A Reconfigurable Shared Scan-in Architecture

Samitha Samaranayake* Emil Gizdarski** Nodari Sitchinava*
Frederic Neuveux** Rohit Kapur** T. W. Williams**

* Massachusetts Institute of Technology, 77 Massachusetts Avenue, Cambridge, MA 02139
** Synopsys Inc., 700 East Middlefield Road, Mountain View, CA 94043

Abstract

In this paper, an efficient technique for test data volume reduction based on the shared scan-in (Illinois Scan) architecture and the scan chain reconfiguration (Dynamic Scan) architecture is defined. The composite architecture is created with analysis that relies on the compatibility relation of scan chains. Topological analysis and compatibility analysis are used to maximize gains in test data volume and test application time. The goal of the proposed synthesis procedure is to test all detectable faults in broadcast test mode using minimum scan-chain configurations. As a result, more aggressive sharing of scan inputs can be applied for test data volume and test application time reduction. The experimental results demonstrate the efficiency of the proposed architecture for real-industrial circuits.

1. Introduction

Recently, a number of solutions have been proposed that reduce the overall cost of test as measured by test pins, data volume and test application time. These methods span from simple changes to the design (Shared Scan-in [1,2,3] and Dynamic Scan [4]) to significant modifications that rely on decompressing patterns on the product for the purposes of test [5,6,7,8,9]. The simple architectures have shown to provide benefits, while minimizing the hardware overhead. In particular, the Illinois Scan architecture shows a lot of potential [2]. Minimizing the number of test patterns that are applied through the serial test mode has been shown to reduce test data volume [3]. Clearly, further reduction can be achieved if all test patterns are applied through the broadcast mode.

The compatibility introduced in [10] provides an elegant ATPG model that can be used to solve this problem. From ATPG point of view, a full-scan circuit can be considered as a combinational circuit. Accordingly, two inputs of this circuit are *compatible* if they can be shorted together without introducing any redundant fault in the circuit. Since the same (or opposite) logic value is applied to all inputs in a compatibility class during testing, the test data volume can be significantly reduced [10]. Further improvement in this direction was achieved using D-compatibility [11], Ck-compatibility [12] and composite Pk-compatibility [13].

This paper is organized as follows. In Section 2, the reconfigurable architecture is described. In Section 3, the synthesis procedure is presented. In Section 4, the experimental results are provided. Section 5 concludes the paper.

2. The reconfigurable shared scan-in architecture

Consider the traditional scan architecture of Figure 1 to apply deterministic ATPG test patterns. In this architecture every scan cell (flip-flop) can be provided a stimulus and every scan cell can be observed through a shift operation. Since the configuration is rigid every test pattern that uses one cell in the scan chain to control or observe values would have to operate all the scan cells. Hence, a typical test pattern would specify all the stimulus and response values for the scan cells.

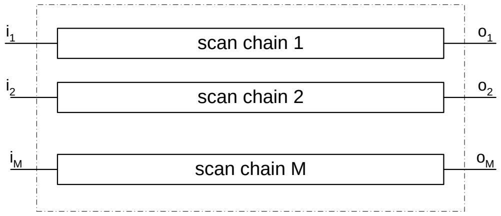

Figure 1: Full-scan architecture for deterministic ATPG

If one looks at the profile of the test pattern that detects a fault a small fraction of the stimulus points need to be specified and observed. Thus, ATPG tools randomly fill the unspecified positions after ATPG compaction is performed to obtain values for every scan flip-flop. This fact has been utilized to create many new solutions to lower the test data volume and test application time [5,6,7,8,9].

The solutions typically address the stimulus part and the response part of the test patterns separately. It should be noted that in all the cases a mix and match of the input side and output side solutions is possible. On the output side the signature register has been the most commonly used mechanism. The dynamic scan architecture proposes an alternative solution for output side test data reduction, while preserving observability and X-tolerance of the circuit. On the input side, shared scan-in and dynamic scan architectures both have shown significant promises with low overhead. In general, the dynamic scan architecture has a higher degree of freedom for scan-chain reconfiguration including bypassing and changing an order of scan-segments within a scan chain. The reason is that dynamic scan architecture is oriented to a passive strategy for test data volume reduction based on analysis and modification of test patterns. In contrast, the shared scan architecture (as it is shown in Figure 2), is oriented to an active strategy for test data volume reduction utilizing the fact that many different test patterns may detect a particular fault. Since the scan chains are shorter a significant test application time reduction can be achieved. However, this architecture creates dependencies in values across the scan cells connected to the same scan-in. As a result, some faults that require different values for the dependant scan-cells cannot be tested in this configuration called a broadcast test mode. These faults are tested in another mode where the all scan chains are configured as a single scan chain called a serial test mode. Benefits are achieved when most of the faults are tested during broadcast test mode. However, as the sharing gets more aggressive fewer faults can be tested during the broadcast mode.

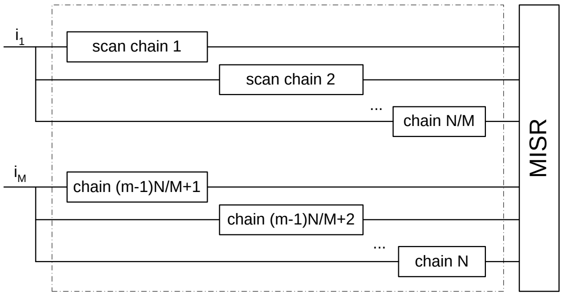

Figure 2: Shared scan-in architecture and MISR to compact input and output stimulus

Clearly, the number of potential conflicts can be initially reduced by performing an analysis to improve the membership of flip-flops in the architecture of Figure 2. However, there always exist faults that cannot be tested in the broadcast test mode. These faults must be tested in the serial scan mode. To avoid this, we propose a reconfigurable scan-chain architecture such that all faults can be detected in broadcast test mode. We apply compatibility analysis to find and minimize the number of these configurations. More formally, the target fault set is divided into subsets such that each subset can be tested by a single configuration. The derived scan-chain configurations define a composite compatibility relation if the intersection of untested detectable faults is empty [13].

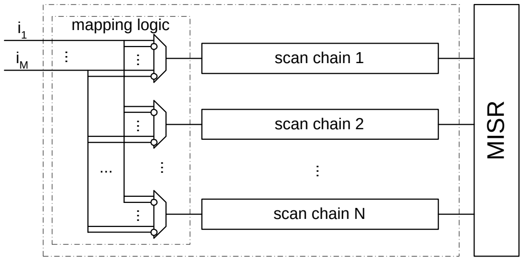

Figure 3: The reconfigurable shared scan-in architecture

The combined scan architecture is shown in Figure 3. Accordingly, the membership of scan chains to the scan-in is defined by mapping logic consisting of muxes, wires and inverters. We only consider direct and inverse compatibility relations between scan chains during the compatibility analysis. Reconfiguring the scan chains provides an efficient mechanism to generate test patterns for all detectable faults in broadcast mode. As a result, we may expect a considerable reduction in both test data volume and test application time for the reconfigurable architecture in respect to Illinois Scan architecture (Figure 2). In the next section, we present more details of the proposed synthesis procedure.

3. Synthesis procedure

The implemented synthesis procedure includes the following steps:

1) Defining a set of scan cells for each scan chain;
2) DFT synthesis and test logic insertion;
3) Selecting the first scan-chain configuration;
4) Defining the target fault list and running compatibility analysis to find next configuration;
5) Generating test patterns for all untested faults using the current scan-chain configuration;
6) Repeating steps 4 and 5 until all detectable faults are tested, i.e., at least one test pattern detect each detectable fault.

Hereafter, we will present more details on the steps of this synthesis procedure in two subsections. In

Section 3.1, we describe the processes of determining the membership of the scan chains and selecting the first configuration. In Section 3.2, we describe the compatibility analysis.

3.1. Conflict analysis-based DFT synthesis

The efficiency of the proposed architecture can be improved by minimizing the number of untested faults in each configuration. If all the faults can be tested in the first configuration, we would not be required multiple configurations [1,14,15]. Unfortunately, this is not possible when a large number of scan chains are connected to a single input. However, by minimizing the number of untested faults for each configuration, the number of configurations, processing time and total number of test patterns can potentially be reduced. This step is especially important for the first configuration, since the target fault list includes all stuck-at faults. Minimizing the number of untested faults in the first configuration significantly reduces the time required to select the subsequent configurations. To achieve this, a simple topological analysis is applied before constructing scan chains such that the number of potential conflicts is minimized. Accordingly, a *potential conflict* occurs if a pair of cells in one logic cone belong to different scan chains driven by the same scan input. Our goal is to create N different scan chains of equal length (L) driven by M inputs such that the number of potential conflicts is minimized. The input for this phase is the original circuit prior to test logic insertion and the output is N scan chains driven by M different scan inputs. To keep the degree of freedom for DFT synthesis as high as possible, the order of each scan chain is not specified. In this way, the DFT synthesis engine has a higher flexibility to order each scan chains such that physical and routing constraints are satisfied. It should be noted that an alternative solution that specifies the exact scan chain ordering would increase the number of detected faults. However, fixing the scan chains without considering physical and routing constraints is unacceptable [16] and therefore was not considered.

When creating the first configuration, it is important to have an algorithm that doesn't relay on ATPG. Therefore, we used the following heuristic approach to determine the first configuration. First, the logic cones are topologically ordered by the output of the cone. Next, we go through the inputs of each cone and assign the first L unassigned scan cells to the first scan chain. The following L unassigned scan cells are assigned to the next scan chain and the process continues until all scan cells are assigned to a chain. Given the way in which the scan chains were selected, it is likely that most scan cells in a given cone are either in the same scan chain or scan chains immediately before and after the scan chain. Therefore, a majority of the scan cells that are dependent on each other are either in the same scan chain or in adjacent scan chains. This means that we can obtain the first configuration by assigning every Mth scan chain to the same scan input. In this way, *width compression (input space reduction)* from N*L to M*L is achieved, i.e., *compression ratio* is N/M. Clearly, if the maximum number of cone inputs K is smaller than M*L then a large percentage of the faults can be tested in the first configuration.

Alternatively, a more computationally intensive Max-Cut algorithm could be used for deriving the first configuration. In this case, the potential conflicts are represented by a graph G=(V,E), where the set of vertices corresponds to the set of scan cells and there exists an edge between any two vertices when the corresponding scan cells share at least one common cone. The number of potential conflicts can be minimized by an N way Max-Cut of equal size. Next, the first configuration can be derived by an M way Max-Cut over the scan chains. Although, this method guarantees a lower number of conflicts than the presented heuristic approach, the computation time can be impractical for some industrial circuits. Therefore, the heuristic method was used in our experiments.

3.2. Compatibility Analysis

In general, two ATPG-based strategies for width compression can be used: *passive* and *active*. The passive strategy [11] is based on analysis of a set of pre-computed test patterns while the active strategy [10,12,13] identifies the compatibility relations by setting constraints during ATPG. Clearly, the active strategy can achieve much higher reduction in the number of compatibility classes. However, the processing time of the active strategy is much higher because the compatibility analysis may involve hundreds of ATPG runs for a target fault list. In this sense, the processing time of compatibility analysis may become an important issue that, in fact, determines the overall efficiency and applicability of the proposed synthesis procedure. Taking this into account, we try to keep processing time of the compatibility analysis at a reasonable level.

Accordingly, the compatibility analysis is run for all untested faults after the first configuration. The goal is to minimize the number of untested faults from the target fault list using the current scan-chain configuration. The target list includes all faults detected by ATPG run without constraints, i.e., any dependency between scan chains. In this way, we assume that the ATPG system should be able to generate a test pattern for each fault in the target list by a given abort limit. This approach takes into account the fact that ATPG is an NP-complete problem and some aborted faults may occur when the abort limit is set low during the compatibility analysis. As a result, the final fault coverage may vary even when the same abort limits are used during regular ATPG and compatibility analysis because some aborted faults may be detected by simulation.

The implemented procedure for compatibility analysis has three phases:

1) *Deriving compatibility classes*: All scan chains are initially considered as independent. First, the number of independent scan chains is minimized using passive strategy for width compression based on the pre-computed set of test patterns. As a result, all directly or inversely compatible chains form up to M compatibility classes.
2) *Minimizing the number of the independent scan chains*: ATPG for the target fault list is run to check compatibility relations between independent scan chains and compatibility classes. If an independent scan chain is either directly or inversely compatible to all scan chains in a compatibility class, then this chain is included into this compatibility class.
3) *Minimizing number of untested faults*: The untested faults introduced by including an independent scan chain into a compatibility class are identified by running ATPG for the target list. The best result determines the compatibility class and compatibility relation for the current independent chain. This phase continue until all independent scan chains are included into M compatibility classes.

The worst case complexity of this compatibility procedure is O(MN′) number of ATPG runs where M is the number of scan inputs and N′ is the number of independent scan chains after the first (passive) phases. More formally, during the first phase, we consider that test patterns are consistent with a certain configuration if there does not exist a conflict, i.e., all test patterns can be applied with this scan-chain configuration. In this way, the number of independent scan chains can be significantly reduced without running ATPG. However, the compatibility analysis could be still time consuming if the number of faults in the target list is too big since many scan chains may remain independent after the first phase.

4. Experimental Results

The proposed synthesis procedure was tested using the ISCAS'89 benchmark circuits and three industrial circuits. Table 1 shows the characteristics of the industrial circuits. Each circuit was run through the proposed procedure and compared against results of the regular ATPG and Illinois Scan. The comparisons were performed for different compression ratios to analyze the behavior of the proposed synthesis procedure.

Circuit	Gates	Faults	Scan cells	Cone size	ATPG time
Circuit A	230k	481k	9700	1887	0:15
Circuit B	390k	554k	12500	916	0:15
Circuit C	1083k	2740k	69000	5454	7:05

Table 1: Characteristics of the industrial circuits

All experiments were run using the Synopsys tools DFT compiler and TetraMax on Ultrasparc II, 400 MHz. For all experiments, the ATPG abort limit was set at 10. First, we created N scan chains with maximum length L as described in section 3.1. Next, we run experiments for the regular ATPG and Illinois Scan using the first scan chain configuration. Finally, we derived a set of scan chain configurations using the compatibility analysis procedure described in section 3.2. These configurations were used to generate test patterns by sequentially going through all configurations until the same or better test coverage than the regular ATPG was achieved. For the industrial circuits, we restricted the number of configurations to 4. For the ISCAS'89 benchmark circuits, the stopping criterion was the same fault coverage as the regular ATPG.

When applying high compression ratio to a circuit, the primary inputs could take up a large portion of the test data volume. To avoid this, a wrapper around the primary inputs and outputs was created so that each wrapper chain was no longer than the longest scan chain. In this experiment, the fault list consisted of all faults in the processed circuit including the combinational part, all scan chains and wrapper logic. The experimental results are given in Table 2.

The data volume reduction (DVR) for Illinois Scan was estimated as $PN/(P_bM + P_sN)$, where P, P_b, P_s are the number of patterns required for regular ATPG, broadcast mode, serial mode and N, M are the number of scan chains, scan inputs for the proposed scheme. The DVR and application time reduction (TATR) for the proposed scheme are calculated as PN/P_rM, where P, P_r are the number of patterns required for regular ATPG and all configurations of the proposed scheme.

In our experiments, we have used a minimum number of scan inputs M to show the effectiveness of the proposed scheme with limited flexibility for reconfiguring. However, lets assume that Circuit A had 8 scan inputs in regular ATPG mode and 2 inputs with 32 scan chains in the proposed scheme. In this case, to achieve the same compression ratio N/M, we need increase the number of scan chains from 32 to 128. Having more scan chains actually gives more flexibility during compatibility analysis but may also increase area overhead. Furthermore, as the scan chains become shorter the DFT synthesis becomes more constrained and routing problems could occur. Therefore, the designer has to take into account these tradeoffs.

The last two columns in Table 2 showed the deviations in the fault coverage and the ATPG overhead in respect to the regular ATPG. When the ratio between input space M*L and the maximum number of cone inputs K is too small the processing time for the compatibility analysis may significantly exceed the processing time of the regular ATPG. Also, we observed a negative impact on the fault coverage when this ratio is too small. The reason for a gain in the fault coverage was that some aborted faults during the regular ATPG were detected in another scan-chain configuration. The reason for a lost in the fault coverage was due to an increased number of aborted faults during compatibility analysis when input space of the circuits was too small.

These results show that Illinois Scan becomes ineffective after a certain compression ratio. Obviously, the proposed synthesis procedure is able to achieve higher DVR than Illinois Scan. The reason is that fewer faults are detected in the broadcast mode as the compression ratio N/M increases. Therefore, Illinois scan takes a big hit in having to run a large number of serial mode patterns. In the proposed scheme, we eliminate this problem by detecting all faults in broadcast mode. Thereby, allowing us to apply scan sharing more aggressively.

For the processed ISCAS'89 benchmark circuits, the proposed synthesis procedure outperforms the technique reported in [6] and extra DVR was 1.6-4x. In general, the proposed procedure can achieve better results as the circuit size gets larger. As the circuit size gets larger, the number of scan cells increases and the length of each scan chain becomes longer for a given compression ratio. Since the maximum number of cone inputs K is limited therefore having longer scan chains will reduce the probability of conflicts. As a result, we are able to achieve higher compression ratio with lower processing and area overheads.

					Regular	Illinois scan			Proposed synthesis procedure				
Circuit	N	M	L	K	ATPG Patterns	Broad. Pat.	Serial Pat.	DVR	Patterns	Conf.	DVR	ΔFC #faults	ATPG overhead
S13207	42	2	21	212	145	177	45	2.71	243	3	12.53	0	2.61
	80	2	11		148	184	58	2.36	296	5	20.00	0	8.02
S15850	42	2	20	183	104	213	32	2.47	319	5	6.85	0	8.33
	77	2	10		100	184	46	1.97	448	12	8.59	0	56.89
S38417	67	2	27	99	139	405	26	3.65	491	4	9.48	0	3.92
	129	2	14		136	542	40	2.81	781	5	11.23	0	18.16
S38584	74	2	25	147	232	322	59	3.43	477	5	18.00	0	6.86
	139	2	13		229	348	93	2.34	636	8	25.02	0	29.51
Circuit A	130	2	77	432	977	1474	117	6.99	1545	3	41.10	32	1.52
	252	2	39		989	1809	232	4.01	2205	3	56.51	39	11.01
	487	4	20		970	1865	207	4.36	2264	4	52.16	0	10.29
	487	2	20		970	2137	249	3.76	3169	4	74.53	141	43.23
Circuit B	135	2	100	282	711	1293	82	7.03	1455	3	32.98	0	1.14
	263	2	51		754	1652	161	4.34	2011	4*	49.30	-194	6.05
	516	4	26		747	1599	206	3.42	2028	4*	47.52	-361	9.87
	516	2	26		747	2055	228	3.17	2328	4*	82.79	-808	33.45
Circuit C	135	2	583	264	2227	2441	56	24.16	2500	2	60.13	46	0.07
	269	2	277		2325	2636	129	15.65	2773	2	112.77	23	0.25
	537	4	135		2330	2665	350	6.30	3038	2	102.96	25	0.50
	537	2	135		2230	3061	405	5.36	3517	2	170.25	0	2.39

Table 2: Experimental results for test data volume reduction

5. Conclusions

In this paper, an efficient procedure for test data volume reduction based on the shared scan-in architecture and compatibility was described. First, a topological analysis was used during DFT synthesis to minimize the number of potential conflicts. Next, compatibility analysis was applied to derive a minimum number of scan-chain configurations such that all detectable faults were tested in broadcast test mode. By avoiding an application of test patterns in serial test mode a reduction up to 170x was achieved in both test data volume and test application time. This experiment also showed that for circuits with similar characteristics, the reduction increases with the size of the circuit. Therefore, the proposed scheme works best for large circuits where test data volume and test application time are critical.

6. References:

[1] K-J. Lee, J-J. Chen and C-H. Huang, "Using A Single Input to Support Multiple Scan Chains", Dig. Tech. Papers, IEEE/ACM Int. Conf. Computer-Aided Design, 1998, pp.74-78.

[2] I. Hamzaoglu and J. H. Patel, "Reducing Test Application Time for Full Scan Embedded Cores," Proc. IEEE Int. Symp. on Fault Tolerant Computing, 1999, pp.260-267.

[3] A. Pandey and J. H. Patel, "Reconfiguration Technique for Reducing Test Time and Test Data Volume in Illinois Scan Architecture Based Designs," Proc. IEEE VLSI Test Symp., 2002, pp.9-15.

[4] S. Samaranayake, N. Sitchinava, R. Kapur, M. Amin and T. W. Williams, "Dynamic Scan: Driving Down the Cost of Test," IEEE Computer, October 2002, pp.65-70.

[5] S. Hellebrand, J. Rajski, S. Tarnik, S. Venkataraman and B. Courtois, "Build-in Test for Circuit with Scan Based on Reseeding of Multiple-Polynomial Linear Feedback Shift Registers," IEEE Trans. on Computers, vol. C-44, No.2, 1995, pp.223-233.

[6] I. Bayraktaroglu and A. Orailoglu, "Test Volume and Application Time Reduction Through Scan Chain Concealment," Proc. ACM/IEEE Design Automation Conf., 2001, pp.151-155.

[7] B. Koenemann, C. Barnhart, B. Keller, T. Snethen, O. Farnsworth, D. Wheater, "A SmartBIST Varian with Guaranteed Encoding", in Proc. Asian Test Symp., 2001, pp.325-330.

[8] C. Barnhart, V. Brunkhorst, F. Distler, O. Farnsworth, A. Ferko, B. Keller, D. Scott, B. Koenemann, T. Onodera, "Extending OPMISR beyond 10x scan test efficiency," IEEE Design & Test of Computers, Vol. 19, Sep-Oct. 2002, pp. 65-73.

[9] J. Rajski, J. Tyszer, M. Kassab, N. Mukerjee, R. Thompson, K. H. Tsai, A. Hertwig, N. Tamarapalli, G. Mrugalski, G. Eide and J. Qian, "Embedded Deterministic Test for Low Cost Manufacturing Test," Proc. Int. Test Conf., 2002, pp. 301-310.

[10] C. Chen and S. K. Gupta, "A Methodology to Design Efficient BIST Test Pattern Generators," Proc. IEEE Int. Test Conf., 1995, pp.814-823.

[11] K. Chakrabarty, B. Murray, J. Liu and M. Zhu, "Test Width Compression for Built-in Self Testing," Proc. IEEE Int. Test Conf., 1997, pp.328-337.

[12] I. Hamzaoglu and J. H. Patel, "Reducing Test Application Time for Built-in Self-Test Test Pattern Generators," Proc. IEEE VLSI Test Symp., 2000, pp.369-375.

[13] E. Gizdarski and H. Fujiwara, "Fault Set Partition for Efficient Width Compression," IEEE Asian Test Symp., 2002, pp.194-199.

[14] Z. Barzilai, J. Savir, G. Markowsky and M. G. Smith, "VLSI Self-Testing Based on Syndrome Techniques," Proc. of the IEEE Int. Test Conf., 1981, pp.102-109.

[15] E. J. McCluskey, "Verification Testing – A Pseudoexhaustive Test Technique," IEEE Trans. on Computers, vol. C-33, No.6, June 1984, pp.541-546.

[16] S. Makar, "A Layout – based Approach for Ordering Scan Chain Flip-Flops," Proc. of the IEEE Int. Test Conf., 1998, pp.341-347.