimization for Weighted Random Pattern Generation

Hangkyu Lee, Hongsik Kim and Sungho Kang

Department of Electrical Engineering, Yonsei University
134 Shinchon-Dong, Seodaemun-Gu, Seoul, Korea
Phone: +82-2-361-2775, Fax: +82-2-312-7735, E-MAIL: shkang@bubble.yonsei.ac.kr

***Abstract*—A lot of circuits have some random pattern resistant faults which cannot be detected by pseudo-random patterns. So weighted random pattern testing of built-in self-test(BIST) is accepted as a good solution for detecting random pattern resistant faults. In weighted random pattern testing it is an important issue to find the optimal weight sets for achieving a high fault coverage using a small number of weighted random patterns. In this paper, we present a new weight set optimization algorithm which can generate the optimal weight sets in an efficient way. Experimental results using ISCAS 85 benchmark circuits prove the effectiveness of the new weight set optimization algorithm.**

1 Introduction

Pseudo-random pattern testing is generally adopted in built-in self-test because of its low hardware overhead. However, random pattern resistant faults can scarcely be detected by it. Weighted random pattern testing [1,2,3,4] has been proposed to overcome this problem. By means of adding a small amount of hardware to LFSR(Linear Feedback Shift Register) a high fault coverage can be achieved within a proper amount of time. It is a crucial issue to find the optimal weight sets in weighted random pattern testing [5,6,7]. Weight sets should be evaluated in terms of the number of weight sets and the weighted random pattern length. As the number of weight sets increases the weighted random pattern length required to provide a high fault coverage decreases. But hardware overhead caused by the weight calculation logic increases. On the contrary, as the number of weight sets decreases the hardware overhead decreases too. But a lot of weighted random patterns are required. Therefore weight set optimization is essential to achieve a sufficient fault coverage for a small number of weighted random patterns using a small amount of hardware. In a case where weighted random pattern generation is based on the multiple weight sets, the precomputed deterministic test pattern set is, for the most part, partitioned into several subsets considering the Hamming distance between test patterns. The number of weight sets and the weighted random pattern length are greatly dependent on the maximum Hamming distance. Therefore it is an important factor in optimizing weight sets to select the proper maximum Hamming distance.

In this paper, a new weight set optimization algorithm is presented and the new method of selecting the proper maximum Hamming distance is suggested.

2 Sampling Probability

Let P_j be the probability that a deterministic pattern t_j is identical with a weighted random pattern. After applying N weighted random patterns the probability that a deterministic pattern has not been sampled is $(1 - P_j)^N$. Let P_{Nj} be the probability that a deterministic pattern t is sampled by N weighted random patterns. Thus

$$P_{Nj} = 1 - (1 - P_j)^N \qquad (1)$$

The both sides of equation(1) can be expressed by means of the logarithm and then the equation is transformed into the following.

$$N = ln\frac{(1 - P_{N_j})}{(1 - P_j)} \tag{2}$$

From equation(2) we can infer that the number of random patterns required for sampling a set of deterministic patterns is mainly dependent on the deterministic patterns which have the lowest sampling probabilities. In other words, weighted random pattern generation systems waste a lot of weighted random patterns to sample them. So we suggest a new algorithm that generates the optimized weight sets by increasing the lowest sampling probabilities to a certain degree. There usually exist don't care bits in a deterministic test pattern. We bias these don't care bits so as for the lowest sampling probabilities to go up. Weight set generation is based on the biased deterministic test pattern set. It is a matter of course that as the lower sampling probabilities increase the higher sampling probabilities decrease. However, the deterministic test patterns of a high sampling probability are likely to be sampled by the weighted random patterns applied for sampling the deterministic test patterns of a low sampling probability. If the lowest sampling probabilities can be raised to a considerable degree the weighted random pattern length may be considerably reduced. Thus the newly generated weight sets can be so effective that a small number of weighted random patterns are required to attain a high fault coverage.

Let's assume that $T = \{t_1, t_2, \ldots, t_n\}$ is a deterministic test pattern set and $t_j[i]$ is i_{th} bit of a deterministic test pattern, t_j. The weight of bit position i, w_i can be computed by the following equation.

$$w_i = \frac{\left|\{t_j \in T | t_j[i] = 1\}\right|}{\left|\{t_j \in T | t_j[i] \neq x\}\right|} \tag{3}$$

3 Weight Set Optimization

The weight set optimization algorithm is shown in Figure 1. First, a deterministic test set is generated by the ATPG system. Then the deterministic test pattern set is divided into several subsets according to the Hamming distance between the deterministic test patterns. Only the largest subset is used in generating a weight set.

If two deterministic test patterns have a large Hamming distance they cannot be included in a test pattern subset because too many conflicts are generated in weight calculation. The Hamming distance between a deterministic pattern and each test pattern of a deterministic test pattern set is calculated.

```
set_optimization_wrpg()
{
   while ( fault_coverage < target_fault_coverage )
   {
      /* generate a deterministic test set */
         deterministic_testgen();
      /* divide a test set into several subset
         and select the largest subset */
         divide_test_set();
      /* generate a weight set */
         generate_weight_set();
      /* find the two lowest sampling probability */
         calculate_sampling_prob();
      /* bias the don't care bits and generate
         a new weight set */
         regenerate_weight_set();
         calculate_sampling_prob();

      /* increase the lowest sampling probability */
         while ( P_new > P_old )
         {
                regenerate_weight_set();
                calculate_sampling_prob();
         }

      /* weighted random pattern generation
         and fault simulation */
         fault_coverage = wrpg_fault_simulation();
   }
}
```

Figure 1: Weight set optimization and weighted random pattern generation

So only the deterministic test pattern which has the Hamming distances smaller than the maximum Hamming distance for all the patterns included in the test pattern set can be used in the weight set generation process. In weighted random pattern generation the maximum Hamming distance is an important factor for the weight set performance. The weighted random pattern length and the number of weight sets are mainly dependent on the maximum Hamming distance. The larger the maximum Hamming distance is the longer the weighted random pattern length is. But a small maximum Hamming distance generates a large number of weight sets. Therefore, the maximum Hamming distance should be optimized so that the generated weight sets can be efficient in terms of the weighted random pattern length and the number of weight sets. So, we suggest the new method of finding the optimal value of the maximum Hamming distance in the next section.

After generating a weight set by equation (3) using the largest subset of the precomputed deterministic test pattern set we can calculate the sampling probability for each test pattern. Let $t_j = (t_j[1], t_j[2], \ldots, t_j[m])$ be a deterministic test pattern included in the largest subset of the precomputed test pattern set. $W = (w_1, w_2, \ldots, w_m)$ is the generated weight set of m input circuit. The sampling probability P_j of a deterministic test pattern t_j can be calculated by the following equation.

$$P_j = \prod_{i=0, t_j[i] \neq X}^{m} \{(w_i \times t_j[i]) + (1 - w_i) \times (1 - t_j[i])\} \quad (4)$$

Two deterministic test patterns that have the lowest sampling probabilities are searched for. And then a small number of don't care logic values which are located on the bits where these two deterministic test patterns have the same specified logic value are biased to it. After that, a weight set is recalculated and the two lowest sampling probabilities are found again. While the new lowest sampling probability, P_{new} is higher than the previous lowest sampling probability, P_{old} the optimization process is repeated. If P_{new} is lower than P_{old} the process is terminated. When the weight set optimization process is completed weighted random patterns are generated using the optimized weight set and fault simulation is performed. Until a sufficient fault coverage is attained the previous procedure is repeated.

4 Hamming Distance

Experimental results were greatly dependent on the maximum Hamming distance. So we have devised the new method of searching for the optimal value of the the maximum Hamming distance. Let *Init_Hdis, Test_length* and *Num_inputs* be the initial value of the the maximum Hamming distance, the deterministic test pattern length and the number of primary inputs of a circuit, respectively. Various experimental results show that the optimal value of the maximum Hamming distance is dependent on the deterministic test length and the number of primary inputs. Therefore, we find the initial value of the maximum Hamming distance according to the following equation.

$$Init\_Hdis = CN \times Num\_inputs + CL \times Test\_length$$

CN and CL are coefficients of the number of primary inputs and the deterministic test pattern length, respectively and can be found from experimental results.

After weighted random pattern generation and fault simulation using the initial value of the maximum Hamming distance is increased by 1 and fault simulation is performed using it. This process is repeated CI times. If the results of these fault simulations indicate a tendency of improving the performance of weight sets in terms of the number of weight sets and the weighted random pattern length the previous procedure is repeated. Otherwise, the procedure is terminated. Occasionally the number of weight sets increases while the number of weighted random pattern length decreases or the number of weight sets decreases while the number of weighted random pattern length increases after weighted random pattern generation and fault simulation. If the best result of the previous weighted random pattern generation and fault simulation shows that the weighted random pattern length is l and the number of weight sets is s the reduction of a weight set is evaluated as same as that of the l/s weighted random patterns. For example, let's assume that l and s are 8000 and 4, respectively. If a weight is increased and 2500 weighted random patterns are decreased the weight set is identified as being improved, for l/s, 2000 is smaller than the reduction of weighted random patterns, 2500. And then, the same procedure except for gradually decreasing the value of the maximum Hamming distance by 1 from the initial value is performed. By means of this method the optimal value of the maximum Hamming distance can be efficiently found in a proper amount of time.

5 Experimental results

We have implemented the weighted random pattern generation system that adopts the new weight optimization algorithm and have performed fault simulation using ISCAS 85 benchmark circuits. Weighted random pattern generation and fault simulation for the pseudo random pattern resistant circuits have been performed. The comparison between X-test [5] and the new method that adopts the weight set optimization is given in Table 1 where

'Optimization' signifies the latter. Weighted random pattern generation is terminated when m consecutive weighted random patterns don't detect any new fault. A large value of m lead to a small number of weight sets and a large weighted random pattern length. On the contrary, a small value of m makes the linking patterns which don't detect any new fault reduced. So, although the number of weighted random patterns is reduced weight sets increase. We fix the value of m to 1024 for experimental results. For all the cases the weighted random pattern length of adopting the new weight set optimization algorithm is smaller than that of not adopting it whereas the number of weight sets is the same. The results of Table 1 prove the effectiveness of the weight set optimization algorithm in terms of the weighted random pattern length.

Circuits	X-test		Optimization	
	Patterns	Weights	Patterns	Weights
c2670	5632	3	4352	3
c3540	3808	2	3488	2
c7552	17728	5	16096	5

Table 1: Results of fault simulation

Circuits	Inputs	Tests	Initial value	Optimal value
c2670	233	720	38	36
c3540	50	710	19	20
c7552	207	933	39	41

Table 2: The initial and the optimal degrees of the limitation of the permissible Hamming distance

The initial and the optimal values of the maximum Hamming distance are shown in Table 2. In these experiments the user defined-values of CN, CL and CI are fixed to 0.1, 0.02 and 3 considering the previous experimental results of various circuits, respectively. The difference between the initial and the optimal values is very small for all the cases. It means that a small number of weighted random pattern generation and fault simulation is required so as to find the optimal value of the maximum Hamming distance.

6 Conclusion

The performance of weight sets in weighted random pattern testing is a crucial issue because it has a great effect on test time and hardware overhead. We have developed the weight set optimization algorithm and have devised the method of finding the optimal value of the maximum Hamming distance. Experimental results show that the new weight set optimization algorithm generates high-performance weight sets in terms of the weighted random pattern length. Therefore, if this weight set optimization algorithm is adopted in weighted random pattern testing the time efficiency can be achieved without extra hardware. The optimal value of the maximum Hamming distance can be effectively found by the new method presented in this paper.

References

[1] J. Hartmann and G. Kemnitz, "How to Do Weighted Random Testing for BIST," in *Proc. of International Conference on Computer Aided Design*, 1993, pp. 568–571.

[2] C-A. C. andS. K. Gupta and S. K. Gupta, "A Methodology to Design Efficient BIST Test Pattern Generators," in *Proc. of International Test Conference*, 1995, pp. 814–823.

[3] R. Kapur, S. Patil, T. J. Snethen and T. W. Williams, "Design of an Efficient Weighted Random Pattern Generation System," in *Proc. of International Test Conference*, 1994, pp. 491–500.

[4] D. J. Neebel and C. R. kime, "Inhomogeneous Cellular Automata for Weighted Random Pattern Generation," in *Proc. of International Test Conference*, 1993, pp. 1013–1021.

[5] B. Reeb and H-J. Wunderlich, "Deterministic Pattern Generation for Weighted Random Pattern Testing," in *Proc. of European Design and Test Conference*, 1996, pp. 30–36.

[6] M. Bershteyn, "Calculation of Multiple Sets of Weighted Random Testing," in *Proc. of International Test Conference*, 1993, pp. 1031–1040.

[7] M. A. Miranda and C. A. Lopez-Barrio, "Generation of Optimized Single Distributions of Weights for Random Built-In Self-Test," in *Proc. of International Test Conference*, 1993, pp. 1023–1030.