A Comprehensive Approach to the Partial Scan Problem using Implicit State Enumeration[1]

Priyank Kalla and Maciej Ciesielski
Department of Electrical and Computer Engineering
University of Massachusetts, Amherst, MA-01003

Designated Contact Author:
Maciej Ciesielski
Department of Electrical and Computer Engineering
University of Massachusetts
Amherst, MA-01003-4410
Ph: (413)-545-0401
Fax: (413)-545-1993
Email: CIESIEL@illiac.ecs.umass.edu

Topics: Design for Testability, Sequential Circuit Testing.

ABSTRACT

This paper presents a novel technique and a practical algorithm for the selection of state registers for partial scan. Our model uses implicit techniques for FSM traversal to identify non-controllable state registers. Non-controllability of registers is evaluated by a systematic analysis of the state transitions and the encoding of the underlying FSM, rather than by empirical models or speculative estimates of flip-flop controllability. The result of applying the proposed algorithm (SIMPSON) clearly demonstrates the superiority of our method over conventional state-of-the-art scan register selection techniques.

1 Introduction

Over the years, attempts to automate test generation for sequential circuits have been pursued extensively. These attempts have met with varying levels of success. Automatic test generation (ATG) has generally had difficulties with large sequential circuits. Because of such limitations, various design for testability techniques have become common practice in industry. The *full scan* technique has been developed to simplify the problem of testing a sequential circuit by converting it into a combinational one. This enables the application of combinational test generation algorithms, such as the D-algorithm [20], PODEM [11] and FAN [10], on such circuits. Although the testing problem is simplified, the area and performance of the circuit are adversely affected due to the necessary circuit modifications required to accommodate the complete scan chain. This also results in unacceptable lengths of the resulting tests, due to extensive serial shifting of test patterns and responses.

Partial scan on the other hand, provides a trade-off between the ease of testing and the costs associated with the scan design. However, the key problem in partial scan design is the selection of scan registers. A lot of research has been devoted to define the criteria to guide the selection of the scan memory elements. These techniques are based on i) testability analysis [23], ii) test pattern generation [1] [17], iii) structural analysis [4] [15] [5], and iv) fault oriented and cost analysis based approach [6], etc. All of the above mentioned techniques have met with some measure of success but have their respective limitations.

[1]This work has been supported in part by a grant from NSF under contract No. MIP-9613864

Partial scan methods based on *testability analysis*, use controllability, observability [12], and sequential depth as measures of circuit testability. Flip-flops with poor controllability measures are selected for scan. Trishler [23] describes a method whereby flip-flops which are not easily controllable are included in the incomplete scan path. **Limitations**: the effectiveness of this method depends entirely on testability analysis, which (depending on the heuristics) may not accurately model the problems faced during test generation [2].

In partial scan method based on *structural analysis* [4] [24], the sequential circuit is transformed into a directed graph, whose vertices represent flip-flops, primary inputs and outputs, and whose arcs represent the combinational paths. Heuristics are used to select a minimal set of flip-flops that eliminate the cycles in the graph. The premise behind this approach is based on the assumption that flip-flops in a loop are hard to control and observe. **Limitations**: such techniques operate on the network topology and do not explicitly analyze the behaviour of the sequential circuit. Thus, there is no guarantee that the selected scan elements are the most non-controllable, which may lead us to the selection of scan registers which do not provide sufficiently high fault coverage [14].

In the partial scan methods based on *test generation* [1] [17], tests are first generated for a large number of faults. Then, for each undetectable (or aborted) fault, a set of flip-flops is found, such that making those flip-flops observable and controllable makes the fault detectable. The incomplete scan path then utilizes a minimal subset of memory elements which influenced the easy detection of as many faults as possible. **Limitations:** such techniques incorporate the cost of test generation as well as the cost of calculating minimal sets of registers to scan, and are thus computationally intensive. Also, this technique relies heavily on test generators. Use of an unsophisticated test generator that aborts too many faults may result in some unnecessary scan registers. The *fault oriented* partial scan design approach [6] is also test generator dependent. Structural analysis of the circuit is enhanced by focusing on the untestable and aborted faults. Thus, it also suffers from the above drawbacks.

Considering the fact that the above techniques do not consider the cost of scan design in selecting scan flip-flops, an optimization based approach [5] was presented that formulates the problem of selecting partial scan registers as an *optimization problem*. Based on this idea, a tool called OPUS [5] was developed which is actively used in both academia and industry. However, the testability criteria for selection of scan flip-flops is based on testability heuristics such as the SCOAP controllability/observability measures or on structural parameters of the circuit such as the length of directed cycles etc. Thus, this approach also suffers from the limitations outlined above.

In this paper, we present a new approach to the partial scan problem that thoroughly analyzes the behaviour of the sequential circuit and its state encoding so as to evaluate the non-controllability factors of the state registers. To analyze the behaviour of the underlying FSM of the sequential circuit over the complete state space, we use *implicit techniques* for FSM traversal. We propose an algorithm and present the results which clearly demonstrate the superiority of our technique over the conventional state-of-the-art scan register selection techniques.

The paper is organized as follows. The next section highlights the contribution of this paper and indicates how and why our approach is different from other partial scan approaches. Section 3 describes the motivation behind this work. Section 4 describes a comprehensive method to select scan registers using implicit FSM traversal techniques. In Section 5 a new algorithm, SIMPSON, is proposed and the results are presented in Section 6. To conclude, Section 7 points out possible future work.

2 Contribution of this Research

In this paper, we present a comprehensive approach to analyze the sequential behaviour of a circuit to accurately evaluate the non-controllability of flip-flops in order to make a judicious choice of scan registers. It is well known that testability of a sequential circuit depends on its state transition behaviour and its encoding [18][25][14]. Thus, in order to accurately assess the non-controllability factors of the flip-flops, we need to thoroughly analyze the behaviour and the encoding of the underlying FSM of the circuit. Implicit state enumeration is a technique that can be exploited to analyze the behaviour of the sequential circuit for testing purposes.

Our model to evaluate flip-flop noncontrollability is based on a systematic behavioural analysis of the underlying FSM of a sequential circuit, rather than on empirical models or speculative estimates of non-controllability of flip-flops. Using this method, we accurately evaluate the non-controllability of flip-flops by determining exactly what values can or cannot be stored, or are difficult to store in the state registers. By doing so, we not only target the untestable faults due to state unreachability of the machine, but also the difficult-to-test faults caused by difficult-to-control flip-flops. The experimental results clearly demonstrate the superiority of our approach over conventional state-of-art partial scan design approaches in terms of higher fault coverage by selecting fewer state registers for partial scan.

Implicit state enumeration techniques using BDDs have been researched extensively over the recent years and are well understood [9] [22] [8] [7]. It is to be noted that implicit state enumeration is just used as a tool to analyze the sequential behaviour of the FSM in order to evaluate the non-controllability of flip-flops, and as such is not a subject of this paper.

3 Motivation

Consider the circuit shown in Fig. 1. Using the sequential circuit test generator HITEC [19], the circuit was found to be 77% testable. The untestable faults are depicted in the figure (marked by $\otimes$ and $\times$). In order to improve upon the fault coverage for this circuit, we decided to use partial scan. The program OPUS [5] was used to select partial scan registers. OPUS selected register r_1 for partial scan, resulting in 84% fault coverage (refer to Table 1). The faults marked by $\otimes$ now became testable as a result of making r_1 controllable. Scanning register r_3 gave the same results (84% fault coverage). However,

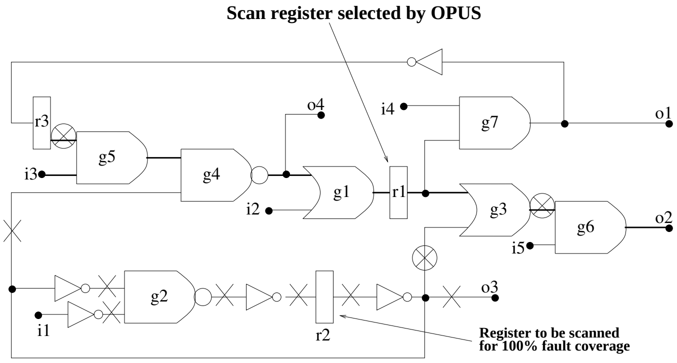


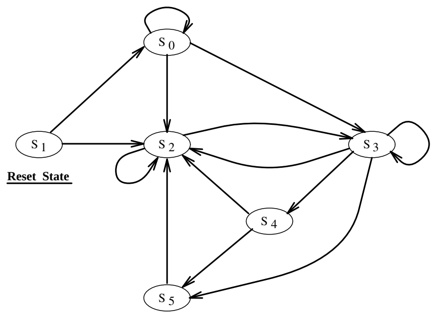
Figure 1: Example circuit depicting the scanned registers.

by selecting r_2 for scan, we were able to achieve 100% fault coverage. The above observation lead us to the following question: How do we know that register r_2 is the best register to scan? To answer this question, let us analyze the state transition behaviour and the encoding of the underlying FSM of this circuit (shown in Fig. 2). We can see from the STG that once the machine is in one of the states determined by the set $\alpha = (S_2, S_3, S_4, S_5)$, it cannot make a transition to any of the states in the set $\beta = (S_0, S_1)$. Once the machine enters the set α, it remains within α, and there is no path in the STG from α to β. Thus, for the encoding in Fig. 2(b), it is not possible to change the value of r_2 from a 0 to

Table 1: Testability data for the example when r_1 and r_2 are scanned.

	OPUS Scan r_1	Scanning r_2
Total # of faults	44	44
# of detected faults	37	44
Percentage fault coverage	84%	100%
# of untestable faults	7	0
# of vectors	13	21

a 1. Clearly, this sort of a behaviour of the underlying FSM of this sequential circuit manifests itself in terms of the non-controllability of register r_2. This, in turn, causes the untestable faults in the circuit.



(a)

State	Code		
	r_1	r_2	r_3
S_0	1	1	0
S_1	0	1	0
S_2	1	0	0
S_3	1	0	1
S_4	0	0	1
S_5	0	0	0

(b)

Figure 2: STG of the example circuit and the corresponding state encoding.

The reason that OPUS failed to identify r_2 as the best register to scan is that its algorithm is based predominantly on the *structural analysis* of the sequential circuit. No information about sequential behaviour of the circuit (state transition graph and the encoding) is used for the selection of scan registers. It becomes clear, however, that the state transition information and the encoding of a sequential circuit are important factors in determining the non-controllability of the state registers. Motivated by the above observation, we investigated how the analysis of the behaviour of the underlying FSM of the circuit could help us to assess its testability measures. Specifically, we could use this analysis to: i) identify non-controllable flip-flops, ii) identify difficult-to-control flip-flops and iii) exploit the information about the unreachable (or illegal) states, in order to select the best possible set of scan flip-flops in a systematic non-greedy fashion.

4 Scan Register Selection Using Implicit State Enumeration

There have been a few attempts to use implicit state enumeration to analyze the circuit behaviour and exploit it for testing purposes. *Cho et. al.* [8] used implicit state enumeration for test generation and redundancy identification. They used implicit state enumeration to perform reachability analysis and used that reachability information during the *state justification* and *state differentiation* phases of test generation. *Long et. al.* [16] also proposed a BDD-based method to enumerate the unreachable states and used this information to identify the sequentially untestable faults. However, none of the above works targeted the partial scan problem.

We shall now present a method to select partial scan registers that uses implicit state enumeration to analyze the circuit's behaviour. By carrying out reachability analysis on the circuit, we not only manage to enumerate the reachable and unreachable states but also pin-point the non-controllable and difficult-to-control flip-flops. We use this flip-flop controllability information to correctly target the registers to be scanned for partial scan design.

4.1 Non-controllable Registers: Missing Transitions

Consider the state transition diagram of our example circuit shown in Fig. 2(a). Starting from the initial state $S_1 = \langle 010 \rangle$, reachability analysis of this machine leads us to the following state traversal:

$$(S_1 : 010) \rightarrow (S_0 : 110, S_2 : 100) \rightarrow (S_3 : 101) \rightarrow (S_4 : 001, S_5 : 000).$$

That is, from S_1, the directly reachable states in one step are S_0 and S_2. From S_0 and S_2, the directly reachable state in one step is S_3, and finally S_4 and S_5, at which point the entire reachable state space has been explored. Let us now examine the FSM traversal trace for register r_2.

$$(S_1 : r_2 = 1) \rightarrow (S_0 : r_2 = 1, S_2 : r_2 = 0) \rightarrow (S_3 : r_2 = 0) \rightarrow (S_4 : r_2 = 0, S_5 : r_2 = 0).$$

Notice that register r_2 can change its value from 1 to 0; *i.e.* it is possible to get a falling transition $(1 \rightarrow 0)$ at the output of r_2. However, a rising transition $(0 \rightarrow 1)$ is missing. In other words, once r_2 gets a value 0, it can never get a value 1. Thus, register r_2 is *unsettable* to a value 1 from a value 0. If certain registers cannot make some transition, then it may not be possible for a test generator to justify the values in the registers during its state justification phase. This, in turn, may render some faults sequentially untestable. Such registers are surely good candidates for scan.

4.2 Difficult-to-Set Flip-Flops

Hartanto et. al. [13] had suggested that identifying the states that are difficult to traverse with test generation tools can significantly speed up test generation for sequential circuits. They proposed a method to identify those flip-flops which were difficult to control. They defined the difficult-to-set flip-flops as follows:

Definition 4.1 *A state element s_d in a sequential machine M is* **difficult-to-set** *to a value v if a test generator, under a specified time and backtrack limit, does not find an input sequence that can bring the machine M from its fully unspecified initial state (consisting of all unknown values in the flip-flops and corresponding to the entire state space) to a state where the value of s_d is v.*

It was indicated that the method to identify difficult-to-set flip-flops was dependent on the test generator used. Also, to identify these difficult-to-set flip-flops they had to modify the circuit by creating a primary output at each flip-flop. The difficult-to-set flip-flops were identified by running a deterministic test pattern generator on the modified circuit and observing the values at the output of each flip-flop.

We now present a method to identify such difficult-to-set flip-flops *without any circuit modifications.* Also, our model to identify difficult-to-set flip-flops does not depend on any test generator. During the implicit forward traversal of a state machine, we record for each register, the largest number of states in a sequence of traversal trace, for which the flip-flops do not change their values. In other words, for each register, we record the length of the longest sequence of 0s and 1s (whichever is greater), which indicates the difficulty in setting a flip-flop to a particular value. To find such difficult-to-set flips, we define a term, *degree of unsettability* of a flip-flop.

Definition 4.2 *The* **degree of unsettability** *of a flip-flop is defined as the length of the longest sequence of states in the implicit traversal trace of an FSM, for which a flip-flop does not change its value.*

Scanning such difficult-to-set flip-flops, identified by their degree of unsettability, would help in detecting the difficult-to-test faults. The reason for this can be explained as follows: If a test generator has to justify a value in a register, say a value of 1, and it encounters a backtrack path of a long sequence of 0s, then it may have to backtrack many time frames in search for a value 1. In doing so, it may abort such faults and classify them as difficult-to-test, which may lead to a reduced fault coverage.

4.3 Sequentially Untestable Faults: Targeting the Illegal States

Knowledge of state space is known to be quite useful in causing early backtracks in test generation. Test generators often spend a significant amount of time on undetectable faults as they eventually have to backtrack a large subset of the state space in order to prove that the values in the registers can not be justified due to the unreachable states. A powerful technique for proving the undetectability of the faults is the identification of illegal states. Formal methods [9] [22] [8], and other recent approaches based on BDDs [16] are widely used to identify illegal states. After computing the reachability information using implicit state enumeration on an FSM, all the reachable states are stored implicitly in a BDD. Complementing this BDD results in the set of all the unreachable states.

We shall now use the information on the unreachable states to target the selection of scan flip-flops. Fig. 3 enumerates all the unreachable states of an MCNC benchmark example *ex*1. All these states are stored implicitly using a BDD which represents the characteristic function of this set of unreachable states.

R1	R2	R3	R4	R5
1	0	0	0	0
1	0	0	1	0
1	0	0	1	1
1	0	1	0	0
1	0	1	0	1
1	0	1	1	0
1	0	1	1	1
1	1	0	1	0
1	1	1	0	0
1	1	1	0	1
1	1	1	1	0

Figure 3: Illegal states of a benchmark circuit (*ex*1).

It is clear from the list of all the unreachable states that register R_1 would be a good candidate for scan. This is because in all the unreachable states, R_1 takes the value 1. Hence, in the reachable state set, it would be difficult to set R_1 to 1. Let us denote the characteristic function of the set of unreachable states by $F_{illegal}$. In this case, $F_{illegal}$ is *unate* in variable R_1. Now the problem of identifying the non-controllable registers from the illegal state set reduces to identifying that state variable over which $F_{illegal}$ is unate.

However, no claims can be made about the unateness of the characteristic functions of the unreachable state set of an FSM in general. Some functions may not be unate in any of its variables, some may be unate in all its variables. Thus, it is necessary for us to define (with some abuse of terminology) the *degree of unateness* in order to measure the non-controllability of the registers.

Definition 4.3 *Let $F_{illegal}$ represent the characteristic function of the set of all the illegal states of an FSM. Let r_i be a variable in the support of $F_{illegal}$. The absolute value of the difference between the number of zeros and the number of ones that a variable r_i can take in the domain of $F_{illegal}$ is defined as the* **degree of unateness** *of the variable r_i.*

Using the above measure of non-controllability, we can select the register for partial scan that has the highest degree of unateness. However, if two or more registers have the same measure of the degree of unateness, they should then be sorted by their degree of unsettability (as defined in the previous section).

5 The SIMPSON algorithm

We now present an algorithm, SIMPSON (Scan register selection using IMPlicit State enumeratiON), that uses implicit state enumeration to analyze the behaviour of the sequential circuit in order to evaluate the non-controllability of the memory elements of the circuit.

Algorithm 5.1 *SIMPSON*

```
begin
  while (Implicit State Enumeration) do
  begin
    for (each state variable) do
    begin
      examine rising and falling transitions;
      record the degree of unsettability;
    end
  end
  for(each state variable) do
  begin
    if (missing transition)
    begin
      scan state variable;
    end
  end
  Illegal States = complement bdd (reachable states);
  for (each state variable) do
  begin
     compute degree of unateness;
  end
  order state variables (degree of unateness, degree of unsettability);
  output state variables interactively;
end
```

The algorithm proceeds as follows. Using symbolic image computation, the FSM is traversed implicitly. During the FSM traversal, both rising and falling transitions on each register are recorded. If a register is found to be missing a transition, that register is selected for scan. Also, during the FSM traversal the degree of unsettability (length of the longest sequence of 0s or 1s, whichever is greater) for each memory element is recorded. After all the reachable states are enumerated, the unreachable states are computed. From this set of unreachable states, the state variables are now sorted in terms of the decreasing order of their degree of unateness. If the degree of unateness of two or more registers is the same, they are sorted in decreasing order of their degree of unsettability. Thus, the algorithm lists the memory elements of a sequential circuit in decreasing order of the degree of their non-controllability with the most non-controllable memory element at the top of the list.

The above algorithm was programmed within the VIS [3] toolset. VIS provides a robust platform for performing reachability analysis using symbolic image computations. The sets of states and transition relations of the FSM are stored in memory using ROBDDs. The CUDD [21] package was used for storage and manipulation of sets of states, Boolean functions, and relations. Extensive experiments on a set of MCNC and ISCAS'89 benchmarks were carried out using SIMPSON. The results depicted in Tab. 2 are compared with OPUS.

Table 2: Experimental results: Applying the SIMPSON Algorithm.

Circuit	Total # Registers	OPUS Selected Registers	OPUS Fault Cov.	OPUS # of Vects.	SIMPSON Selected Registers.	SIMPSON Fault Cov.	SIMPSON # of Vects.
bbsse	6	3	99.66%	69	1	99.66%	131
false	3	1	85%	10	1	100%	21
sse	6	3	99.66%	69	1	98.7%	142
s386	4	3	100%	82	1	100%	157
ex1	5	4	100%	152	3	100%	304
s820	5	1	97%	542	1	100%	894
rie	5	3	100%	171	2	100%	89
planet	7	5	100%	252	4	100%	64
s1488	6	3	96.5%	1033	3	100%	455
s1494	6	2	99.88%	472	2	99.8%	472
s510	6	5	100%	392	3	100%	480
s344	15	1	94.83%	85	1	97.76%	74
s420	16	0	98.3%	36	0	98.3%	36
s641	19	7	100%	172	3	99.37%	236
s713	19	5	91%	203	3	96%	257
s838	32	-	–	–	-	–	–
imec10	8	6	100%	350	5	100%	716

6 Analyzing the Results

Results depicted in Tab. 2 are impressive. For almost all examples, SIMPSON suggests a better set of registers to scan than OPUS as by scanning fewer registers than OPUS, higher fault coverage is achieved. For benchmarks *false, s1488, s344* and *s820*, OPUS and SIMPSON select the same number of registers for scan. However, SIMPSON selects different registers than OPUS and provides higher fault coverage. For benchmark *s1494*, OPUS and SIMPSON select the same registers for scan and hence their testability statistics are identical. The benchmark *s420* is an easily testable machine and SIMPSON indicates that all the flip-flops are controllable. OPUS also doesn't select any register for scan in this example. For benchmark *s838*, VIS was unable to completely traverse the FSM because of computer memory limitations. The set of reachable states was too large to be compactly represented by a monolithic BDD. In all other cases SIMPSON selects fewer registers for scan than OPUS and provides higher fault coverage. For none of these benchmarks, did SIMPSON produce worse results than OPUS. The results clearly indicate the superiority of our approach over the optimization-based approach used by OPUS.

7 Conclusions and Future Work

The approach used by SIMPSON is still quite greedy. It does not take into account correlations between latches. By scanning a register, it may become possible for us to indirectly control other registers. Scanning such indirectly controllable registers would be unnecessary. Thus, a straight-forward extension to SIMPSON would be the incorporation of a technique that analyzes latch correlations. *Cho et. al.* [7] suggested a model to evaluate latch dependencies, latch affinities and latch correlations. They used it to decompose the complete state space of a huge FSM into interactive FSMs so that implicit state enumeration could now be carried out on decomposed FSMs of relatively smaller size. Such a model could be readily incorporated within SIMPSON. While selecting a set of scan registers, correlations of all the registers with respect to a pre-selected scan element could be analyzed to select the next best candidate for partial scan.

Also, by scanning a register, some of the unreachable states will become reachable. Subsequently, the size of the unreachable state set should shrink. Thus, after selecting a register for scan, we could recompute the set of reachable and unreachable states and apply the algorithm again. This would help us in selecting those registers that can be indirectly controlled by scanning other registers in a more systematic and non-greedy fashion.

For very large circuits, it becomes infeasible to represent and manipulate the set of all the reachable and unreachable states using BDDs. For such large circuits, approximate reachability analysis [7] could be carried out to analyze the behaviour of the underlying sequential machine. We are currently in the process of extending SIMPSON to incorporate approximate reachability analysis as well as to analyze latch correlations.

Acknowledgements

The authors would like to thank Professor Prem Menon of University of Massachusetts, Amherst, for fruitful discussions and valued suggestions throughout the course of this project.

References

[1] Agarwal, V. D., Cheng, K. T., Johnson, D. D., and Lin, T. "A complete solution to the partial scan problem". *Proceedings of the International Test Conference,* 1987, pp. 44–51.

[2] Agarwal, V. D., and Mercer, M. R. "Testability measures - what do they tell us?". *Proceedings of the International Test Conference,* 1982, pp. 391–396.

[3] Brayton, R. K., Hachtel, G. D., Sangiovanni-Vencentelli, A., Somenzi, F., Aziz, A., Cheng, S.-T., Edwards, S., Khatri, S., Kukimoto, Y., Pardo, A., Qadeer, S., Ranjan, R., Sarwary, S., Shiple, S., S. G., and Villa, T. "Vis: A system for verification and synthesis". *Proceedings of the Computer Aided Verification Conference,* 1996.

[4] Cheng, K. T., and Agarwal, V. D. "An economical scan design for sequential logic test generation". *Proceedings of the 19th International Symposium on Fault-Tolerant Computing,* 89, pp. 28–35.

[5] Chikermane, V., and Patel, J. H. "An optimization based approach to the partial scan design problem". *Proceedings of the International Test Conference,* 1990, pp. 377–386.

[6] Chikermane, V., and Patel, J. H. "A fault oriented partial scan design approach". *Proceedings of the International Test Conference,* 91, pp. 400–404.

[7] Cho, H., Hachtel, G., Macii, E., Poncino, M., and Somenzi, F. "Automatic state space decomposition for approximate fsm traversal based on circuit analysis". *IEEE Transactions on Computer Aided Design,* 1996, pp. 1451–1464.

[8] Cho, H., Hachtel, G., and Somenzi, F. "Redundancy identification/removal and test generation for sequential circuits using implicit state enumeration". *IEEE Transactions on Computer Aided Design,* 93, pp. 935–945.

[9] Coudert, O., and Madre, J. C. "A unified framework for the formal verification of sequential circuits". *Proceedings of the International Conference on Computer Aided Design,* 1990, pp. 126–129.

[10] Fujiwura, H., and Shimono, T. "On acceleration of test generation algorithms". *IEEE Transactions on Computers,* 1983, pp. 1137–1144.

[11] Goel, P. "An implicit enumeration algorithm to generate tests for combinational logic circuits". *IEEE Transactions on Computers,* 1981, pp. 215–222.

[12] Goldstien, L. H. "Controllability/observability analysis of digital circuits". *IEEE Transactions on Circuits and Systems,* 1979, pp. 685–693.

[13] Hartanto, I., Boppana, V., and Fuchs, W. K. "Identification of unsettable flip-flops for partial scan and faster atpg". *Proceedings of the International Conference on Computer Aided Design,* 1996.

[14] Kalla, P., and Ciesielski, M. J. "Testability of sequential circuits with multi-cycle false paths". *Proceedings of the 15th IEEE VLSI Test Sympsium,* 1997, pp. 322–328.

[15] Lee, D. H., and Reddy, S. M. "On determining scan flip-flops in partial scan design". *Proceedings of International Conference on Computer Aided Design,* 1990, pp. 322–325.

[16] Long, D. E., Iyer, M. A., and Abramovici, M. "Identifying sequentially untestable faults using illegal states". *Proceedings of the VLSI Test Symposium,* 1995, pp. 4–11.

[17] Ma, H. K. T., Devadas, S., Newton, A. R., and Sangiovanni-Vincentelli, A. "An incomplete scan design approach to test generation of sequential machines". *Proceedings of the International Test Conference,* 1988, pp. 730–734.

[18] Marchok, T. E., El-Maleh, A., Maly, W., and Rajski, J. "Complexity of sequential atpg". *Proceedings of the Design Automation Conference,* 1995, pp. 252–261.

[19] Niermann, T. M., and Patel, J. H. "Hitec: A test generation package for sequential circuits". *Proceedings of the European Conference on Design Automation,* 1991, pp. 214–218.

[20] Roth, J. P. "Diagnosis of automata failures: A calculus and a method". *IBM Journal of Res. Dev.,* 1966, pp. 278–291.

[21] Somenzi, F. "Colorado decision diagram package". *Computer Programme,* 1997.

[22] Touati, H., Savoj, H., Lin., B., Brayton, R. K., and Sangiovanni-Vincentelli, A. "Implicit state enumeration of finite state machines using bdds". *Proceedings of the International Conference on Computer Aided Design,* 1990, pp. 130–133.

[23] Trishler, E. "Incomplete scan path with an automatic test generation methodology". *Proceedings of the International Test Conference,* 1980, pp. 153–162.

[24] Wunderlich, H. J. "The design of random-testable sequential circuits". *Proceedings of the 19th International Symposium on Fault Tolerant Computing,* 1989, pp. 110–117.

[25] Xiang, D., Venkataraman, S., Fuchs, W. K., and Patel, J. H. "Partial scan design based on circuit state information". *Proceedings of the 33rd Design Automation Conference,* 1996, pp. 807–812.