ATPG For Scan Chain Latches and Flip-Flops

Samy R. Makar* and Edward J. McCluskey

Center for Reliable Computing
Gates Hall 2A
Stanford University
Stanford, CA 94305

Abstract

A new approach for testing the bistable elements (latches and flip-flops) in scan chain circuits is presented. In this approach, we generate test patterns that apply a checking experiment to each bistable element in the circuit while checking their response. Such tests guarantee the detection of all detectable combinational defects inside the bistable elements. The algorithm is implemented by modifying an existing stuck-at combinational test pattern generator. The number of test patterns generated by the new program is comparable to the number of traditional stuck-at patterns. This shows that this approach is practical for large circuits.

1. Introduction

Scan was introduced to overcome the difficulties of sequential test generation [1], [2]. The basic idea of scan is to allow easy access to the flip-flops in the design so that test patterns can be applied directly to the inputs of the internal combinational logic, and the outputs of the internal combinational logic can be "captured" by the bistable elements and scanned to the primary output for comparison with expected values. This makes it possible to use combinational circuit test generation algorithms on sequential circuits.

A difficulty with scan based methods is that they do not address faults within the bistable elements. [3], [4] and [5] showed that tests for stuck-at faults at flip-flop inputs and outputs miss many internal faults. Simulation results presented in this paper confirm these results. The three bit binary counter shown in Fig. 1-1 was used in our simulations. We found 37 faults that were not detected by a traditional test (this test detected 100% of input and output stuck-at faults on the combinational gates as well as the flip-flops, and included a flush test of 01100 throughout the scan chain), but affect the normal operation of the counter.

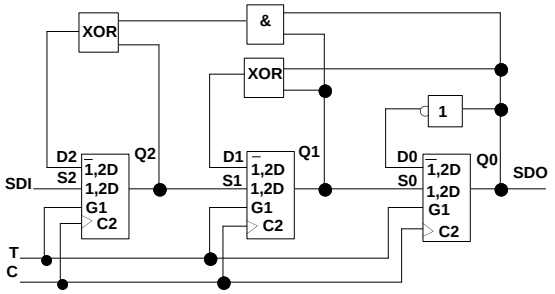

Figure 1-1 Three Bit Binary Counter.

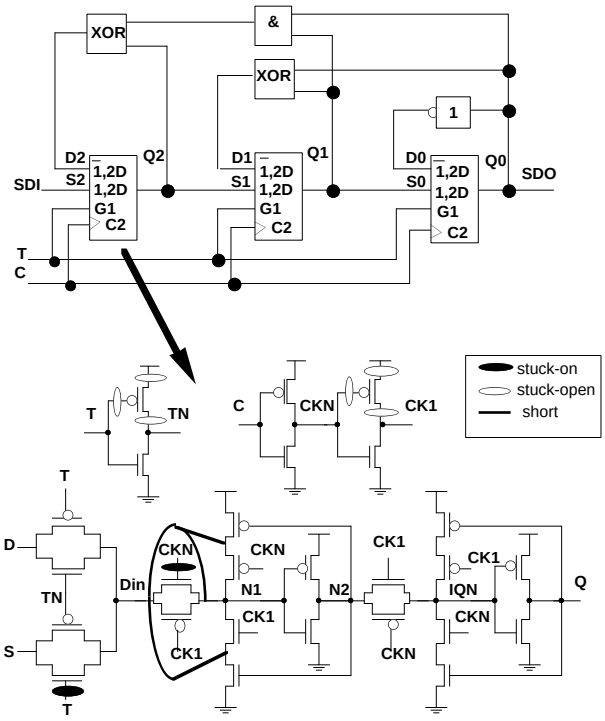

Figure 1-2 Faults in First Flip-Flop of Binary Counter Missed by 100% Stuck-At Test That Affect Normal Operation.

Fig. 1-2 shows some of these faults. All these faults are detected by our test.

Our test is based on checking experiments. A *checking experiment* is a sequence of inputs that, when applied to a circuit, gives different outputs than any

---

* Samy is currently with Cirrus Logic Inc.

other circuit with the same input, outputs, and same number of or fewer states [6]. The main advantage of a checking experiment approach over other methods (such as stuck-at ATPG) is that it is independent of the fault model, and will detect any defect that does not increase the number of states in the circuit. The main problem with checking experiments is that the number of test patterns can be very large, making it impractical for large circuits. However, we use a checking experiment only for the bistable elements. We show that such tests are comparable in size to stuck-at tests, indicating that they are practical for large circuits. We presented some early results of this work in [7]. That paper described a technique for finding tests for scan chain latches, but cannot be used for flip-flops. The technique described here applies to both flip-flops and latches. Another important difference is that here we implemented our algorithm and generated tests for all the ISCAS-89 circuits [8]. The test lengths of our tests were comparable with stuck-at test lengths.

The rest of this paper is divided as follows. In Section 2, we describe the basis of our algorithm, and in Section 3, we describe an implementation of our algorithm. In Section 4, we present the fault simulation results for a single MD flip-flop. The simulations include test patterns for stuck-at faults and checking experiment based test patterns. The results indicate that there are many faults missed by the stuck-at test. In the same section, we present test generation results for all the ISCAS-89 benchmark circuits [8]. The number of test patterns is compared with traditional stuck-at patterns. Results indicate that the number of test patterns from our program increase at the same rate as stuck-at patterns. This implies that they are practical for large circuits.

2. Checking Experiments for Bistable Elements

Flip-flops can have many flow tables, but there is only one primitive flow table (barring isomorphism) for each flip-flop type [9]. Therefore we use primitive flow tables in our analysis. In a *primitive flow table*, each row contains only one stable state. A checking experiment for a primitive flow table will detect all defects that do not increase the number of states in any column.

Given the primitive flow table of a bistable element, we can easily derive a checking experiment for the bistable element. However, if a bistable element is embedded inside a circuit, we need to find a way to get the checking experiment from primary inputs to the bistable element, and to observe the output of the bistable element under test from the primary output. This may be very difficult or impossible to do. Also, there are many possible checking experiments. Fig. 2-1 shows a small circuit with three flip-flops. We will show how to generate a checking experiment for the shaded MD flip-flop (F_3).

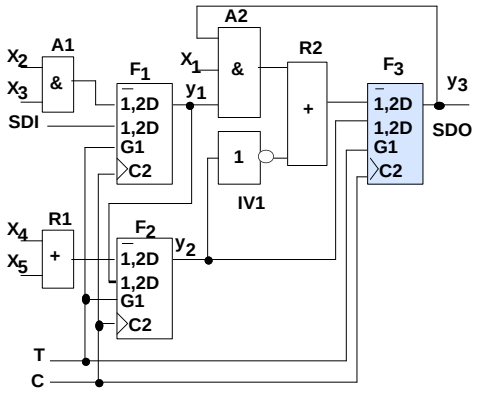

Figure 2-1 Circuit Under Test.

Instead of trying to apply a complete checking experiment directly from primary inputs, we use a divide-and-conquer approach. Every checking experiment must identify all the stable and unstable states in its primitive flow table. For each of these states a sequence of inputs must be applied to the bistable element under test, and the output of the bistable must be observed at a primary output. An example of such an input sequence for an MD flip-flop is shown in Fig. 2-2. We can use the scan chain to supply a test pattern that would set the inputs and output of the flip-flop under test to the initial values (d=0, s=1, and q=1).

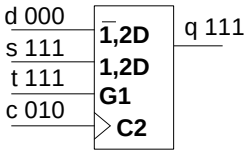

Figure 2-2 A Required Input Sequence for an MD Flip-Flop.

The method for finding a test pattern that would set d=0, s=1, and q=1 is similar to combinational ATPG. In our example, the test pattern would be $X_1 = 0$ and $y_2 = y_3 = 1$ would set d, s and q to the desired values (see Fig. 2-3). After applying C = 010, d and s will depend on the new values in F_1 and F_2. Thus, the test pattern we use must preserve s = 0 and d = 0 after the clock pulse. In our example, this can be done with the test pattern $X_1 = 0$ and $y_1 = y_2 = y_3 = 1$ (see Fig. 2-4). The difference between this and the earlier test pattern is that y_1 is set to 1, so that after the clock pulse y_2 is still 1.

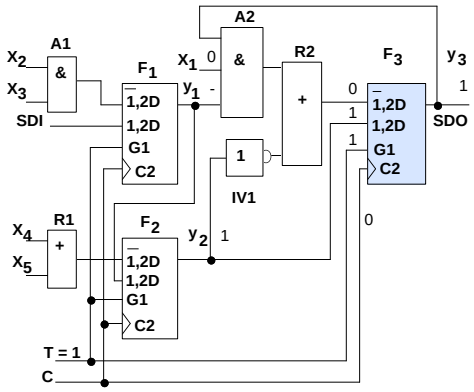

Figure 2-3 Circuit Under Test With Test Pattern.

This analysis is similar to sequential ATPG because more than one time frame is considered, i.e. we consider the values on the flip-flop before and after the clock pulse. Since we had only one pulse on C, we have to deal with only two time frames. This makes the problem much easier than sequential ATPG. In the above example, we had T = 1, and applying a pulse on C caused the scan chain to shift once. Thus the operation of generating a test pattern for such a sequence is called a shift operation. Other required sequences need other types of operations. These elementary operations are summarized in Table 2-1. The first two elementary operations are used with sequences for which there is no pulse on the clock. The third (the one we showed in the example) and fourth are used with sequences in which there is a pulse on the clock. The last elementary operation is used in conjunction with the other elementary operations when the next flip-flop in the scan chain cannot "capture" the output of the flip-flop under test.

Elementary operations can be used to generate test patterns for all the required input sequences [10]. Therefore, we can define an algorithm for generating test patterns for the bistable elements using the algorithm in

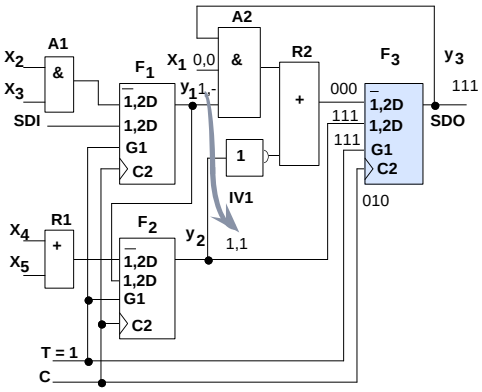

Figure 2-4 Circuit Under Test With Test Pattern.

Table 2-1 Elementary Operations

Operation	Description
Single Cycle	Determine bit values of a test pattern that would set lines in the circuit to desired values.
Single Cycle Change	Determine bit values of a test pattern that would set lines in the circuit to desired values, and by changing values only on the primary inputs would change the value of a line in the circuit.
Shift Operation	Determine bit values of a test pattern that would set lines in the circuit to desired values, and after the scan shifts by one, would again set some lines in the circuit to desired values. The values on the lines need not be the same for both cycles
Normal Operation	Determine bit values of a test pattern that would set lines in the circuit to desired values, and after a normal cycle (bistable element input selected from combinational logic), would again set some lines in the circuit to desired values. The values on the lines need not be the same for both cycles.
Combinational Logic Sensitization	Determine bit values of a test pattern that would sensitize a line in the circuit to a primary output or an input of a bistable element.

Fig. 2-5. In this algorithm, we use elementary operations to find a test pattern for each required sequence of each flip-flop in the circuit. The test patterns are placed in pattern tables that are compacted using standard test pattern compaction techniques. There are four common scan chain architectures [9]. The architectures use different bistable elements for scan cells. Different bistable element types have different required sequences, and thus even though their algorithms have the structure in Fig 2-5, each will have a different implementation.

```
for each bistable element {
  for each required sequence {
    Apply Appropriate Elementary Operation
    Add Test Pattern To Appropriate Table
  }
}
Compact Tables
Print Tables
```

Figure 2-5 Algorithm for ATPG for Bistable Elements.

3. Implementation

We implemented our algorithm by modifying an existing stuck-at ATPG program in SIS [11]. This was done by first creating elementary functions and then using the elementary functions to write procedures for the four different bistable element types used in the scan chain architectures.

As with most ATPG programs, this program reads a gate level description of the circuit. However, unlike most ATPG programs, the output is not simply a file with test patterns, but rather a set of files with test patterns. The number of test pattern files depends on the scan architecture used. Each file of patterns corresponds to a different type of sequence that has different timing on the clock and control inputs. Details can be found in [10].

4. ATPG Results

The effectiveness of a test can be measured by the number of defects it can detect. Even though the stuck-at models are often used for fault simulation, we use the more accurate (for CMOS circuits) CrossCheck fault models, [12] and [13], for our simulation. The fault models comprise shorted interconnects (STI), open interconnects (OPI), short-to-power (STP), short-to-ground (STG), transistor stuck-on (SON), and transistor stuck-open (SOP). In the simulations, faults are injected by modifying a copy of the circuit description. The faulty circuits were simulated using HSpice [14].

In CMOS, there are some faults whose presence does not change the functionality of the host circuit. Some of these cannot be detected (and thus are untestable or redundant). Others that cannot be detected by a Boolean voltage test (since the circuit functionality is correct) can, nevertheless, be discovered by a current test or a delay test [15]. The simulations reported here record whether tests caused excessive supply current (IDDQ) or incorrect outputs. The current limit for IDDQ testing is often determined experimentally, by plotting the values of many good and bad die, and selecting an appropriate threshold that would detect as many faulty circuits as possible without discarding many good ones [16] and [17]. For our simulations, the current limit is determined by plotting the maximum observed current for each fault, and selecting an appropriate threshold from the graph.

In Section 4.1, we present simulation results for an MD flip-flop, comparing traditional tests with checking experiment based tests. In Section 4.2, we present ATPG results for all the ISCAS 89 circuits. We compare the length of our tests with the length of traditional stuck-at tests. The test lengths increase at the same rate, indicating that not many more test patterns are needed for large circuits.

4.1 MD Flip-Flop Fault Simulation.

Four different tests for the MD flip-flop were simulated using HSpice. The first test, a traditional test, is based on scanning in and out the 01100 test pattern, and test patterns that would detect stuck-at 0 and stuck-at 1 faults on the D input of the flip-flop. The second test is a pin fault test set, which targets stuck-at faults on the input and output of the MD flip-flop. The other two tests are a checking experiment for the MD flip-flop and a checking experiment for the MD-latch in a scan chain [10]. The flip-flop implementation used for the simulation is shown in Fig. 4.1-1. This implementation is selected because it is a commonly used structure.

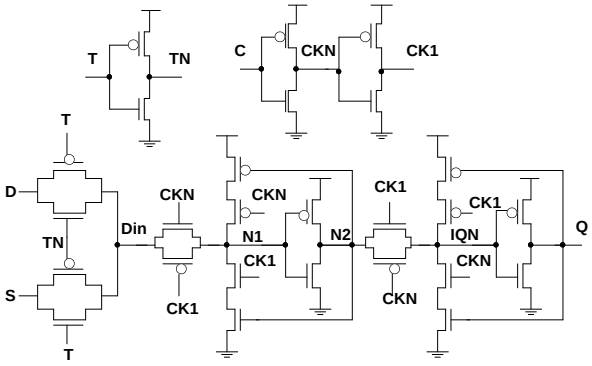

Figure 4.1-1 MD Flip-Flop Implementation Used in Simulation.

The results of the simulations are shown in Table 4.1-1. From the table, there are 19 faults that were not detected by the checking experiment. These faults are shown graphically in Fig. 4.1-2. The table also shows that the pin fault test misses ten faults that are detected by the checking experiment. These faults are shown in

Table 4.1-1 Number of Faults Detected in MD Flip-Flop (Total Faults = 256).

	Boolean and IDDQ	Boolean Alone (100 ns and 10 ms)	Boolean Alone (100 ns, 10 ms)	IDDQ Alone
a	212	167	(145,166)	155
b	227	184	(162,183)	161
c	237	207	(186,204)	182
d	237	206	(184,204)	181

a = Traditional Test, b = Pin Fault Test, c = MD Flip-Flop Checking Exp., d = Scan MD Flip-Flop Checking Exp.

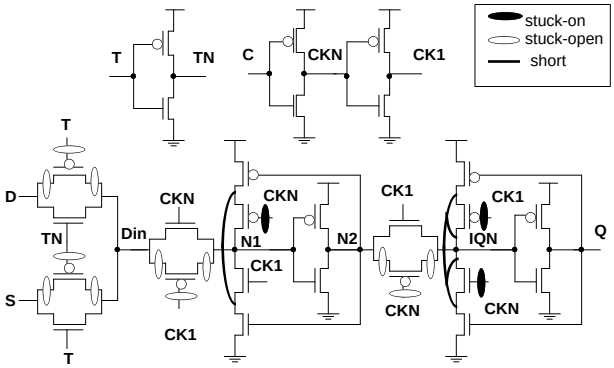

Figure 4.1-2 Faults Missed by Checking Experiment of MD Flip-Flop (19 of them).

Fig. 4.1-3. In these figures white ovals indicate stuck-open or open-interconnect faults, black ovals indicate SON faults, and thick black lines indicate shorted-interconnect faults. All short-to-power and short-to-ground faults are detected by all tests.

The faults missed by the checking experiment fall into two groups. The first group of faults missed by the checking experiment is the stuck-open faults on the transmission gates. These faults, though undetectable, could add a delay to the circuit, and will thus behave as delay faults. A test pattern that would detect a path delay fault to the input of the flip-flop may be able to detect these faults. The other group of faults missed by the checking experiment, the stuck-ons and shorted-interconnects, will turn the master or slave latch into a dynamic latch. Since a dynamic latch cannot guarantee holding its value for a very long time, then loading a value and waiting a long time may change the value in the flip-flop and the fault would be detected. Thus a very slow test (data retention test) is needed for these faults.

The traditional test and the pin fault tests miss many faults (about 5 %) detected by the checking experiment.

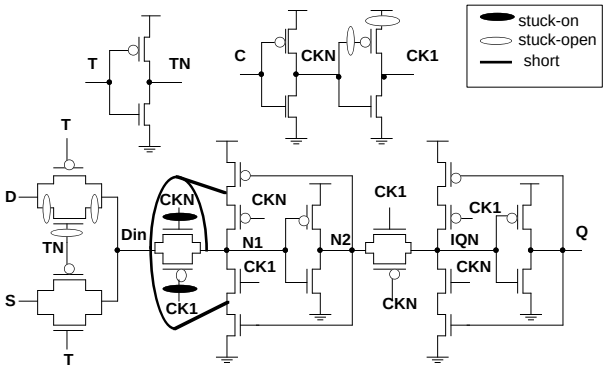

Figure 4.1-3 Faults Missed by Pin Fault Test Detected by Checking Experiment of MD Flip-Flop (10 of them).

4.2 Circuits Using MD Flip-Flop

One practical concern with testing chips is the size of the test being applied. To address this issue, we generated test patterns for the ISCAS 89 benchmark circuits for all four architectures, and compared them to the stuck-at test lengths.

Table 4.2-4 shows the number of vectors for all the ISCAS 89 circuits for each architecture, and for the stuck-at tests. The name of the ISCAS 89 circuits indicates the number of lines in the circuit. This is directly related to the size of the circuit. The number of test patterns for the LSSD architecture is always the smallest of our tests, and the number of test patterns for the MD flip-flop architecture is always the largest.

Table 4.2-4 Number of Test Patterns for Different Tests.

Circuit	MD-Latch	LSSD	MD Flip-Flop	TP Flip-Flop	Stuck-At
S27	29	19	118	52	14
S298	71	47	356	162	66
S344	97	62	395	165	65
S349	91	61	375	158	66
S382	135	85	578	244	74
S386	49	31	282	114	88
S400	133	85	576	242	71
S444	97	65	487	212	80
S510	65	43	306	149	78
S526	106	68	572	257	139
S641	187	128	578	257	124
S713	123	84	572	255	133
S820	54	37	346	144	161
S832	54	37	349	148	171
S1196	126	80	440	233	201
S1423	341	216	1640	756	218
S1488	68	45	421	179	247
S1494	68	45	422	180	243
S5378	571	336	2139	1023	700

Table 4.2-5 Number of Test Patterns Divided by Stuck-At Test Length.

Circuit	MD-Latch	LSSD	MD Flip-Flop	TP Flip-Flop	Stuck-At
S27	2.07	1.36	8.43	3.71	1.00
S298	1.08	0.71	5.39	2.45	1.00
S344	1.49	0.95	6.08	2.54	1.00
S349	1.38	0.92	5.68	2.39	1.00
S382	1.82	1.15	7.81	3.30	1.00
S386	0.56	0.35	3.20	1.30	1.00
S400	1.87	1.20	8.11	3.41	1.00
S444	1.21	0.81	6.09	2.65	1.00
S510	0.83	0.55	3.92	1.91	1.00
S526	0.76	0.49	4.12	1.85	1.00
S641	1.51	1.03	4.66	2.07	1.00
S713	0.92	0.63	4.30	1.92	1.00
S820	0.34	0.23	2.15	0.89	1.00
S832	0.32	0.22	2.04	0.87	1.00
S1196	0.63	0.40	2.19	1.16	1.00
S1423	1.56	0.99	7.52	3.47	1.00
S1488	0.28	0.18	1.70	0.72	1.00
S1494	0.28	0.19	1.74	0.74	1.00
S5378	0.82	0.48	3.06	1.46	1.00

To compare our test size with the test size of the stuck-at test, we calculate the ratio of the size of our tests to the size of the stuck-at tests. These ratios are shown in Table 4.2-5. The numbers in this table were calculated by dividing the number of test patterns for the bistable elements by the number of stuck-at test patterns. We use the number of patterns instead of the

number of cycles, because most of the cycles in a test pattern are used to shift patterns in and out of the scan chain. This implies that one of our test patterns will take about the same time on the tester as a stuck-at pattern. Since the ratios do not show an increase with circuit size, we conclude that the size of our test will not be a problem with large circuits.

5. Conclusions

We presented a new approach for testing bistable elements in digital circuits. Traditional approaches for testing bistable elements in a scan chain involve shifting in a sequence of zeroes and ones. We showed that this approach misses many faults in the circuit. These faults may affect normal circuit operation. Our new approach is based on checking experiments for the bistable elements. Checking experiments are used because they guarantee the detection of all faults that do not increase the number of states. Since a checking experiment makes no assumption about the circuit implementation, it is implementation independent. This is especially useful since designers often use different implementations of bistable elements to optimize their circuits for area and performance.

Our test was compared with the traditional test by performing fault simulation of some of the bistable elements. The results clearly indicate that there are faults that traditional tests miss that are detected by our new test. We also showed that the test size increases with circuit size by about the same rate as the test for stuck-at faults. In conclusion, tests based on checking experiments for latches and flip-flops are a thorough economic technique for testing the bistable elements of digital circuits.

Acknowledgment

The authors would like to thank Jonathan Chang for his valuable comments. This work was supported in part by the Ballistic Missile Defense Organization, Innovative Science and Technology (BMDO/IST) Directorate and administered through the Department of the Navy, Office of Naval Research under Grant No. N00014-92-J-1782, in part by the Advanced Research Projects Agency under Contract No. DABT63-94-C-0045, and in part by the National Science Foundation under Grant No. MIP-9107760. It was also funded in part by Cirrus Logic.

References

[1] Williams, M.J. and J.B. Angel, "Enhancing Testability of Large Scale Integrated Circuits via Test Points and Additional Logic," *IEEE Trans. on Computers*, C-22, No. 1, pp. 46-60, January, 1973.

[2] Eichelberger, E.B., and T.W. Williams, "A Logic Design Structure for LSI Testability," *Proc. 14th Des. Autom. Conf.*, New Orleans, LA. pp. 462-468, June 20-22, 1977.

[3] Reddy, M.K. and S.M. Reddy, "Detecting FET Stuck-Open Faults in CMOS Latches and Flip-Flops," *IEEE Design and Test of Computers*, Vol. 3, No. 5, pp. 17-26, October, 1986.

[4] Lee, K.J. and M.A. Breuer, "A Universal Test Sequence for CMOS Scan Registers," *IEEE Custom Integrated Circuits Conference*, pp. 28.5.1-28.5.4, 1990.

[5] Al-Assadi, W.K., "Faulty Behavior of Storage Elements and Its Effects on Sequential Circuits," *IEEE Transactions on VLSI*, Vol. 1, No. 4, December, 1993.

[6] Hennie, F.C., "Fault Detecting Experiments for Sequential Circuits," *Proc. of the Fifth Annual Switching Theory and Logical Design Symposium*, S-164, Princeton, New Jersey, pp. 95-110, 1964.

[7] Makar, S.R. and E.J. McCluskey, "Functional Tests for Scan Chain Latches," *Proc. ITC*, pp. 606-615, 1995.

[8] Brglez, F., D. Bryan and K. Kozminski, "Combinational Profiles of Sequential Benchmark Circuit," *IEEE International Symposium on Circuits and Systems*, pp. 1929-1934, 1989.

[9] McCluskey, E.J., *Logic Design Principles*, Prentice-Hall, New Jersey, 1986.

[10] Makar, S.R. and E.J. McCluskey, "Checking Experiments for Scan Chain Latches and Flip-Flops," CRC Technical Report 96-5, August 1996.

[11] Sentovich, E.M., K.J. Singh, L. Lavagno, C. Moon, R. Mrgai, A. Saldanha, H. Savoj, P. R. Stephan, R.K. Brayton, A.S. Vincentelli, "SIS A system for Sequential Circuit Synthesis," *Electronics Research Lab* Memorandum, No. UCB/ERL M92/41, 1992.

[12] Sucar, H., "High Performance Test Generation for Accurate Defect Models in CMOS Gate Array Technology," *ICCAD*, pp. 166-169, 1989.

[13] Chandra, S., K. Pierce, G. Srinath, H. Sucar and V. Kulkarni, "CrossCheck: An Innovative Testability Solution," *IEEE Design and Test of Computers*, Vol. 10, No. 2, pp. 56-67, June, 1993.

[14] Kielkowski, R., *Inside SPICE Overcoming the Obstacles of Circuit Simulation*, McGraw Hill, USA, 1994.

[15] Ma, S., P. Franco and E.J. McCluskey, "An Experimental Chip to Evaluated Test Techniques, Experimental Results," *Proc. ITC*, pp. 663-672, 1995.

[16] Hawkins, CF., "Quiescent Power Supply Current Measurement for CMOS IC Defect Detection," *IEEE Trans. on Industrial Electronics*, pp. 211-218, May, 1989.

[17] Perry, R., "IDDQ Testing in CMOS Digital ASICs - Putting It All Together," *Proc. ITC*, pp. 151-157, 1992.