Functional Test Generation for Non-Scan Sequential Circuits

M.K. Srinivas
Indian Institute of Science
CAD Lab / SERC
Bangalore 560012, India

James Jacob
Indian Institute of Science
Department of ECE
Bangalore 560012, India

Vishwani D. Agrawal
AT&T Bell Laboratories
600 Mountain Avenue
Murray Hill, NJ 07974, USA

Abstract—The feasibility of generating high quality functional test vectors for sequential circuits using the Growth (G) and Disappearance (D) fault model has been demonstrated earlier. In this paper we provide a theoretical validation of the G and D fault model by proving the ability of this model to guarantee complete stuck fault coverage in combinational and sequential circuits synthesized employing algebraic transformations. We also provide experimental results on a wide range of synthesized FSMs. A comparison with a state-of-the-art gate level ATPG tool demonstrates the efficiency and limitation of the functional approach.

I. INTRODUCTION

The *growth* (G) and *disappearance* (D) faults in the combinational function of a circuit form a subset of the faults normally modeled in the programmable logic array (PLA) implementation [12]. It is known that the tests for G and D faults cover all stuck faults in any two level implementation of the combinational logic [8]. For certain synthesis styles [7, 11], these tests will also cover *all* single stuck faults in the multi-level combinational circuit.

The main contributions of this paper are the theoretical validation of the G and D fault model and the experimental results on a broad range of synthesized circuits. We model the combinational logic at the functional level by its personality matrix (PM) and employ an efficient cube based test generation algorithm to obtain test sequences for G and D faults in the FSM. A time frame expansion technique is used [1]. The functional test sequences provide a high coverage of stuck faults in the algebraically synthesized multi-level circuit. Our recent research [8, 13] has shown the feasibility of this approach.

There are cases like the arithmetic or parity functions where the number of cubes in the two level sum of products form is exponential in the number of primary inputs. Our method, presently will not handle these cases efficiently. However, the technique can be extended to large gate level combinational and sequential circuits if we partition them into interconnection of moderately sized functional blocks. Such an approach is under investigation.

II. FAULT COVERAGE THEOREMS

The primary usefulness of G and D faults stems from their ability to model stuck faults in irredundant two-level circuits and a certain class of multi-level combinational circuits as shown by the following results available in the literature:

(i) All single stuck faults in an irredundant two-level single or multiple output circuit are detected by the tests for G and D faults of the equivalent PLA, provided the tests set each primary output to 0 at least once [9].

(ii) In an irredundant two-level circuit in which all single stuck faults are detectable, the test vectors for all single stuck faults will also detect all multiple stuck faults, provided we can find an ordering $z_1 \ldots z_q$ among the q output functions such that all stuck faults in the subcircuit feeding output z_j are detected via one or more outputs $z_1 \ldots z_i$ $(1 \leq i \leq j \leq q)$ [9]. This result, together with result (i) implies that the test vectors that detect all single G and D faults in the equivalent PLA, will detect all multiple stuck faults in the irredundant two-level multiple output circuit provided the vectors conform to an output ordering constraint. Such a constraint can be easily satisfied by any test generator.

(iii) If we only use algebraic factorization of the minimized irredundant two-level single output Boolean function to realize a multi-level circuit, then all multiple faults in the multi-level circuit will be testable [7]. Also, the test vectors that detect all single stuck faults in the irredundant two-level single output circuit will cover all single and multiple stuck faults in the synthesized multi-level circuit.

(iv) Testability preserving transformations consisting of algebraic factorization, applied to any prime and irredundant combinational circuit preserve single fault testability [11]. This means that the tests for all single stuck faults in the original circuit will cover all single stuck faults in the synthesized (transformed) circuit.

Our main results follow next.

Theorem 1: *The test sequences for the single G*

1063-9667/95 $4.00 © 1995 IEEE

and D faults of a single output minimized personality matrix will cover all single (and multiple) stuck faults in the multi-level combinational circuit that is synthesized using testability preserving transformations.

Proof: A single output minimized personality matrix (PM) is prime and irredundant with respect to every output and there is no product term sharing for any output. Let N1 be a single output minimized PM, N2 the equivalent two-level AND-OR circuit and let N3 be synthesized from N1 using only algebraic factorization. Algebraic factorization ensures that N3 is also prime and irredundant with respect to every output [11]. The test set for single G and D faults in N1 will cover all multiple G and D faults in N1 and all single and multiple stuck faults in N2 as the output ordering criteria according to result (ii) is automatically satisfied. Now every single and multiple stuck fault in N3 has an equivalent single or multiple stuck fault in N2 according to results (iii) and (iv). Therefore, the test vectors derived for single G and D faults of N1 will cover all single and multiple stuck faults in the multi-level combinational circuit N3, synthesized using testability preserving transformations. ■

Theorem 1 is not directly applicable to sequential circuits or finite state machines (FSM) as the results (i) through (iv) of the previous section are valid only for combinational circuits. It should be noted that the tests for G and D faults of the PM of an FSM do not guarantee a complete coverage of all single stuck faults on those primary inputs (PI) and present state (PS) stem lines that have fanouts reconverging only at the next state (NS) lines, and the multiple stuck faults in the equivalent two-level (AND-OR) FSM. This is due to the fact that these PI and PS stem faults and the multiple stuck faults can be masked in the reconvergence structure across the time frames in an Iterative Logic Array (ILA) model of the FSM. If, however, the G and D fault tests of a single output minimized PM of an FSM cover all single stuck faults of an equivalent two-level (AND-OR) FSM, then these tests will also cover all single stuck faults of the multi-level FSM whose combinational portion is synthesized using testability preserving transformations. For complete single stuck fault coverage of the synthesized FSM, in addition to the single G and D faults a few of the multiple G and D faults that are equivalent to the PI/PS stem faults may have to be considered for test generation (only if they are not already covered by the single G and D tests). Test generation for multiple G and D faults in our cube based algorithm is as simple as test generation for single G and D faults. Introduction of multiple G and D fault involves changing of more than one bit in the PM for the faulty FSM. This multiple fault is introduced only during justification and propagation, whereas for the activation vector any of the tests for the constituent single G or D fault will be sufficient to activate the multiple fault.

Theorem 2: *The test sequences for the single G and D faults in a single output minimized PM of an FSM will cover all single stuck faults, except the faults on the PI and PS stem lines (that have fanouts leading only to NS lines and not to any PO) in the multi-level FSM whose combinational portion is synthesized using testability preserving transformations.*

Proof: By results (i) and (ii) the single G and D faults of a single output minimized PM will cover all single and multiple faults in the equivalent two-level AND-OR combinational circuit. This is because the output ordering requirement is implicitly satisfied by a single output minimized PM, as there are no shared product terms.

Let N1 be a single output minimized PM and N2 the equivalent two-level AND-OR circuit. The multi-level circuit N3 is synthesized from N1 using only algebraic transformations. According to Theorem 1, the vector set T1 that detects all single G and D faults in N1, will also detect all multiple G and D faults in N1 and all single and multiple stuck faults in N2 and N3, as every fault in N3 has an equivalent fault in N2 [7, 11].

Now flip-flops are added to N1, N2 and N3 to form sequential circuits. Some of the PIs which are transformed to PS inputs become non-controllable and some of the POs which are transformed into NS outputs become non-observable in a single time frame. Now the single G and D fault test set for FSM N1 does not guarantee to cover all multiple G and D faults as these faults may get masked in the fanout reconvergence structure across time frames in the ILA model of the FSM.

The vector set T (derived for N1) consists of excitation vectors for all faults (single and multiple) in FSMs N2 and N3. The state transition graphs (STGs) for FSMs N1, N2 and N3 are identical for the good circuits. For every single fault *f3* in FSM N3 there is an equivalent single fault *f2* in FSM N2 [11], and an equivalent single G or D fault *f1* (or a multiple G and D fault for the PI/PS stems that have fanouts) in FSM N1. If *t1* is the excitation vector in T for *f1*, *t1* is also the excitation vector for *f2* and *f3*. The STG with *f3* for FSM N3 is exactly the same as the STG with *f2* for FSM N2, and the STG with *f1* for FSM N1. Therefore if PS of *t1* can be justified in FSM N1, it can also be justified in FSMs N2 and N3, and the jus-

Table 1: Test Generation for Combinational Part of FSMs (SUN Sparc 2)

Circuit Name	PI, PO, Prod. Terms	Personality Matrix			Multi-level Implementation				
		GDCOMB					Gentest		
		G-D Faults	No. of Vect.	CPU Sec.	Stuck Faults Total	Stuck Faults Cov %	No. of Vect.	Cov %	CPU Sec.
ex1	14, 24, 145	881	155	0.17	567	100	67	100	26.25
ex2	7, 7, 62	314	77	0.01	318	100	52	100	1.13
ex7	6, 6, 40	149	32	0.01	166	100	30	100	0.48
sse	11, 11, 62	383	103	0.06	331	100	51	100	1.25
cse	11, 11, 95	590	151	0.12	428	100	74	100	2.27
s1	13, 11, 188	1316	275	0.41	759	100	112	100	5.07
dk16	7, 8, 104	625	106	0.06	526	100	73	100	3.94
planet	13, 25, 235	1341	204	0.29	1066	100	108	100	11.81
sand	16, 14, 228	1547	346	0.53	1087	100	139	100	13.96
styr	14, 15, 228	1678	369	0.53	1127	100	149	100	16.28

tification sequence is the same for all the three FSMs, N1, N2 and N3. Similarly if the good and faulty NS of *t1* can be differentiated in FSM N1, they can be differentiated in FSMs N2 and N3, and the propagation sequence will be same in all FSMs. The presence of a justification and propagation sequence for a fault entirely depends on the STG and is independent of the structure of any particular implementation. As the PI/PS stem (that have fanouts leading only to NS lines and not to any PO) faults in FSMs N2 and N3 are equivalent to a multiple G and D fault in FSM N1, these faults are not guaranteed to be detected by the test set for single G and D faults of FSM N1. The single G and D faults that are equivalent to all other single stuck faults (except PI/PS stem faults having no branches leading to a PO) of the FSMs N2 and N3, are explicitly considered for test generation in the FSM N1. The stuck faults on the stem of a PI/PS line which have at least one fanout branch leading to a PO, will be detected by the single G or D fault on this fanout branch leading to the PO, as such faults will be detected in a single time frame. Hence if all single G and D faults are detected in the FSM N1 by the test sequence S1, the same sequence S1 will cover all single stuck faults except the PI and PS stem line (that do not have a fanout branch leading to a PO) faults and the multiple stuck faults in the equivalent two-level FSM N2 and the synthesized FSM N3. ■

III. SYNTHESIZED COMBINATIONAL CIRCUITS

We have implemented a cube based test generation and fault simulation program, GDCOMB, in C language as explained in [13].

We employed GDCOMB to derive tests from the personality matrix description of the combinational portion of 10 of the MCNC synthesis benchmark FSMs. These results are given in Table 1. GDCOMB derived vectors to cover all G and D faults. These vectors were then used to simulate all *collapsed* single stuck faults in synthesized multi-level implementations of these circuits. The coverage, as shown in Table 1, was 100% for all circuits.

Table 1 also gives the results of test generation for stuck faults in multi-level circuits by a gate level test generator, Gentest [3]. While both test generators could cover all faults, the run times of GDCOMB are significantly better. Vector sets of Gentest are, however, smaller. This is because the vector sets of GDCOMB are independent of the implementation. Such implementation-independent tests can also be derived from Gentest if vectors are generated for all single stuck faults in two-level AND-OR circuits. The fault set size and vector set size then will be comparable to those of the G-D faults and GDCOMB tests but the run time of Gentest will be even higher than that given in Table 1. The use of test vectors generated from two-level AND-OR description was suggested by Dave and Patel [5].

The multi-level combinational circuits were synthesized from the *single-output minimized* two-level description, employing *algebraic* factorization and a simple technology mapping scheme that uses only primitive gates of up to four inputs and inverters. The synthesis system MIS [2] was used in our experiments. Since only testability preserving transformations [7, 11] were employed, the 100% fault coverage was expected. The two-level and multi-level combinational circuits of the FSMs were irredundant with respect to single stuck faults.

IV. SYNTHESIZED FSMs

We use an extension of the PLA test generation method to derive tests for the G and D faults in the FSM. We sandwich the combinational test vector between the state justification and fault propagation sequences to obtain a complete test sequence for the G or D fault under consideration. The pseudocode for the

```
Testgen(M)
FSM M;
Goodstate(M) = Reset;
For every undetected fault F do {
    If Faultystate(F) ≠ Goodstate(M) then {
        Pseq = Propagate(F);
        If propagated to PO then Faultsim(Pseq);
    } else {
        Cvecs = Generate_excitation_vector(F);
        If Cvecs generated then {
            Jvecs = Justify(Cvecs);
            If justified then Pseq = Propagate(F);
            If justified and propagated to PO then
                Faultsim(Jseq+Pseq);
        }
    }
}
```

Figure 1: The main test generation algorithm

main test generation algorithm is shown in Figure 1. We use a reverse time processing for justification and a forward time processing for propagation using the ONsets of the PO and NS functions in the FSM with the fault introduced in every time frame. In an earlier paper [13] we give more details on justification, propagation and fault simulation.

We developed a C program, GDSEQ, to generate test sequences for PLA based FSMs and general sequential circuits whose combinational function can be obtained in personality matrix form. We experimented on 10 of the synthesis benchmark FSMs. The characteristics of these circuits are shown in Table 2. These circuits were available as symbolic state tables. The combinational portions of these FSMs were obtained by synthesis using MIS [2], performing only algebraic factorization, after state assignment. The last four circuits in Table 2 were state minimized before state assignment and synthesis. This was done to show that state minimization improves testability of the FSM, and is evident from the results which give 100% stuck fault coverage. The lower G and D fault coverage in Planet is due to propagation failures.

The results obtained from GDSEQ are given in Table 3. GDSEQ generated test sequences for all G and D faults in 5 of these circuits. The coverage of G and D faults in other circuits was lower due to various reasons like sequential redundancy, the value of time frame limit, and the backtrack time limit used in justification stage of the program. As stated earlier, a power-up reset was assumed only at the beginning of the test sequence.

Next, the GDSEQ vectors were used to simulate all collapsed single stuck faults in the multi-level gate implementations of FSMs. A differential fault simulator [4] was used for fault simulation. As shown in Table 3, these vectors gave 100% fault coverage for 6 of the synthesis benchmarks. The lower stuck fault coverages for the remaining circuits is due to reasons like lower G and D fault coverage by GDSEQ, sequential redundancies in the FSM, and some of the PI/PS stem faults not being covered by the functional vectors.

It should be noticed that the stuck fault coverage is always higher than that of the G and D fault coverage. The *useful vectors* given in Table 3 were obtained when the vector set was truncated after the detection of the last fault. For comparison, the coverage of a set of random vectors having the same number of vectors as the useful vectors is also given in Table 3. The random vector coverage is consistently lower.

We used the sequential test pattern generator Gentest [3] to verify the efficiency of GDSEQ. Gentest is a gate-level test generator and uses the time frame expansion method. It has a differential fault simulator [4] to remove detected faults from the fault list after a test sequence is generated for a target fault. We generated test sequences for stuck faults using Gentest in the multi-level implementations of FSMs. The vector set size, fault coverage (for detected faults), and CPU times for Gentest on SUN Sparc 2 are given in Table 3. Here also power up reset is assumed. It may be seen that the stuck fault coverage of functional vectors is always equal or higher than the stuck fault coverage obtained by Gentest vectors. Further, the CPU time for GDSEQ (including time to simulate the functional vectors) is far less than the test generation time taken by Gentest. GDSEQ turns out to be 66 to 1152 times faster on these circuits.

V. LARGE CIRCUITS

Tables 4-6 give results for some larger circuits obtained in the Berkeley Logic Interchange Format (BLIF). These are networks of interconnected functional blocks, each described as a single output sum of products. We used MIS to convert BLIF to a single PM by the functions read_blif and write_pla. The circuit parameters after synthesis using algebraic transformations are indicated as number of gates and stuck faults in Table 5. Table 4 give results of GDCOMB for the combinational parts of the FSMs. Even though the time taken *per vector* by GDCOMB is better than Gentest for s953, scf and sbc, in overall time GDCOMB performs worse as the ratio of G and D faults to stuck faults becomes large.

In Table 6, we give results of GDSEQ for these

Table 2: Characteristics of Benchmark FSMs

FSM	No. of Inputs	No. of Outputs	No. of Flip-Flops	Personality Matrix		Synthesized Circuit	
				Prod. Terms	G-D Flts.	No. of Gates	Stuck Flts.
ex1	9	19	5	145	881	198	567
ex2	2	2	5	62	314	113	318
ex7	2	2	4	40	149	58	166
sse	7	7	4	62	383	112	331
cse	7	7	4	95	590	157	428
s1	8	6	5	188	1316	279	759
dk16	2	3	5	104	625	182	526
planet	7	19	6	235	1341	360	1066
sand	11	9	5	228	1547	354	1087
styr	9	10	5	228	1678	379	1127

Table 3: GDSEQ Vs. Gentest for Synthesized FSMs (Sun Sparc 2)

FSM	Personality Matrix		Multi-level Implementation							
	GDSEQ						Random Vector Cov %	Gentest		
	No. of Vect.	G-D Flt. Cov %	TGen. CPU Sec.	FSim. CPU Sec.	Useful Vect.	Stuck Fault Cov %		No. of Vect.	Stuck Fault Cov %	CPU Sec.
ex1	690	99.54	9.54	3.28	690	99.8	86.8	251	98.4	4693.0
ex2	283	100.0	1.19	1.19	283	100.0	86.8	247	98.7	2022.0
ex7	58	61.74	0.76	0.21	58	80.7	60.8	83	80.1	1118.0
sse	439	93.47	1.49	1.37	439	99.1	67.9	233	98.8	395.0
cse	676	100.0	4.53	2.56	676	100.0	60.9	340	98.6	470.0
s1	1141	85.86	44.14	6.56	1141	98.8	97.7	304	93.5	3925.0
dk16	191	100.0	1.56	1.44	179	100.0	97.9	346	100.0	1403.0
planet	644	99.9	50.80	5.39	624	100.0	99.0	509	100.0	8446.0
sand	1536	100.0	56.66	12.52	1519	100.0	97.6	580	100.0	22425.0
styr	1237	100.0	90.66	12.52	1208	100.0	74.7	754	100.0	35868.0

Table 4: Test Generation for Combinational Part of Large FSMs (SUN Sparc 2)

Circuit Name	PI, PO, Prod. Terms	Personality Matrix			Multi-level Implementation				
		GDCOMB					Gentest		
		G-D Faults	No. of Vect.	CPU Sec.	Stuck Faults		No. of Vect.	Cov %	CPU Sec.
					Total	Cov %			
s838	66, 33, 593	2803	1181	48.0	724	100	144	100	2.0
s641	54, 42, 912	8452	2485	178.0	879	100	187	100	6.7
s713	54, 42, 912	8452	2490	175.0	879	100	187	100	6.7
s953	45, 52, 234	1972	496	6.0	1079	100	115	100	3.8
scf	34, 61, 385	2989	436	6.0	1655	100	198	100	6.0
sbc	68, 84, 555	3194	819	31.0	1726	100	286	100	15.2
dsip	452, 421, 4022	25128	9208	16393.0	6491	100	36	100	100.0
s5378	199, 213, 8195	86717	18852	49720.0	6529	100	979	100	344.1
key	486, 421, 5926	39576	10296	28184.0	8466	100	401	100	308.4
bigkey	486, 421, 5926	39576	9962	26041.0	9051	100	621	100	762.0

Table 5: Characteristics of Large FSMs

FSM	No. of Inputs	No. of Outputs	No. of Flip-Flops	Personality Matrix		Synthesized Circuit	
				Prod. Terms	G-D Flts.	No. of Gates	Stuck Flts.
s838	34	1	32	593	2803	365	724
s641	35	23	19	912	8452	301	879
s713	35	23	19	912	8452	301	879
s953	16	23	29	234	1972	400	1079
scf	27	54	7	385	2989	583	1665
sbc	40	56	28	555	3194	564	1726
dsip	228	197	224	4022	25128	2370	6491
s5378	35	49	164	8195	86717	2443	6529
key	258	193	228	5926	39576	3089	8466
bigkey	262	197	224	5926	39576	3563	9051

Table 6: Functional Test Generation for Large FSMs (Sun Sparc 2)

FSM	GDSEQ				Gentest		
	Time in sec.	No. of vecs.	GD flt. cov %	Stuck flt. cov %	Time in sec.	No. of vecs.	Stuck flt. cov %
s838	3801	592	6.4	25.4	11675	4102	41.0
s641	2278	3203	42.9	77.4	477	318	82.0
s713	2264	3183	43.2	78.8	793	345	81.1
s953	547	1558	79.9	99.1	178	334	96.8
scf	66	1228	78.7	93.5	23558	497	93.4
sbc	2382	899	54.7	74.2	976	299	76.0
dsip	24965	4572	61.4	92.7	109	641	100.0
s5378	64873	1732	30.9	42.5	2650	942	60.5
key	23788	2612	88.3	96.4	668	1401	99.2
bigkey	18059	2146	85.3	95.0	138	610	99.3

FSMs. As these circuits require large amounts of CPU time and memory we ran GDSEQ considering one out of every ten faults for test generation but all faults for fault simulation. Table 6 also shows the stuck fault coverage of functional vectors in the synthesized circuit, which again is higher than the coverage of functional faults. Here scf is the only FSM where GDSEQ performs better than Gentest. In all other cases the ratio of G and D faults to stuck faults, and the ratio of product terms to number of gates is high. Gentest performs better in such cases.

The lower G and D fault coverages in Tables 3 and 6 are due to failures in propagation, as our implementation of the propagation algorithm is not complete. The time frame limit for justification and propagation is also a limiting factor in obtaining higher fault coverages. The number of vectors generated by GDCOMB and GDSEQ can be reduced by compaction techniques [10], but it will remain higher than the number of stuck fault tests, which are implementation dependent.

VI. CONCLUSION

We have shown theoretically that the G and D fault model is capable of guaranteeing a high coverage of stuck faults in combinational and sequential circuits synthesized employing algebraic transformations. The functional fault model also allows us to generate tests that are independent of the specific logic implementation. A major advantage of this approach is that functional test generation combined with fault simulation is considerably faster than gate level algorithms that target stuck faults in a specific implementation. For the relatively few stuck faults that may not be detected by the functional test sequence, it is possible to generate additional tests using any gate level sequential circuit test generator. However, our results also show that, when the ratio of G and D faults to stuck faults is high, this method is not efficient. In such cases, partitioning into an interconnection of small personality matrices may be better, and that is the objective for our future research.

References

[1] V. D. Agrawal and S. C. Seth, *Test Generation for VLSI Chips*, IEEE Computer Society Press, Los Alamitos, CA, 1988.

[2] R. K. Brayton, R. Rudell, A. Sangiovanni-Vincentelli and A. R. Wang, "MIS : A Multiple Level Logic Optimization System", *IEEE Trans. CAD*, vol. 6, pp. 1062-1081, November 1987.

[3] W. T. Cheng and T. J. Chakraborty, "GENTEST: An Automatic Test-Generation System for Sequential Circuits", *Computer*, vol. 22, pp. 43-49, April 1989.

[4] W. T. Cheng and M. L. Yu, "Differential Fault Simulation for Sequential Circuits", *Jour. Electronic Testing: Theory and Applic.*, vol. 1, pp. 7-13, February 1990.

[5] U. Dave and J. H. Patel, "A Functional-level Test Generation Methodology using Two-level Representations", *Proc. 26th Design Automation Conf.*, pp. 722- 725, 1989.

[6] D. L. Dietmeyer, *Logic Design of Digital Systems*, Allyn and Bacon, Boston, 1971.

[7] G. Hachtel, R. Jacoby, K. Keutzer and C. Morrison, "On Properties of Algebraic Transformations and the Multifault Testability of Multilevel Logic", *Proc. Int. Conf. Computer Aided Design*, pp. 422-425, 1989.

[8] J. Jacob and V. D. Agrawal, "Functional Test Generation for Sequential Circuits", *Proc. 5th Int'l. Conf. VLSI Design*, Bangalore, India, pp 17-24, January 1992.

[9] J. Jacob and V. D. Agrawal, "Multiple Fault Detection in Two-Level Multi-output Circuits", *Jour. Electronic Testing: Theory and Applic.*, vol. 3, pp. 171-173, May 1992.

[10] J. Jacob and N. N. Biswas, "PLATES: An Efficient PLA Test Pattern Generator", *Proc. Third Int'l Workshop VLSI Design*, Bangalore, India, pp. 147-154, January 1990.

[11] J. Rajski and J. Vasudevamurthy, "Testability Preserving Transformations in Multi-level Logic Synthesis", *Proc. Int'l Test Conf.*, pp. 265-273, 1990.

[12] J. E. Smith, "Detection of Faults in Programmable Logic Arrays", *IEEE Trans. Comput.*, vol. C-28, pp. 845-853, November 1979.

[13] M. K. Srinivas, J. Jacob and V. D. Agrawal, "Finite State Machine Testing Based on Growth and Disappearance Faults", *Proc. 22nd Fault Tolerant Computing Symp.*, pp. 238-245, July 1992.