Test Time Reduction to Test for Path-Delay Faults using Enhanced Random-Access Scan

Kim T. Le
School of Info. Sc. and Engg.
University of Canberra
Australia
kim.le@Canberra.edu.au

Dong H. Baik*
Elect. and Comp. Engg.
University of Wisconsin-Madison
USA
dong.h.baik@intel.com

Kewal K. Saluja
Elect. and Comp. Engg.
University of Wisconsin-Madison
USA
saluja@ece.wisc.edu

Abstract

Studies of Random-Access Scan (RAS) architecture have largely limited their scope to reduce test application time, test volume and test power to detect conventional stuck-at faults. In this paper we propose an enhanced RAS latch design for two pattern tests. The proposed latch is a minor modification of the RAS latch and is well suited for delay-fault tests. In contrast, the traditional serial scan latch needs a major enhancement. As a result the RAS may offer a hardware advantage while the test time is nearly halved over the serial scan design. We demonstrate the test time advantage in this paper for various test sets for benchmark circuits and we argue that the advantage is even larger when test sets are generated for RAS architecture in mind, as well as by the exploitation of unspecified bits in test vectors.

1. Introduction

Serial scan is the most commonly used design for testability techniques at present. In the serial-scan architecture, the storage elements (flip-flops) of a circuit-under-test (CUT) form one or more shift registers so that these elements are fully accessible for read and write operations. The data shifting in serial scan circuits introduces some problems: long test application time, large test data volume, and high test power consumption. Random Access Scan, (RAS) first proposed by Ando [1], was found to be somewhat impractical on the grounds of area overhead but recently a number of researchers have shown that RAS offers a practical solution to the test time, test volume and test power [2,3,8,13,14] to test for the conventional stuck-at faults. Baik and Saluja [2] proposed a RAS based test architecture, called Progressive RAS, that appears to be highly area efficient and the area overhead for this architecture is shown to be comparable to that of serial scan.

The reduction in feature size coupled with the increase in operating speed introduces new failure models, including the presence of path-delay faults (PDF) [3] due to manufacturing defects [6,7] or crosstalk [11]. Considerable literature exists that deals with enhanced serial-scan architecture, with hold latches, for application of two pattern tests [3, 4]. However, for RAS only Mudlapur et al. [8] proposed a delay test strategy, but their tests have limited capability in the sense that the two vectors that are used to test a PDF must differ in only one bit position. In this work, we study the capability of the RAS architecture to test PDFs using two pattern tests that can differ in arbitrary number of bit positions. We compare our method with that of an enhanced serial scan approach.

The paper is organized as follows. First, Section 2 presents requirements for PDF tests. Design architectures and test procedures for serial-scan and RAS circuits are presented in Sections 3 and 4 respectively. Both these sections also develop the necessary mathematics that is used to compute the length of the tests in the respective methods. Section 5 introduces a method to reduce test times for RAS circuits, and experimental results are reported in Section 6. The paper ends with a brief conclusion in Section 7.

2. Requirements for PDF Tests

Digital circuits are usually synchronized by clock signals. If for some reasons the delays (transistor or interconnect) change, such as due to fabrication defects or cross-talk effects [7, 11], an incorrect datum may be loaded into the flip-flop or observed at the primary output. Such a failure is called a *path-delay fault (PDF)*. We will limit our attention only to incorrect contents of flip-flops because primary outputs can be observed directly. Delay faults due to cross-talk effects have special properties and will not be discussed further in this paper.

A test for a PDF is composed of a vector pair *(I, J)* that should satisfy some logic requirements [3]. The vectors *I* and *J* are called the initializing and transition vectors respectively. To perform a PDF test, first, the combinational circuit is initialized with vector *I* to generate

* Dong Baik is currently working at Intel Corporation. Contribution to this work had been done before he joining Intel Corporation.

20th International Conference on VLSI Design (VLSID'07)
0-7695-2762-0/07 $20.00 © 2007 IEEE

an output vector *P*. The transition vector *J* is then applied to induce logic transition(s) on the path(s) to be tested. Finally, the result output *Q* from the combinational circuit is latched into flip-flops. The time interval between the application of *J* and the capture of *Q* must be kept exactly the same as in a normal operation, the expected delay of the circuit.

Three different test strategies are used to test for PDFs, namely broad-side, independent, or linked [9,10,12]. For a broad-side test set, each vector is the initializing vector and its result *P* from the combinational circuit plays the role of the transiting vector. For independent tests, every pair of vectors is a test and typically pairs are non-overlapping. This is the conventional PDF testing in scan environment. For a linked test set, every two consecutive vectors form a test, and they must be applied in the listed order.

In the next two sections, we will briefly review both serial and random-access scan architectures for PDF tests. Methods to reduce test time for broad-side testing are similar to those discussed in [2, 8, 12] and their gains have already been established. Therefore, in this paper we limit our investigation only to the applications of independent and linked test sets.

3. PDF Tests for Serial Scan Circuits

To enable PDF tests, a hold latch is inserted, between the traditional scan flip-flop and the combinational logic circuit, to make a scan-hold flip-flop [3]. However, in this case the inserted latch will be in critical path. An alternative solution is to use Stable Shift Register Latch (SSRL) shown in Figure 1 below, in which the hold latch L2 is connected to the master L1 in parallel with the slave L3. Note that this design is similar to the design proposed in [4] for fault-tolerance applications. The SSRL can be operated in either the shift or the normal mode. In the shift mode, both signals Φ_1 and Φ_2 are kept low, while L1 and L3 form one stage of a shift register.

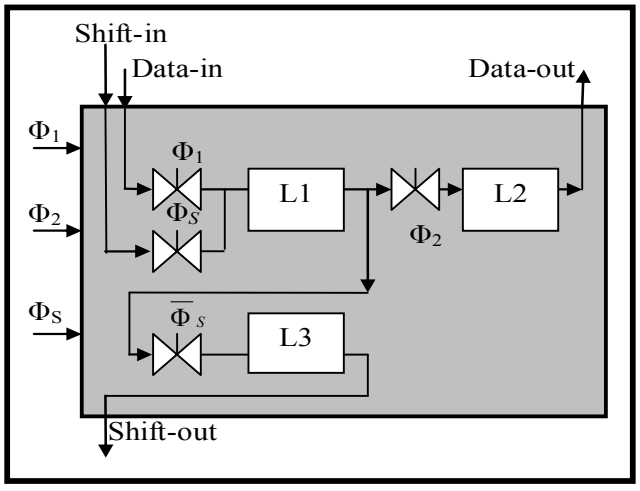

Figure 1: Stable Shift Register Latches for serial-scan

In this design the shift operation is controlled by the clocking scheme (Φ_S, $\overline{\Phi}_S$) and $\overline{\Phi}_S$ signal may be generated within an SSRL. In the normal mode, L1 and L2 are combined to form a master-slave flip-flop controlled by two non-overlapping clock signals Φ_1 and Φ_2, and the signal Φ_S is kept low.

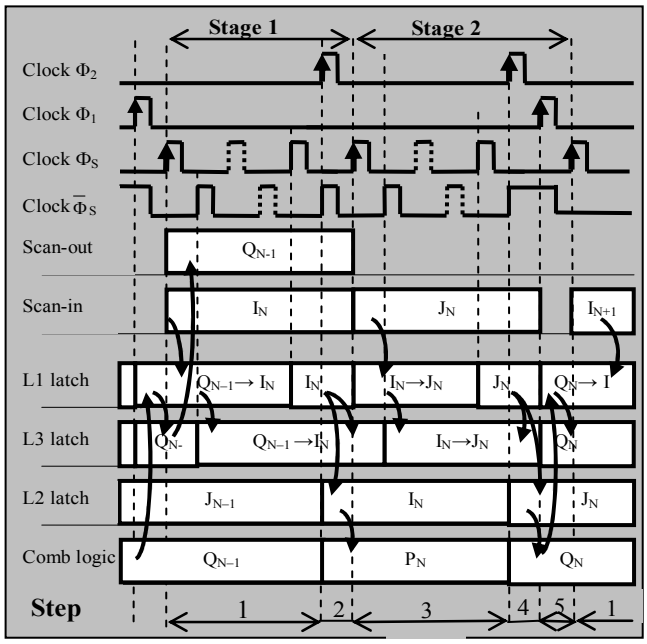

Figure 2: Two stages of a PDF test for a serial-scan circuit

An independent PDF test is performed in enhanced serial-scan environment using Algorithm 1 given below.

Algorithm 1: The test is performed in two stages as follows (refer to Figure 2):

Stage 1: Circuit initialization with vector *I* in two steps

1) I-Shift – IS: The CUT is in the shift mode and the initializing vector *I* is shifted-in using (Φ_S, $\overline{\Phi}_S$) as follows. When Φ_S is asserted a bit from the scan-in line is shifted into Latch L1 of the first flip-flop (FF), and all other L1 latches receive data from the respective L3 latches. Next, when $\overline{\Phi}_S$ is asserted the content of L1 is transferred to L3. The time required for this step is $T_{IS} = n_{FF} T_S$, where, n_{FF} is the number of flip-flops and T_S is the shift clock period.
2) I-Apply – IA: This step deals with the last bit of the vector *I* in Step 1, and the CUT is placed in the normal mode. In this step, both Φ_2 and $\overline{\Phi}_S$ are asserted, and in every FF, the content of L1 are transferred to the corresponding L3 latch as well as to L2 latch and through to the combinational circuit.

Stage 2: Circuit testing with vector *J* in three steps

3) J-Shift – JS: This step is similar to Step 1 of stage 1 but to shift the transiting vector *J*. The time required for this step is $T_{JS} = n_{FF} T_S$.

4) J-Apply – JA: This step is similar to Step 2 and deals with the last bit of vector *J*. The time T_{JA} for this step is the same as that for normal operation to satisfy the requirement in Section 2.

5) Q-Latch – QL: In this step, while CUT is still in the normal mode, all L1 latches are loaded with the output of the combinational circuit. Note that the result Q of the current test also flows through to L3 because $\overline{\Phi}_S$ is high. The time required for this step is a Φ_1 pulse, but we compute it to be one full period T_N of the normal operation, thus $T_{QL} = T_N$.

The test time to apply one test for both stages is:

$$T_{SIT} = T_{IS} + T_{JS} + T_{QL}$$

$$T_{SIT} = 2n_{FF} T_S + T_N$$

For test length comparison, we will assume in the rest of this paper that the shift clock and the normal system clocks have the same period, i.e. $T_N = T_S = 1$, therefore for each independent test (a pair of vectors) $T_{SIT} = 2n_{FF} + 1$. Thus, for a test set with S tests, the total test time is $\Sigma T_{SIT} = (2n_{FF} + 1)S + n_{FF}$, this includes the time to shift out the last test result.

For application of linked tests with serial-scan circuits, the Algorithm 1 is modified as follows. For the first linked test, both Stage 1 and Stage 2 of the Algorithm 1 are applied. For the succeeding tests, only the last three steps of Stage 2 are applied, and the test time for each of these tests is $T_{SLT} = n_{FF} + 1$. The total test time for a linked test set with S tests is $\Sigma T_{SLT} = (n_{FF} + 1)S + 2n_{FF}$.

3. PDF Tests for RAS Circuits

For RAS application we assume the underlying Progressive Random Access Scan (PRAS) [1] architecture. The architecture and the basic cell are illustrated in Figure 3. In PRAS architecture all scanned flip-flops are arranged in an array of $n_{ROW} \times n_{COL}$. A PRAS flip-flop has two latches L1, and L2. The circuit can be operated in either the normal or the PRAS mode. In the normal mode, the latches L1 and L2 are controlled by the clocking scheme (Φ_1, Φ_2). In the PRAS mode, both signals Φ_1 and Φ_2 are kept low, and each L1 can be accessed individually for read/write operations by enabling its row and column select lines in a similar way as in accessing a Random Access Memory (RAM) cell.

There is a read /write unit for each column. All flip-flops of a row are read at the same time. To start a read, a row select line is activated so that all L1 latches of the row can send their data on their column sense lines. The outputs from Read/write units might be compacted before being scanned out. For a write operation, while a row select line is high, a column select line is activated so that its Read/write circuit can drive the sense lines, and as a consequence, the datum on the scan-in line will be written into the flip-flop. The Read/write signal is used for read (low) and write (high) operations and the durations of a row read and a flip-flop write are T_{READ} and T_{WRITE} respectively.

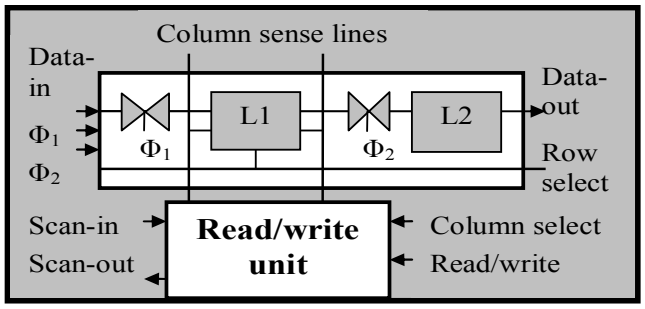

Figure 3: The PRAS flip-flops using a RAM based cell

Algorithm 2 describes the method to apply a PDF test using PRAS architecture. It is a modification of the Algorithm 1 with appropriate consideration to the properties of the PRAS architecture in read and write operations.

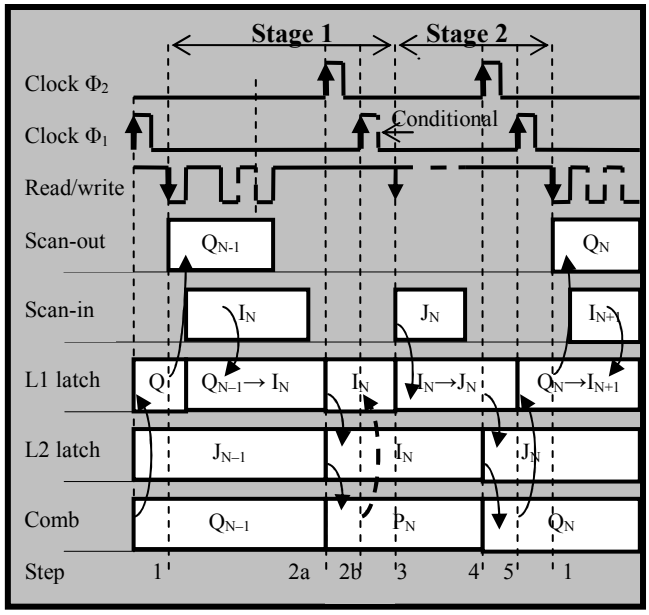

Figure 4: Two stages of a PDF test in PRAS architecture

Algorithm 2: An independent PDF test in the PRAS architecture is performed in two stages (see Figure 4 for timing details) as follows:

Stage 1: Circuit initialization with vector *I*

- Step 1 (R/W): The CUT is set to the PRAS mode. Read and write operations are performed progressively on all

rows of the flip-flop array. For each row, first, the Read/Write signal is set low and a row select line is raised high for a row read. Next, the Read/Write signal is asserted high to perform flip-flop writes as follows. If the content of an L1 latch of a flip-flop (which is its value as the result of the previous test) of the selected row is different from that of the corresponding bit of the current test vector *I*, then the corresponding column select line is activated (0 or 1) to write the test bit into the L1 latch. The time required for this step is:

$$T_{R/W} = n_{ROW}\, T_{READ} + n_{WI}\, T_{WRITE}$$

where, n_{WI} is the number of bits of I_N that need to be written. Note that a write is not required for a test bit with a value X (don't care).

- Step 2a (I Apply - IA): On completion of Write operation on the last row, the circuit is switched to the normal mode, and at $\Phi_2\uparrow$ Step 2a starts with the application of I_N on the combinational circuit as the data is transferred from L1 to L2 and the output of the combinational circuit is *P*. The time for this step is a Φ_2 pulse and we assume it to be one clock period, hence $T_{IA} = T_N$
- Step 2b (P Load – PL): This is an optimizing step as follows. Let d_{IJ} and d_{PJ} be the numbers of bits *J* and *P* differ from the corresponding bits of *I*, respectively. When the generation of the output *P* is completed, Φ_1 is asserted if $d_{PJ} < d_{IJ}$ to load *P* into L1. The time for this step is already counted in T_{IA}. Note that the loading of *P* is similar to that in a broad-side test, but in this case *P* is used as a template to write *J*, whereas in a broad-side test, *P* is used as a transition vector.

Stage 2: Circuit testing with vector *J*

- Step 3 (J Write - JW): This step is similar to Step 1 but only writes are performed. For each row, first the row select line is activated, then if a bit of *J* is different from the current content of the corresponding L1 latch, which can be a bit of *I* (or *P*, if Step 2b is applied), then the new value is written into the L1 latch. If a row does not require any write, that row is skipped, resulting in saving of a clock cycle. The test time for this step is $T_{JW} = n_{WJ} \times T_{WRITE}$, where n_{WJ} is the number of flip-flops needed to be updated to load the vector *J*.
- Step 4 (J Apply - JA): This step is the same as Step 2a but with vector *J*. We have $T_{JA} = T_N$.
- Step 5 (Q Latch - QL): This step is to capture the result output vector Q as in Step 5 of Algorithm 1. The time for this step is already included in T_{JA}.

The total test time for each independent test is:

$$T_{PIT} = T_{R/W} + T_{IA} + T_{JW} + T_{JA}$$

$$T_{PIT} = (n_{WI} + n_{WJ})T_{WRITE} + n_{ROW}\, T_{READ} + 2\, T_N$$

The first part $(n_{WI} + n_{WJ})T_{WRITE}$ of the test time for the current test pair (I_N, J_N) is depended on the values of the two vectors as well as on the previous test pair (I_{N-1}, J_{N-1}). This time is upper bounded by $n_{WI} = n_{WJ} = n_{FF}$.

Again, assuming that $T_{WRITE} = T_{READ} = T_N = 1$, and $n_{ROW} = \sqrt{n_{FF}}$; we have $T_{PIT} = (n_{WI} + n_{WJ}) + \sqrt{n_{FF}} + 2$.

For a test set with *S* tests the total test time is $\Sigma T_{PIT} = \sum (n_{WI} + n_{WJ}) + \left(\sqrt{n_{FF}} + 2\right)S + \sqrt{n_{FF}}$, including the time to read $\sqrt{n_{FF}}$ rows at the end of the last test. The ratio of the test time with PRAS and that with serial scan is $\Sigma T_{PIT}^{REL} = \frac{\Sigma T_{PIT}}{\Sigma T_{SIT}}$ and for large values of n_{FF} and *S* this ratio converges to $r_{WIT} = \frac{1}{2\,S\,n_{FF}} \sum_{Path} (n_{WI} + n_{WJ})$. However, there is an optimal order of tests that minimizes the test application time ΣT_{PIT} for PRAS. A method to find such an order will be discussed in the next section.

For the linked test sets with PRAS architecture, the Algorithm 2 is modified as follows. For the first test, both Stage 1 and Stage 2 of Algorithm 2 are used. For the other tests, only the last three steps 3) JW, 4) JA and 5) QL of Stage 2 are applied. The test time for each of these tests is $T_{PLT} = n_{WJ} + 1$. The total test time for a test set with *S* linked tests is $\Sigma T_{PLT} = \sum (n_{WJ}) + S + n_{FF} + 2\sqrt{n_{FF}} + 1$. Note that in this case the order of tests is already determined. The ratio the test time with respect to serial scan is $\Sigma T_{PLT}^{REL} = \frac{\Sigma T_{PLT}}{\Sigma T_{SLT}}$ and as before this ration converges to $r_{WLT} = \frac{1}{S\,n_{FF}} \sum (n_{WJ})$ for large values of n_{FF} and *S*.

4. Optimization of PRAS Test Time

As mentioned in Section 3, the test time for RAS architecture is much depended on the average rates of writes r_{WLT} or r_{WIT}. We now present a technique that can be used to reduce these rates for PDF tests. The technique is a modification of the method presented in [2].

Let HD be the Hamming distance between two vectors. Consider two consecutive tests (I_{N-1}, J_{N-1}) and (I_N, J_N) and their corresponding test outputs (Q_{N-1}, Q_N). The number of writes to flip-flops for test *N* is $n_{(N-1)\to N} = n_{WI} + n_{WJ}$ where $n_{WI} = HD(Q_{N-1}, I_N)$ and $n_{WJ} = HD(I_N, J_N)$. Suppose that these test pairs can be applied in any order without affecting the fault coverage of the PDF test set. As a result, the test ordering problem to reduce n_{WI} and

n_{WJ} can be formulated as a traveling-salesman problem to obtain an optimal order for the tests. For example, consider the Circuit s27, which has 4 primary inputs, 4 primary outputs, and 3 flip-flops. A test set for this circuit has 18 linked tests. For demonstration, let us consider the first six vectors shown in Table 1. Only the flip-flop inputs and outputs related to the six vectors are shown in the table.

Table 1: Partial test set for s27

Vector		Output	Vector		Output
1	010	010	2	011	011
3	000	100	4	110	001
5	010	010	6	110	001

First we consider the six vectors as three independent test pairs. For the serial-scan architecture, the total test time is $\Sigma T_{SIT} = (2n_{FF}+1)S + n_{FF} = 24$. For a PRAS design, we calculate the number of writes from one test to the next. This is shown in Table 2 and the numbers of writes are indicated as "cost" function and X indicates the start of the test. In the table, infinite (∞) cost indicates the invalid sequence, e.g. we do not allow an *I* vector to be equal to *J*. Notice that for test 2 (composed of vectors 3&4), $d_{PJ} = 1$, and $d_{IJ} = 2$, therefore, $n_{WJ} = 1$ (Step 2b in Algorithm 2). An optimal sequence obtained using a traveling-salesman solver [4] is X→1→3→2. For this sequence $\sum (n_{WI} + n_{WJ}) = 8$, $\Sigma T_{PIT} = 18$, $\Sigma T_{PLT}^{REL} = 75\%$ and $r_{WIT} = 44\%$. Note that if the sequence X→1→2→3 is applied, $\sum (n_{WI} + n_{WJ}) = 10$, providing a small reduction in the test the length.

Table 2: cost for the PDF tests

Test	1	2	3	X
1	∞	3	2	∞
2	3	∞	3	∞
3	3	2	∞	∞
X	4	4	4	∞
X→1→2→3 = 10				
X→1→3→2 = 8				

If these test are assumed to be five linked tests, we will have $\Sigma T_{SLT} = 26$, $r_{WLT} = 67\%$, and $\Sigma T_{PLT} = 21$ or $\Sigma T_{PLT}^{REL} = 81\%$.

5. Comparison and Experimental Results

As shown in Figures 1 & 3, both enhanced serial scan and PRAS flip-flops have 3 control lines and 4 data lines. A detailed comparison of hardware overhead of PRAS and traditional serial-scan circuit is reported in [2], in which the circuits are designed for stuck-at-fault tests. To enable PDF tests, the serial scan circuit is further modified to have SSRLs shown in Figure 1, by adding a transmission gate and a latch to the traditional-scan flip-flop. While, the modification required in the flip-flop of the PRAS architecture of [2] is that the signals Φ_1 and Φ_2 must be able to be separately controllable. However this requirement is also needed for the SSRL flip-flop. We can therefore conclude that the *hardware overhead for PDF testing in RAS environment is likely to be less than the hardware overhead for enhanced serial scan*.

To compare the test times. We assume that each CUT has a single scan chain. The test times for circuits with multiple scan chains will reduce inversely proportional to the number N of scan chains. This is also true if scanned FFs are arranged in N arrays for the PRAS architecture. We apply the algorithms for PDF testing using the 1-detection transition test sets of [9] for linked and independent tests. The experimental results for the linked tests are given in Table 3. The test times for PRAS architecture ΣT_{PLT}^{REL} are about 55% of the test times ΣT_{SLT} of SSRL designs. The average rate of writes, r_{WLT}, v/s the number of flip-flops for each circuit is plotted in Figure 5. It is evident that test time in nearly halved in all cases.

Table 3: Test times with linked test sets

Circuit	n_{FF}	S	Σn_W	ΣT_{SLT}	ΣT_{PLT}^{REL}	r_{WLT}
s298	14	151	1049	2293	53	49
s1196	18	749	8795	14267	67	65
s1423	74	316	12096	23848	52	51
s344	15	105	824	1710	55	52
s382	21	150	1617	3342	53	51
s400	21	139	1484	3100	53	50
s510	6	354	911	2490	51	42
s526	21	305	3245	6752	53	50
s641	19	188	1590	3798	47	44
s820	5	633	1385	3808	53	43
s953	29	558	7432	16798	47	45
s5378	179	1267	113637	228418	50	50
s9234	228	1856	226303	425480	53	53

For independent tests, we consider two consecutive vectors to form a test pair, and ignore the last vector if the test set has an odd number of vectors. The experimental results for

the independent test sets are tabulated in Table 4. We can see the efficiency of the optimization by examining the average rates of writes r_{WIT}, which are between 0.19 and 0.42. Once again the total test time is nearly 50% of that of enhanced serial scan.

Table 4: Test time with independent test sets

Circuit	n_{FF}	S	Σn_W	ΣT_{SIT}	ΣT_{PIT}^{REL}	r_{WIT}
s298	14	76	657	2218	46	30
s1196	18	375	5027	13893	52	37
s1423	74	158	9670	23616	47	41
s344	15	53	584	1658	51	36
s382	21	75	1214	3246	51	38
s400	21	70	1124	3031	51	38
s510	6	177	458	2307	50	21
s526	21	153	2231	6600	47	34
s641	19	94	1071	3685	44	29
s820	5	317	608	3492	53	19
s953	29	279	5777	16490	46	35
s5378	179	634	94857	227785	45	41
s9234	228	928	180949	424324	46	42

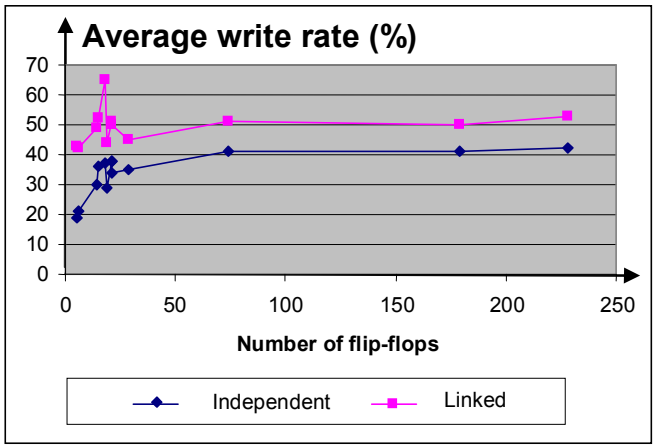

Figure 5: Average write rates for benchmark circuits.

6. Conclusion

In this paper, we examined the test time reduction problem for path delay faults requiring two pattern tests. We provideded the necessary modification require for serial scan and for RAS architectures. We developed algorithms for applying linked and independent PDF tests. Experimental results show that test times for RAS circuits are nearly half of that for enhanced serial-scan, whereas the hardware overhead is comparable. However, we believe that a PDF test generator that takes into account the independent accessibility of the scan cells in RAS architecture may offer even larger test time reduction.

Acknowledgements

The authors would like to thank I. Pomeranz and S.M. Reddy for providing the test sets used in this paper.

References

[1] H. Ando, "Testing VLSI with Random Access Scan", in Proc. of the COMPCON, pp. 50-52, Feb. 1980.

[2] D. Baik and K. Saluja, "Progressive Random Access Scan: A simultaneous solution to test power, test data volume and test time", International Test Conference, pp. 359-368, 2005.

[3] M. L. Bushnell and V. D. Agrawal, "Essentials of Electronic Testing for Digital Memory and Mixed-Signal VLSI Circuits", Kluwer Academic Publishers, 2000.

[4] S. DasGupta et al., "An enhancement to LSSD and some applications of LSSD in reliability, availability, and serviceability", Proceedings of FTCS-11, pp. 32-34, 1981.

[5] K. Helsgaun, LKH User Guide, from http://homer.ruc.dk/~keld/public_html/research/LKH

[6] W. Maly, "SIA roadmap and Design & Test", http://www.ece.cmu.edu/~maly/maly/SIA97.pdf

[7] W. Moore, G. Gronthoud, K. Baker, and M. Lousberg, "Delay-fault testing and defects in deep sub-micron ICs-does critical resistance really mean anything ?", International Test Conference, pp. 95-104, 2000.

[8] A. Mudlapur, V. Agrawal and A. Singh, "A Random Access Scan Architecture to Reduce Hardware Overhead", International Test Conference, pp. 350-358, 2005.

[9] I. Pomeranz and S.M. Reddy, "On n-Detection Test Sets and Variable n-DetectionTest Sets for Transition Faults", IEEE Trans. on CAD, vol. 19, no. 3, pp. 372-383, March 2000.

[10] J. Savir and S. Patil, "On Broad-Side Delay Test", Proc. VTS, pp. 284-290, 1994.

[11] H. Takahashi, K. J. Keller, K. T. Lee, K. K. Saluja and Y. Takamatsu, "A Method for Reducing the Target Fault List of Crosstalk Faults in Synchronous Sequential Circuits", IEEE Trans. on CAD, vol. 24, no. 2, February, pp. 252-263, 2005.

[12] W. Xiaoqing, Y. Yamashita, S. Morishima, S. Kajihara, Laung-Terng Wang, K. K. Saluja, and K. Kinoshita, "Low-Capture-Power Test Generation for Scan-Based At-Speed Test", International Test Conference, pp. 1019-1028, 2005.

[13] D. Baik and K. Saluja, "Test Cost Reduction Using Partition Grid Randon Access Scan", International Conference on VLSI Design, India, Jan. 2006.

[14] D. Baik, "A Cost Effective Solution to Multiple Test Problems", Ph.D. Thesis, University of Wisconsin, December 2005.