A Novel Random Access Scan Flip-Flop Design

Anand S. Mudlapur[1], Vishwani D. Agrawal[1] and Adit D. Singh[1]

Abstract

Serial scan design causes unnecessary switching activity during testing causing enormous power dissipation. The test time increases enormously with the increase in number of flip-flops. An alternate to serial scan architecture is Random Access Scan (RAS). Here every flip-flop is uniquely addressed using an address decoder. Although it may seem to have solved most of the current problems associated with testing integrated circuits, yet one may impulsively conclude that the routing and area overhead associated with RAS is prohibitive. We present a design of the RAS flip-flop which uses a unique "toggle" mechanism, possible only in RAS. We minimize the number of gates (transistors) and eliminate the need for two globally routed (scan in and test control) signals present in earlier designs. Our design is built keeping in focus the address decoder complexity to a bare minimum. Our multistage scan-out system enables the addressed flip-flop to be observed without compromising performance due to a slow output bus. We have estimated the additional gates required to implement RAS over serial scan (SS). The design obtained equal fault coverage, 60% test vector reduction and 99% lesser power dissipation as compared to SS.

1. Introduction

Testing sequential circuits has been one of the most challenging areas in digital circuits. Automating test generation for large sequential circuits without Design for Testability (DFT) logic has met with marginal success. Additional hardware is usually added to boost the fault coverage to a desired level. Serial scan (SS) design has been one of the most successful methods in testing digital circuits. Although it enables the application of combinational test generation algorithm, alternative techniques are sought after because of some inherent drawbacks like increased test time and test power consumption. Several methods are suggested and implemented to circumvent this problem. A widely successful method is partial scan [1]. But it is a trade off between the ease of testing and the costs associated with scan design. Cross check methodology [2] provides a comprehensive solution to test sequential circuits and almost solves all the problems related to test application time. It provides massive controllability and observability.

Power consumption during testing is much higher than during normal circuit operation. It is important and vital to target low power dissipation during testing, since excessive heat can damage the circuit under test. The long scan-in/scan-out sequences trigger random circuit activity resulting in high power consumption. Test scheduling is a common approach to avoid the damage of complex devices, such as SOC [3, 4]. As a result test parallelism is reduced and testing time eventually increases. It is a well known fact that serial scan

[1] Auburn University, Dept. of ECE, 200 Broun Hall, Auburn University, AL 36849, USA; Email: {mudlaas, vagrawal, singhad}@eng.auburn.edu.

operation may create unacceptably high activity due to frequent transitions in scan chain. To prevent this, the scan clock is slowed down [5].

ATPG based methods have also been used to target the power issue [6]. However, this method often results in longer test sequences. Compaction of test vectors can reduce the length of tests, but the compacted vector set generally induces more activity resulting in higher power consumption [7]. To overcome this problem modification of test vectors for power saving have also been addressed [8]. Another method studied to reduce test power and/or test application time is modifying the order of scan cells or inserting inversion logic between scan cells after the test generation [9]. Seth et al. in [10] describe a scheme known as double-tree scan architecture to reduce test power. Although the power saving is quite significant, the test time and test data volume either remains the same or more. A modified scan-architecture to reduce test time in full-scan circuits has been addressed in [11]. They illustrate a reduction of test time by 50%; nevertheless test power still remains a matter of concern.

Testing for path delay faults in non-scan sequential circuits is complicated by the limited state transitions during normal operation. Normal-scan sequential circuits can be tested for delay faults, but the vector-pairs must be specially generated [13]. However, high fault coverage is dependent on the circuit and cannot be guaranteed due to the correlation between the two vectors. Our design can be used to test delay faults very efficiently [21].

All the problems stated above are due to the underlying architecture used, which is SS! Random Access Scan (RAS) [14, 15] is a single concurrent solution to all of them. As the name implies, each scan-cell is randomly and uniquely addressable. The architecture described in [16] targets reduction of both test application time as well as power consumption simultaneously, which are otherwise complementary objectives. A modified scheme of RAS has been described in [17], although with a different name. In this technique, the captured response of the previous pattern in the flip-flops is used as a template and modified by a circular shift for the subsequent pattern.

In this paper, we describe the design of our unique RAS-cell, aiming at minimizing the routing complexity in contrast to the architecture described in the previous work [16]. A variety of latch designs have been proposed to ease the difficulty associated with scan testing [18, 19]. The rest of the paper is organized as follows: Our new toggle RAS latch design along with an optimized routing of decoder signals is described in Section 2. An algorithm to compact the test vectors is described in Section 3. Experimental results on ISCAS ’89 and ’93 benchmark circuits are presented in Section 4.

2. RAS Flip-Flop Design

In RAS, a decoder is used to address every flip-flop. Hence at any given point of time only one flip-flop is accessed while the other flip-flops retain their state. Therefore circuit activity is localized to that part of the circuit that is glued to the FF. The architectures described in the literature [14, 15, 16, 20] mainly consists of a scan-in signal that is broadcasted to all the flip-flops, a test control signal that is also broadcasted to all the flip-flops and a unique decoder signal from the decoder to every flip-flop. The output from the flip-flop is either fed into a MISR or they are ORed to a primary output justifying the logic. Let us consider the design of RAS FF described in [16] as a reference. It can be seen from Figure 3, the number of signals associated with the flip-flops are: 1. Scan-in, which is a globally routed signal unlike serial scan, 2. Test control signal, 3. A unique decoder signal from the decoder, 4. An output signal that feeds a MISR and 5. A clock signal.

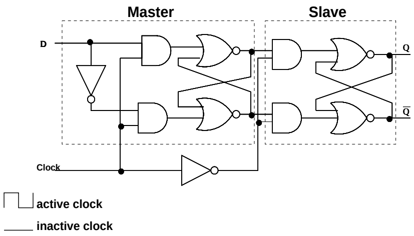


Figure 1. Master-slave flip-flop

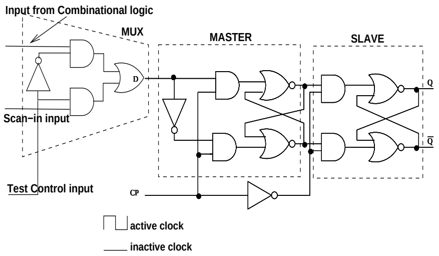


Figure 2. Serial scan flip-flop

In the design that we have developed, we use a unique toggling scheme wherein the addressed flip-flop toggles its present state in the test mode, there by reducing a separate globally routed scan-in signal. The output from the flip-flop is fed into a bus. Thus the addressed flip-flop places its value on the bus in the test mode providing the necessary observability.

The design of our RAS flip-flop can be described by three operations that are essential to satisfy the test requirements, which are, to capture the response of the circuit in the normal mode, to toggle the current state of the flip-flop being addressed and retrieve the contents simultaneously, and finally make sure that all unaddressed flip-flops hold their previous states while one flip-flop is being accessed during test mode. The operations are summarized in the first column of Table 1. We have assumed the flip-flop to be made up of a master and a slave latch as shown in Figure 1 as a reference. Every flip-flop gets two inputs, one from the row (x) and one from the column (y) decoder. The other inputs are clock and data from the combinational logic. The combinations used for the three functions are listed in Table 1.

Table 1. RAS signals.

Function	Clock	Address decoder outputs	
		Row (x)	Row (y)
Normal data	active	0	0
Toggle data	inactive	1	active clock
	inactive	active clock	1
Hold data	inactive	1	0
	inactive	0	1
	inactive	0	0

The operation of the modified scan-flip-flop can be described using Figure 4. In the normal mode of operation, the x and y lines are ‘0’s and the decoders are disabled. The output at every AND gate inside the flip-flop is ‘0’ enabling the OR gate and routing the data from the combinational logic through the mux to be captured in the flip-flops. The master is latched at the pulse of the clock and the slave is latched subsequently. In the test mode, the clock is stopped and the row and column decoders select one line each to address a flip-flop at its intersection. Hence only the flip-flop which is addressed sees a ‘1’ at both x and y lines. The mux now routes the inverted contents of the flip-flop to the master, we address this as the toggle mode. The signal on the x or y is then made ‘0’, performing the function of a clock to load the slave latch. This operation can happen at any desired frequency. Hence the addressed flip-flop toggles its state and at the same time the tri-state buffer is enabled to route the data previously stored in the flip-flop to a common bus. Meanwhile, the other flip-flops have to hold their previous states while the toggle operation is being performed on one flip-flop. Since the output from the AND gate is ‘0’, the master latch never gets activated since the clock is turned off and hence the slave latch holds its previous state. One must note that addressing a flip-flop reads the contents of the flip-flop as well as toggles its contents. Hence the contents of the flip-flop after a read operation would be opposite to the value that was read out. Care is to be taken to avoid race condition in the flip-flop.

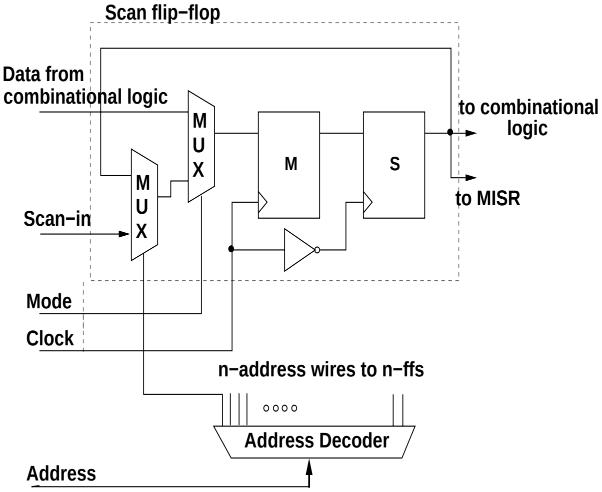
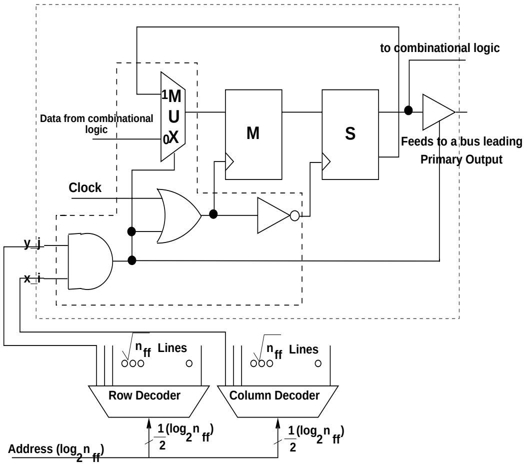


Figure 3. Design of RAS as described in [16].


to combinational logic
1M U 0X
Data from combinational logic
M
S
Feeds to a bus leading Primary Output
Clock
y_j
x_i
$\sqrt{n_{ff}}$ Lines
$\sqrt{n_{ff}}$ Lines
Row Decoder
Column Decoder
$\frac{1}{2}(\log_2 n_{ff})$
$\frac{1}{2}(\log_2 n_{ff})$
Address ($\log_2 n_{ff}$)

n_{ff} = Total number of flip−flops

Figure 4. Modified scan flip-flop to implement RAS.

All the flip-flops can be cleared initially by using a built-in circuitry, which in the clear mode would read each flip-flop and based on its current contents determine if another read operation is to be performed to clear it. For example, during the clear mode, if a flip-flop is read and is found to contain logic ‘0’ then the same flip-flop is addressed again to toggle its present state which is logic ‘1’. This calls for two clock cycles. While in the case when the first read results in logic ‘1’, the next cycle is a dummy cycle and the flip-flop is left unaddressed. Hence, the number of clock cycles to clear all flip-flops would be twice the number of flip-flops in the circuit.

The row and column decoders are built in such a way that the row and column lines intersect to address a flip-flop. This design has the least area and routing overhead compared to other decoding schemes. The total number of rows and columns depends on the number of flip-flops and the actual layout. The least number of horizontal and vertical lines would be the case when both are equal in number and numerically equal to the square root of the number of flip-flops in the circuit. Let us assume that the row decoder decodes one among the m lines and the column decoder decodes one among the n lines, where the total number of flip-flops are m x n. It is assumed that the inputs to the decoder fans-out from the primary inputs of the circuit. Therefore the number of inputs to the circuit must be greater than $\log_2 m + \log_2 n$.

While using cross check [2], an entire row needs to be addressed and a single flip-flop value cannot be set until the contents of all other flip-flops in that row are known. Hence a single change cannot be performed in cross check since the outputs are fed to a MISR and the contents of the flip-flops cannot be monitored. In our architecture we can address any flip-flop and read its value without any constraint correspondingly.

2.1 Routing

The architecture described in [16] used three separate signals to control any given flip-flop apart from the signal feeding-in from the combinational logic. This design is illustrated in Figure 3. Our design performs the equivalent function using only a decoder signal. There by eliminating two globally routed signals to the flip-flop. The output from every flip-flop is connected to a bus [21] that leads to a primary output pin. This is analogous to the “Test control” signal being routed in the SS, only that the TC signal is connected to every flip-flop from a primary input pin. The scan-in signal, which forms a chain from a primary input to a primary output through all the flip-flops in SS, is eliminated and a signal from the decoder to each flip-flop is added. The conventional decoder scheme used in [16] becomes very complex and cumbersome to implement since a single wire would have to be routed to every flip-flop. Also the decoder complexity will grow proportionally. For 65,536 (64K) flip-flops, 65,536 unique wires will have to be routed across the IC and would require 64K 16-input AND gates to decode 16 address lines. The outputs of the flip-flops are fed to a MISR, i.e., every flip-flop feeds to an MISR in the previous RAS designs.

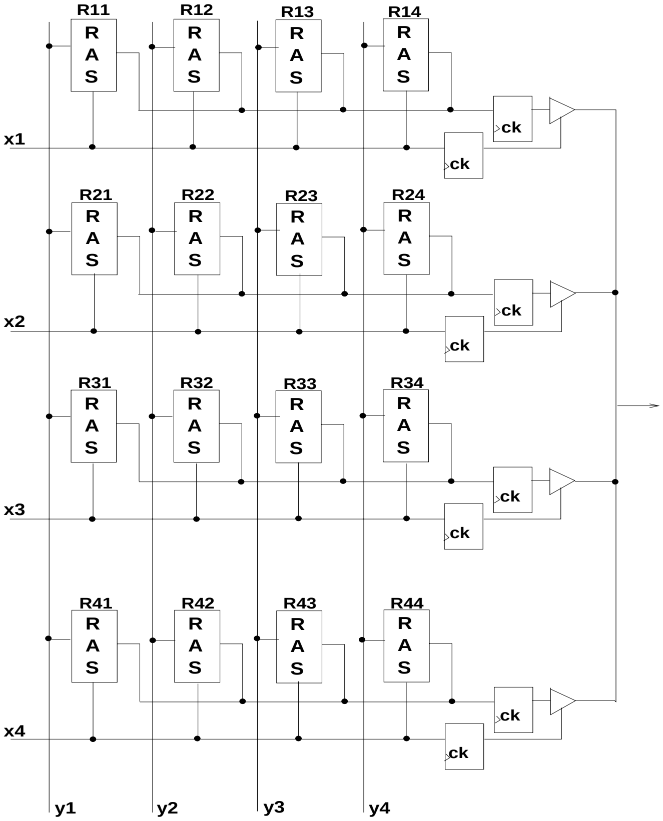


Figure 5. Routing of decoder signals in RAS.

2.2 Scan-out Design

Our scan-out design is a hierarchical structure that ensures there is no loading on the flip-flops while driving the output. The idea is illustrated in Figure 5. A cluster of flip-flops in a close proximity, feed to a common bus. The contents of the bus are stored in a normal D-flip-flop clocked by the normal clock which is suppressed in the test mode to the rest of the circuit. The row address that activates the flip-flop is also captured in another flip-flop. The contents of the flip-flop are propagated further in the next clock cycle. This scheme was developed considering the drivability of the tri-state buffers. The outputs are pipelined to minimize the delays that may have resulted without the hierarchical structure. It is intuitive from Figure 5 that the succeeding stage would have the outputs enabling the tri-state buffers, ORed and fed to a flip-flop to preserve the address. We are also evaluating a scheme with sense amplifiers and pre-charged lines to read the contents of the flip-flops.

2.2.1 Area Overhead

Assuming a circuit has n_g gates and n_{ff} flip-flops, each consisting of 10 gates and considering that the flip-flop is designed as shown in Figure 1, the gate overhead of SS [12] and RAS is given by equations (1) and (2) respectively;

$$\textit{Gate overhead of serial scan} = \frac{4 \times n_{ff}}{n_g + 10 \times n_{ff}} \times 100\% \qquad (1)$$

The RAS flip-flop has 4 gates of the mux similar to scan-flip-flop and the gates in Figure 1, the additional gates that are added are one AND-OR-INVERT (AOI) and a tri-state buffer as shown in Figure 4. i.e., the logic can be minimized by using one complex gate (AOI) and using the same inverter that is used to invert the clock in a flip-flop. The logic shown within the dotted box in Figure 4 can be further minimized. For the number of gates increased by the decoder, let us assume a decoder structure built using pass transistors. The number of transistors required to decode '$\log_2$ c' lines to 'c' lines approximately equals 2 × c. Assuming that a gate is made up of 4 transistors and the number of horizontal and vertical lines equals $2 \times \sqrt{n_{ff}}$, the gate overhead of RAS can be approximated by equation (2);

Let us consider a circuit with 5120 gates, let us also assume that there are 512 flip-flops in the circuit. The gate overhead of serial scan is 20% based on equation (1) and the gate overhead of RAS is 30.2% based on equation (2). Hence we estimate an increase of 10% in the x-dimension of the layout.

$$\textit{Gate overhead of RAS} = \frac{6 \times n_{ff} + \sqrt{n_{ff}}}{n_g + 10 \times n_{ff}} \times 100\% \qquad (2)$$

Comparing the transistor level implementations of SS and RAS from the schematic that were obtained using Design Architect tool by Mentor Graphics on Sun Ultra 5 machine, the RAS flip-flop design had an addition of 16 transistors compared to SS. Hence we can formulate the transistor overhead similar to the gate overhead calculation as follows:

$$\textit{Transistor overhead of serial scan} = \frac{10 \times n_{ff}}{n_t + 28 \times n_{ff}} \times 100\% \qquad (3)$$

Here n_t is the number of transistors in the circuit without the flip-flops and each flip-flop is made up of 28 transistors. There are 16 transistors extra in RAS compared to serial scan, hence the equation becomes,

$$\textit{Transistor overhead of RAS} = \frac{26 \times n_{ff} + \sqrt{n_{ff}}}{n_t + 28 \times n_{ff}} \times 100\% \qquad (4)$$

3. Algorithm to Compact Test Vectors

A greedy algorithm is developed to compact the test vectors. Here the vectors for the combinational circuit are obtained using an ATPG[2]. The vectors are sequenced based on the response captured by the flip-flops for an input vector along with the change in state of those flip-flops that are read where the faults have propagated during the application of the previous vector. The algorithm is as follows;

1. *Obtain the combinational vectors along with good circuit responses and store the results in a stack*

[2] Vectors were obtained from HITEC/PROOFS [22, 23] and circuit responses and outputs where faults were detected on each vector were obtained using AUSIM [24]

```
2. Find the flip-flops where faults are propagated at each
   vector
3. While number of vectors > 0
       (a) Read all the flip-flop where the faults are detected
       (b) Choose the next vector from stack that has least
           hamming distance from current flip-flop states
4. End While
```

Assuming that the power dissipation in the CUT is directly proportional to the number of transitions in the primary inputs and the transitions in the states of flip-flops, the power dissipation in RAS is reduced drastically. This is the consequence of the fact that the only activity during scan mode is the transition in state of a single flip-flop under consideration and transitions at the primary input pins that control the decoder.

Table 2. Power estimation based on transitions at the inputs for different benchmark circuits

Circuit	**No. of Transitions in SS tests**	**No. of Transitions in RAS tests**	**Test power saving (%)**
s208	1866	1209	35.21
s349	4755	1233	74.07
s386	2495	1515	39.28
s420	11587	4708	59.37
s510	3141	2382	24.16
s641	27715	7924	71.41
s838	72914	17782	75.61
s1196	57409	10601	81.53
s1269	77755	7880	89.87
s3271	1744149	45971	97.36
s3384	4299362	77665	98.19
s5378	8947677	175710	98.04
S13207	230176409	211048	99.91

4. Results

The proposed architecture was modeled and tested on ISCAS benchmark circuits. The algorithm was implemented and the fault coverage was observed to be the same as SS. A reduction in test vectors up to 60% can be observed (Table 3) in most of the circuits. Maximum reduction is achieved when the average number of faults per combinational vector is small and the number of flip-flops is proportionally higher. Since in these cases the setup time of scan flip-flops would increase compared to RAS. The reduction in test time is slightly lower than that described in [16]. This is because of the improvement that we made in the design, by minimizing the number of signals that needs to be routed to every flip-flop. The increase in number of gates over SS is between 6%-11% (Table 3).

Relative reduction of power dissipation in the circuit is calculated assuming that, the power dissipated is directly proportional to the number of transitions in the primary inputs and states of flip-flops. The results were obtained for both SS and RAS (Table 2). It can be observed that as the size of the circuits increases, reduction in power dissipation up to 99% is achieved using RAS.

Table 3. Results of vector compaction for different benchmark circuits.

Circuit	No. of flip-flops	No. of Combi. Vectors	No. of SS vectors	No. of RAS vectors	Test time red. (%)	Gate overhead SS (%)	Gate overhead RAS (%)	(%) Increase in gate area over SS
s208	8	64	584	301	48.46	18.18	28.88	10.7
s349	11	42	687	36	46.72	19.29	30.18	10.89
s386	6	138	972	450	53.70	10.96	17.56	6.6
s420	16	128	2192	1056	51.82	17.98	28.09	10.11
s510	6	110	776	344	55.67	8.86	14.19	5.33
s641	19	142	2859	1148	59.85	13.36	20.80	7.44
s838	32	240	7952	3595	54.79	18.03	27.84	9.81
s1196	18	344	6554	2447	62.66	10.16	15.83	5.67
s1269	37	118	4521	1981	56.18	15.76	24.29	8.53
s3271	116	264	31004	12540	59.55	16.98	25.87	8.89
s3384	183	260	48759	21119	56.69	20.83	31.62	10.79
s5378	179	618	111419	48677	56.31	15.67	23.80	8.13
s13207	638	1138	727820	309132	57.53	17.80	26.89	9.09

5. Conclusion

RAS has started gaining acceptance gradually. A practically implement able architecture for RAS is proposed here. An algorithm is constructed to re-order and compact the test vectors. The flexibility of the design helps to detect non-targeted faults, since any arbitrary vector can be applied and any arbitrary flip-flop can be observed. Simulation results show that power dissipation is reduced up to 99%, and up to 60% reduction in test vectors is observed compared to serial scan. Test application time as well as power consumption in a circuit are complementary objectives in SS, but are addressed concurrently in RAS, where both are reduced simultaneously. This work is based on the premise that as technology improves and test complexity increases, a marginal increase in chip area for design for testability is least prohibitive.

References

[1] Agrawal, V. D., Cheng, K.-T., Johnson, D. D. and Lin, T. (1988), *Designing Circuits with Partial Scan*, In *IEEE Design & Test of Computers*, vol. 5, Apr, pp. 8–15.

[2] Chandra, S. J., Ferry, T., Gheewala, T. and Pierce, K. (1991), *ATPG based on a Novel Grid Addressable Latch Element*, In *ACM/IEEE Design Automation Conf.*, pp. 282–286.

[3] Chou, R. M., Saluja, K. K. and Agrawal, V. D. (1994), *Power Constraint Scheduling of Tests*, In *Proc. 7th International Conference VLSI Design*, pp. 271–274.

[4] Chou, R. M., Saluja, K. K. and Agrawal, V. D. (1997), *Scheduling Tests for VLSI Systems under Power Constraints*, In *IEEE Trans. VLSI Systems*, vol. 5, June, pp. 175–185.

[5] Chandra, A. and Chakrabarty, K. (2001), *Combining Low-Power Scan Testing and Test Data Compression for System-on-a-chip*, In *Proc. Design Automation Conf.*, pp. 166–169.

[6] Wang, S. and Gupta, S. K. (1997), *ATPG for Heat Dissipation Minimization during Scan Testing*, In *Proc. Design Automation Conf.*, June, pp. 614–619.

[7] Sankaralingam, R., Oruganti, R. R. and Touba, N. A. (2000), *Static Compaction Techniques to Control Scan Vector Power Dissipation*, In *VLSI Test Symposium*, pp. 35–40.

[8] Kajihara, S., Ishida, K. and Miyase, K. (2002), *Test Vector Modification for Power Reduction During Scan Testing*, In Proc. in VLSI Test Symposium, pp. 160–165.

[9] Dabholkar, V., Chakravarty, S., Pomeranz, I. and Reddy, S. M. (1998), *Techniques for Minimizing Power Dissipation in Scan and Combinational Circuits During Test Application*," In *IEEE Tran. on Computer-Aided Design of Integrated Circuits and Systems*, pp. 1325–1333.

[10] Bhattacharya, B., Seth, S. and Zhang, S. (2003), *Double-Tree Scan: A Novel Low-Power Scan-Path Architecture*, in International Test Conference, pp. 470–479.

[11] Hamzaoglu, I. and Patel, J. (1999), *Reducing Test Application Time for Full Scan Embedded Cores*, in FTCS, pp. 260–267.

[12] Bushnell, M. L. and Agrawal, V. D. (2000), *Essentials of Electronic Testing for Digital, Memory and Mixed-Signal VLSI Circuits*. Boston: Kluwer Academic Publishers.

[13] Cheng, K.-T., Devadas, S. and Keutzer, K. (1993), *Delay Fault Test Generation and Synthesis for Testability Under a Standard Scan Design Methodology*, In *IEEE Trans. on Computer-Aided Design*, vol. 12, Aug, pp. 1217–1231.

[14] Ando, H. (1980), *Testing VLSI with Random Access Scan*, In *Proc. of the COMPCON*, Feb, pp. 50–52.

[15] Wagner, K. D. (1983), *Design for Testability in the AMDAHL 580,* In *Proc. of the COMPCON,* pp. 384–388.

[16] Baik, D. H., Saluja, K. K. and Kajihara, S., (2004), *Random Access Scan: A Solution to Test Power, Test Data Volume and Test Time*, In *Proc. 17th International Conf. VLSI Design*, Jan, pp. 883–888.

[17] Arslan, B. and Orailoglu, A. (2004), *Test Cost Reduction through a Reconfigurable Scan Architecture*, In *International Test Conference*, Oct, pp. 945–952.

[18] Saluja, K. (1982), *An Enhancement of LSSD to Reduce Test Pattern Generation Effort and Increase Fault Coverage*, In *Proc. Design Automation Conf.*, pp. 489–494.

[19] Weste, N. and Eshraghian, K. (1992), *Principles of CMOS VLSI Design,* Reading, MA: Addison-Wesley, 2nd Ed.

[20] Plíva, Z., Novák, O. and d'Aguerre, P. B. (2003), *Hardware Overhead of Boundary Scan and RAS Design Methodologies*. Accessed 15 April 2005. Available http://www.fm.vslib.cz/ kes/pub/ecms03.pdf.

[21] Mudlapur, A. S., Agrawal, V. D. and Singh, A. D. (2005), *A Random Access Scan Architecture to Reduce Hardware Overhead*, In *International Test Conference*.

[22] Niermann, T. M. and Patel, J. H. (1991), *HITEC: A Test Generation Package for Sequential Circuits*, In *Proc. of the European Design Automation Conference*, pp. 214–218.

[23] Niermann, T. M., Cheng, W.-T. and Patel, J. H. (1992), *PROOFS: A Fast, Memory-Efficient Sequential Circuit Fault Simulator*, In *IEEE Trans. Computer Aided Design of Integrated Circuits and Systems*, vol. 11, Feb, pp. 198–207.

[24] Stroud, C. E. (2004), *AUSIM: Auburn University SIMulator - Version L2.*2., Dept. of Electrical & Computer Engineering, Auburn University, Auburn, AL.