Partial Scan Using Multi-Hop State Reachability Analysis

Sameer Sharma and Michael S. Hsiao
{sameers, mhsiao}@ece.rutgers.edu
Department of Electrical and Computer Engineering, Rutgers University, Piscataway, NJ

Abstract

Sequential test generators fail to yield tests for some stuck-at-faults because they are unable to reach certain states necessary for exciting/propagating these target faults. Adding scan to the circuit increases reachability of these hard-to-reach and/or previously unreachable states. In this paper, we postulated that fewer scan flip-flops are needed to make these states reachable. The states necessary for detecting the hard-to-detect faults, when reached, will facilitate reaching other hard-to-reach states in one or more hops by the sequential test generator, resulting in significantly higher fault coverage. We collect information on the hard-to-reach, aborted, and easy states in our analysis. Results from our approach have indicated that higher fault coverage can be achieved with significantly fewer scan flip-flops for some circuits.

I Introduction

Partial scan is a DFT technique which involves selecting a subset of flip-flops in the circuit for scan. This approach has several advantages over full-scan which selects all the flip-flops in the circuit for scan. Since the number of flip-flops which have to be modified for scan is less, it involves less area and power overhead. Moreover, fewer scan flip-flops leads to shorter test vectors, resulting in reduction of test application time. A test engineer aims at minimizing the number of flip-flops scanned while maximizing the fault coverage in the resultant circuit, and this problem becomes extremely complex for large circuits. Because circuits with partial-scan are still sequential in nature, our method attempts to maximally utilize the power of sequential test generators. By interacting the test generator with the flip-flop selection program, we ensures that the selection of flip-flops is of maximum advantage to the sequential test generator.

The previous work done for the selection of flip-flops for partial scan is usually classified into 3 major categories: structural-analysis based [2]-[8], testability-analysis based [9]-[12], and test-generator-based [3, 8, 13, 14, 15, 16]. Some of these [3, 8, 16] also use a combination of these approaches. Structural-analysis based techniques represent the circuit as a graph and attempt to remove all possible feedback by scanning flip-flops. However, removal of a minimal vertex set is an NP-complete problem, and moreover, it has been shown that a sequential test generator may not detect all the faults even with all cycles (except self-loops) removed [9]. Testability-cased approaches, on the other hand, are simple in terms of computational complexity, but they usually do not yield good fault coverage for the some of the circuits with more complex structures. In the more recent past, symbolic techniques [18] and implicit state enumeration [17] have been used for selecting the flip-flops for partial scan. In [18], the authors define a new testability measure calculated on the basis of analysis of state transition graph. This is then used to assign weights to the flip-flops. The approach in [17] is based on the traversal of FSM for the circuit to identify the non-controllable and difficult-to-control flip-flops.

Automatic Test Pattern Generation (ATPG)-based techniques seek to utilize the information generated by the test generator to try and detect the aborted faults. This information may be the aborted faults, aborted states, test vectors, etc. These approaches have been the most effective recently since they work in conjunction with the test generator, utilizing the information generated by the ATPG and feeding back the information that is most pertinent. E-STG [16] is one such approach which takes in a list of aborted states which the test generator was unable to justify and tries to make those states reachable by selecting the minimal set of flip-flops. However, since E-STG attempts to make the state reachable within a single transition from a known reachable state, the set of flip-flops selected may not necessarily be minimal and it may not provide the best fault coverage. Another approach called IDROPS [19] uses a combination of testability measures and information given by the ATPG. A new dynamic reachability and observability measure (adopted fault detectabilty fault potential) is introduced and is used as the criteria for flip-flop selection.

As the test generation for a partial-scan circuit is still sequential in nature, there is no reason for us to constrain ourselves to a single-transition solution as in [16]. In our work, we allow the test generator to go from an easy-to-reach state to a hard-to-reach state in multiple hops. We must ensure that the multi-hop sequence is easy for

the ATPG to obtain. In doing so, we can scan fewer flip-flops while keeping ATPG time to a minimum. We collect information on the hard-to-reach, aborted, and easy states in our analysis. Experimental results from our approach have indicated that higher fault coverage can be achieved with significantly fewer scan flip-flops for some circuits.

The remainder of this paper is organized as follows: Section II provides the definitions necessary for this work. Section III explains the key ideas for the multi-hop reachability and the partial-scan algorithm. Experimental results are given in Section IV, and Section V concludes the paper.

II Definitions

In this section we present a few definitions which will be used in the explanation of the algorithm later.

Definition 1: *Easily-reachable states set:* A representative subset of all the states which can be easily and quickly reached by the test generator either by applying random vectors or its own set of guided vectors.

Definition 2: *Hard-to-reach states set :* A representative subset of all the states which were needed to excite/propagate fault(s) but could not be justified by the test generator.

The test generators classify these states as *aborted* and the corresponding faults are termed *aborted faults.* Our algorithm aims at making these states a part of reachable state space, possibly via multiple transitions.

For the rest of the definitions, consider a given circuit C with n flip-flops, H is the set of hard-to-reach states with cardinality h and E is the set of easily reachable states with cardinality[1] e. The state space for such a circuit is depicted in Figure 1. The easy-to-reach and hard-to-reach states are indicated.

Definition 3: *Multi-hop state traversal:* This refers to a situation where we can reach a particular hard-to-reach state from a given state in one or more state transitions.

We can do so by overloading the circuit with a particular state and then try to reach other hard-to-reach states by applying random/guided vectors. This is depicted as the dotted path $(H_2, H_3, H_4, H_5, H_6)$ in Figure 1.

Definition 4: *Scan-eligible flip-flop (SEFF)*: For a particular easy-to-reach state $E_i = (ff_1, ff_2 ...., ff_n)$ and a hard-to-reach state $H_j = (FF_1, FF_2, ..., FF_n)$, where $ff_i, FF_i \in \{0, 1, X\}$ are the values of the flip-flops, flip-flop number k is a scan-eligible flip-flop if $ff_k \neq FF_k$ and $FF_k \neq unknown(X)$.

[1] Cardinality of the set is the number of elements in that set

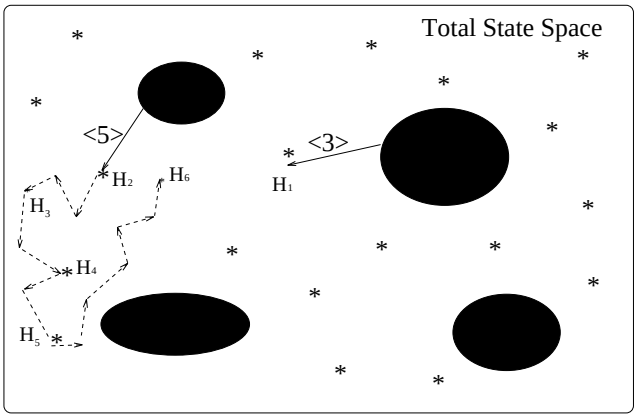

Figure 1: State space for a circuit C.

In the example given in Table 1 for a circuit with 9 flip-flops, flip-flops 1, 2, 4 and 8 are the SEFFs. Please note that even though flip-flops 3 and 7 *may* have different values in the two states, they need not be scanned since the hard-to-reach state H_j does not require a particular value for this flip-flop. This is a measure of the number of flip-flops we will have to scan to make H_j reachable from E_i . It should be intuitive that such a transformation will ensure that H_j can be reached form E_i in a single transition.

Table 1: Selecting SEFFs

FF #	**1**	**2**	3	**4**	5	6	7	**8**	9
H_i	**0**	**1**	X	**0**	0	1	X	**1**	X
E_j	**1**	**0**	0	**X**	0	1	1	**X**	X

Definition 5: *Resemblance Index* $(RI(H_j, S))$: For a particular hard-to-reach state $H_j = (ff_1, ff_2 ...., ff_n)$ and another state $S = (FF_1, FF_2, ..., FF_n)$ where $ff_i, FF_i \in \{0, 1, X\}$ are the values of the flip-flops, $(RI(H_j, S))$ is defined as the total number of flip-flops which have identical values in the two states. Any flip-flop which has value **X** in state S is not counted.

This is a metric of similarity between the two states. Please note that the combination $ff_i = X$ and $FF_i = 0$ *or* 1 is not counted towards the Resemblance Index. For the example given in Table 1, considering E_i as S, $RI(H_j, S) = 2$ (FF # 5 and 6).

Definition 6: *Scan Cost Index (SCI) and Closest Neighbor*: The minimum number of flip-flops we will have to scan to make a hard-to-reach state (say H_i) a part of the reachable state space.

For the circuit C, we calculate SCI_{H_i} for each state $H_i \in H$ as follows

$$SCI_{H_i} = MIN_{\forall E_j \in E}(SEFF(H_i, E_j)) \quad (1)$$

The E_j for which $SEFF(H_i, E_j)$ is minimum is called the *Closest Neighbor* of H_i.

Definition 7: *Gain value*: The ease of reaching other hard-to-reach states from a given state in *one or more hops*. This is calculated for each unreachable state H_j using a multi-hop state analysis algorithm described later. We also calculate the *Normalized Gain (N.F.)* by the following formula.

$$N.F.(H_j) = Gain(H_j)/SCI(H_j) \quad (2)$$

Consider a circuit with 4 flip-flops, 3 easy-to-reach states and 2 hard-to-reach states as shown in Table 2 . On the basis of definitions given in this section, $SEFF(H_1, E_1) = 3$, $SEFF(H_1, E_2) = 4$, $SEFF(H_1, E_3) = 2$, $SEFF(H_2, E_1) = 2$, $SEFF(H_2, E_2) = 1$, $SEFF(H_2, E_3) = 3$. From these values, $Closest\ Neighbor_{H_1} = E_3$, $SCI_{H_1} = 2$, $Closest\ Neighbor_{H_2} = E_2$, $SCI_{H_2} = 1$.

Table 2: Example for a Circuit with 4 flip-flops

E_1	1	X	0	1
E_2	1	0	X	1
E_3	X	X	0	0
H_1	0	1	0	0
H_2	0	0	X	1

III Multi-Hop(MH) scan - Partial Scan Selection Algorithm

Making all necessary states reachable within a single state transition will result in a non-minimal scan flip-flop set for a given target fault coverage. This is because the states we try to make reachable by scanning may not be the best states to scan. In addition, the circuit with such scanned flip-flops may not provide the best fault coverage, as sequential test generation is not fully utilized. This is clearly shown in the results given in section IV. Instead of limiting all hard-to-reach states to be reachable in a single hop, we allow for multi-hop transitions. This way, the test generator can go from an easy-to-reach state to a hard-to-reach state in more than one hop. Again let us use Figure 1 as an example; we can reach states H_3, H_4, H_5 *and* H_6 in multiple hops once we reach state H_2. On the other hand, state H_1 does not offer the same benefit because even if we reach H_1, it does not facilitate the reachability of any other hard-to-reach state. So, even though it may seem that it is more costly to make H_2 reachable by scanning 5 flip-flops as opposed to 3 flip-flops in case of H_1, quite the reverse is true.

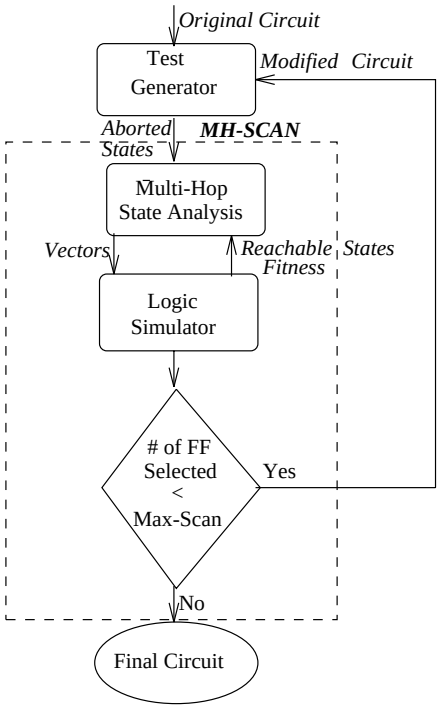

Figure 2: Overall flow for Multi-Hop-Scan Partial Scan selection algorithm

Figure 2 describes the partial-scan algorithm. We start by running a quick pass of the test generator on the original circuit. The test generator used was STRATEGATE [1], a state-of-the-art genetic-based test generator. With a quick pass, information about the aborted states is collected. These states are the ones which the test generator needed to detect the aborted faults. This set of *aborted states* becomes the input to the **Multi-Hop Scan** algorithm and are marked as asterisk (*) in Figure 1. We also form a list of easy-to-reach states indicated by the shaded areas in Figure 1. This can be done by generating random vectors or reading in the vectors generated by STRATEGATE. The algorithm takes in the list of aborted states, easy-to-reach states and vectors. It uses the logic simulator to calculate the *Gain* for all the states. This Gain is used as the criteria to rank all the states. On the basis of this ranking it selects the *best candidate* i.e. the state with the highest Gain is selected as the *best candidate* and it is made reachable by scanning all the SEFFs between this hard-to-reach state and its *Closest Neighbor* in the easy-to-reach state space. Various parameters supplied to **Multi-Hop Scan** are

- Number of vectors to be used for simulation.
- Type of vectors, i.e. random/guided (STRATEGATE) vectors.

- Maximum percentage of FFs to be scanned.
- Maximum number of hops allowed for multi-hop (MAX-HOP). This also depends on the efficacy of the Test Generator i.e. over how many time frames can it usually justify states.

A Multi-hop state analysis and Gain calculation

The procedure for simulation and collecting information on the hard-to-reach states shown in Figure 3.

```
begin
  for each hard-to-reach state Hi
    initialize Gain = 0
      while vectors left
        overload circuit with that state
        simulate to get to state S
          for all hard-to-reach states Hj
            calculate Resemblance Index, RI ()
            Gain = Gain + RI(Hj, S)
        until (Easy-to-reach state reached or
        Hops > MAX-HOP)
end
```

Figure 3: Multi-Hop partial scan algorithm

In this procedure, we simulate the circuit with some vectors after overloading it with some hard-to-reach state. We observed that in many cases, the circuit frequently returns to one of the easy-to-reach states. This is because of the following reason: the circuit naturally returns to easy-to-reach states with random vectors. In the case of overloading the circuit with an illegal state, due to the way synchronous circuits are usually designed, the circuit immediately returns to a legal state which would be easy-to-reach. For example, consider a modulo-5 counter with 3 flip-flops. The legal states are **{000, 001, 010, 011, and 100}**. State **111** is illegal, so when we draw the state transition diagram prior to circuit synthesis, we need to make sure that if the circuit starts in the state **111**, it should immediately transition to one of the legal state in the very next transition. After calculating the Gain function for each hard-to-reach state, we normalize it with respect to the SCI for that state.

B Ranking and scan phase

After we have finished the state analysis and ranked the states, we pick the state with the highest Gain value. This is called the Best Candidate (BC). We make it reachable by scanning all the SEFFs for the pair ($BC, Closest\ Neighbor(BC)$). This ensures that BC can be reached in one transition from $Closest\ Neighbor(BC)$). Consider the example given in Table 2. Assume we overload the circuit with states H_1 and H_2 and simulate with one vector to reach states **A(X0X1)** and **B(0001)** respectively. On the basis of the definitions given, $RI(H_2, A)$= 2. Also, $RI(H_1, B)$ = 2, Therefore $Gain(H_1)$=2, $NF(H_1)$=1.0 and $Gain(H_2)$=2 and $NF(H_2)$=2. Hence H_2 becomes the $Best\ Candidate$ and E_2 becomes $Closest\ Neighbor(BC)$. We will scan flip-flop #1 which makes H_2 reachable from E_2 in one hop.

Once the test generator reaches this state it will also be able to reach many other hard-to-reach states. Because this state was *close* to many other hard-to-reach states, scanning the more difficult to control/observe flip-flops in this state will give the test generator more leverage when it tries to generate tests for the circuit. The iterative nature of the algorithm ensures that this benefit is incrementally maximized.

Table 3: STRATEGATE Results

Circuit (# FFs)	Total Faults	STRATEGATE	
		Det	Vec
s298(14)	308	265	306
s344(15)	342	329	86
s400(21)	426	384	2424
s444(21)	474	424	1945
s526(21)	555	454	2642
s641(91)	467	404	166
s820(5)	850	814	590
s832(5)	870	818	701
s1423(74)	1515	1414	3943

Det: number of faults detected Vec: test set length

IV Experimental Results

In this section we present the results for the algorithm. The program was run on ISCAS 89 benchmark circuits. We compare our results with the results obtained by E-STG [16] and IDROPS [19]. E-STG is based on the concept of making the aborted states reachable by looking for single transitions in the state transition graph; thus it does not make full use of the fact that the modified circuit will also be given to a sequential test generator and so the aborted states can be reached from the easy-to-reach states in multiple transitions.

We first show the fault coverage STRATEGATE gives on various circuits with no scan in Table 3. This table gives the total number of faults, detected faults, and the test set size for ISCAS 89 sequential benchmark circuits.

In the table 4 and 5 we present the results for Multi-Hop Scan and compare it with the results of E-STG [16] method and IDROPS [19] respectively.

Table 4: Results for ISCAS 89 benchmark circuits and comparison with E-STG [16]

Ckt.	% Scan	E-STG							MH-Scan						
		Det	Unt	Abt	Time	Vec	FC	TE	Det	Unt	Abt	Time	Vec	FC	TE
s298	10	**302**	6	0	23	653	98.05	100	292	15	1	23.5	386	94.8	99.65
s298	20	304	4	0	6	398	98.7	100	**305**	3	0	1.51	111	99.02	100
s344	10	328	9	5	177	95	95.9	98.48	**337**	4	1	113	263	98.54	99.7
s400	10	386	14	24	642	398	91.03	94.15	**402**	21	5	269	576	93.9	98.8
s400	20	**414**	10	0	42	321	97.6	100	402	22	4	180	282	93.9	99.01
s420	10	44	141	270	5967	14	9.7	14.01	**68**	202	185	3524	12	14.9	26.9
s420	20	**96**	141	218	4858	28	21.1	30.6	**96**	141	218	1612	14	21.1	26.9
s526	10	326	20	209	4643	241	58.7	60.9	**520**	14	21	454	542	93.7	96.1
s526	20	517	22	16	996	1155	93.1	97	**525**	12	18	377	429	94.6	96.7
s641	10	408	59	0	41	205	72	100	**439**	28	0	52	173	77.4	100
s641	20	**464**	3	0	12	255	81.8	100	**464**	3	0	51	244	81.8	100
s820	20	**850**	0	0	98	920	100	100	**850**	0	0	19	461	100	100
s832	20	855	14	1	98	914	98.3	98.7	**856**	14	0	45	428	98.4	100
s1423	10	943	11	561	13972	235	62.2	62.7	**1442**	15	58	4473	5586	95.2	96.1
s1423	20	1127	10	378	8626	254	74.4	74.9	**1452**	15	48	4742	4187	95.8	96.8

Det: # faults detected Unt: # untestable (redundant) faults Abt: # faults aborted
Vec: # test set length Time: measured in seconds (for test generation on the circuit with scan)
FC: Fault coverage as % TE: Test Efficiency as %

For these results, the program was run on Sun UltraSparc-1 workstation. Our method provides the option of selecting the flip-flops in "one-shot" or in an incremental way. It was empirically found that the results were better with iterative selection. Fault coverage is calculated as percentage of total faults which are detected. Testing efficiency on the other hand is calculated as percentage of detectable faults (total faults - redundant faults) that are detected. The higher fault coverage / number of detected faults are indicated in bold. In cases where both methods achieve the same coverage, both values are shown in bold.

In table 4, we have given the results with 10% and 20% scan for most of the circuits. The number of detected, untestable, and aborted faults are reported for each circuit. ATPG test generation times in seconds and the number of vectors generated are also reported. All these results were obtained by using 1000 random vectors for obtaining the set of easy-to-reach states.

In almost all the cases, circuits obtained by Multi-Hop Scan show higher fault coverage than E-STG. Also, the testing efficiency is significantly higher. In the cases where the number of detected faults is less, a significant portion of undetected faults are redundant (Unt) faults. The benefit of Multi-Hop algorithm becomes more obvious as we deal with circuits with more flip-flops. This gives more leverage to the algorithm and it is able to select the flip-flops in a better manner. This is clearly demonstrated in s526 and s1423 where the number of faults detected by Multi-Hop Scan is much higher. For s344, Multi-Hop Scan detects more faults (337) with 10 % scan than E-STG detects(331) [16] with 20 % scan. Both s820 and s832 have 6 flip-flops each, so it there is no difference in 10% and 20 % scan. Even though E-STG matches Multi-Hop Scan in s641 and s420 with 20% scan, it detects lesser faults on the same circuit with 10% scan. Thus, it is clearly indicated that the set of flip-flops selected by Multi-Hop Scan is minimal.

In table 5, we compare the fault coverage and testing efficiency for Multi-Hop Scan and IDROPS [19] for various circuits. As shown, Multi-hop scan matches the performance of IDROPS for most of the circuits. Multi-Hop reaches a higher fault coverage with comparatively less scan. Both in s444 and s526, it gives higher fault coverage by scanning 2 flip-flops than IDROPS does by scanning 3 flip-flops. In s1423, IDROPS has to scan 9 flip-flops to give almost the same coverage which Multi-Hop gives with 7 scanned flip-flops.

V Conclusions

We have presented a new algorithm for selecting the flip-flops for scan based on a multi-hop analysis of hard-to-reach states from a subset of the reachable state space. We have demonstrated that many hard-to-reach states necessary for detecting the aborted faults can be reached with fewer scan flip-flops. Since our approach works on interactively with the test generator, the results are of maximal utility to the test generator. As we don't enumerate the set of all reachable states, this approach is tractable for large circuits. We are currently working on developing heuristics to make the algorithm faster for large circuits. Our method also allows for speeding up the test generation process by feeding back the information about *how*

Table 5: Comparison with results for IDROPS [19]

Ckt.	IDROPS			MH-Scan		
	Scan	FC	Time	Scan	FC	Time
s420	-	-	-	2	8.4	4218
	3	20.9	1587	3	20.9	4517
	4	22.9	1454	4	27.03	4813
s444	-	-	-	2	93.5	702
	3	93.2	67.9	3	93.7	362
	4	94.7	54.5	4	93.7	511
s526	-	-	-	2	93.7	1550
	3	87.2	225	3	93.9	1490
	4	94.2	242	4	94.6	1414
s1423	9	95.3	685	7	95.2	5521
	15	95.8	646	15	95.8	4945

Scan: # flip-flop scanned FC: Fault coverage as %
Time: measured in seconds (includes test generation time)

precisely to reach those hard-to-reach states. Results in Table 5 show that our multi-hop scan method surpasses those in previously proposed approaches; equal or greater fault coverage is obtained with fewer scan flip-flops.

References

[1] M. S. Hsiao, E. M. Rudnick, and J. H. Patel, "Sequential circuit test generation using dynamic state traversal," *Proc. European Conf. Design Automation*, pp. 22–28, 1997.

[2] K.-T. Cheng and V. D. Agrawal, "A partial scan method for sequential circuits with feedback," *IEEE Trans. Computers*, vol. 39, pp. 544–548, 1990.

[3] V. Chickermane and J. H. Patel, "A fault oriented partial scan design approach," *Proc. Int. Conf. Computer-Aided Design*, pp. 400–403, 1991.

[4] S. E. Tai and D. Bhattacharya, "A three-stage partial scan design method to ease ATPG," *Journal of Electronic Testing*, vol. 7, pp. 95–104, Aug./Oct. 1995.

[5] S. T. Chakradhar, A. Balakrishman, and V. D. Agrawal, "An exact algorithm for selecting partial scan flip-flops," *Proc. Design Automation Conf.*, pp. 81–86, 1994.

[6] P. Ashar and S. Malik, "Implicit computation of minimum-cost feedback-vertex sets for partial scan and other applications," *Proc. Design Automation Conf.*, 77–80, 1994.

[7] T. Orensten, Z. Kohavi, and I. Pomeranz, "An optimal algorithm for cycle breaking in directed graphs," *Journal of Electronic Testing*, vol. 7, pp. 71–82, Aug./Oct. 1995.

[8] D. Xiang and J. H. Patel, "A global algorithm for the partial scan design problem using circuit state information," *Proc. Int. Test Conf.*, pp. 548–557, 1996.

[9] V. Chickermane and J. H. Patel, "An optimization based approach to the partial scan design problem," *Proc. Int. Test Conf.*, pp. 377–386, 1990.

[10] K. S. Kim and C. R. Kime, "Partial scan flip-flop selection by use of empirical testability," *Journal of Electronic Testing*, vol. 7, pp. 47–60, Aug./Oct. 1995.

[11] P. S. Parikh and M. Abramovici, "A cost based approach to partial scan design," *Proc. Design Automation Conf.*, pp. 255–259, 1993.

[12] P. S. Parikh and M. Abramovici, "Testability-based partial scan analysis," *Journal of Electronic Testing*, vol. 7, pp. 61–70, Aug./Oct. 1995.

[13] F. Corno, P. Prinetto, M. Rebaudengo, and M. Sonza Reorda, "Partial scan flip flop selection for simulation-based sequential ATPG's," *Proc. Int. Test Conf.*, pp. 558–564, 1996.

[14] H. K. T. Ma, S. Devadas, A. R. Newton, and A. Sangiovanni-Vincentelli, "An incomplete scan design approach to test generation for sequential machines," *Proc. Int. Test Conf.*, pp. 730–734, 1988.

[15] I. Park, D. S. Ha, and G. Sim, "A new method for partial scan design based on propagation and justification requirements of faults," *Proc. Int. Test Conf.*, pp. 413–422, 1995.

[16] V. Boppana and W. K. Fuchs, "Partial scan design based on state transition modeling," *Proc. Int. Test Conf.*, pp. 538–547, 1996.

[17] P. Kalla and M. Ciesielski, " A comprehensive approach to the partial scan problem using implicit state enumeration ," *Proc. Int. Test Conf.* pp. 651-657,1998 .

[18] F. Corno, P.Prinetto, M. Sonza Reorda and M. Violante, "Exploiting symbolic techniques for partial scan flip flop selection," *Proc. Design, Automation and Test in Europe Conf.Proc. Design,* pp. 670-677, 1998.

[19] Michael S. Hsiao, Gurjeet S. Saund, Elizabeth M. Rudnick, and Janak H. Patel, "Partial scan selection based on dynamic reachability and observability information ," *Proc. VLSI Design,* pp. 174-180, 1998.