A Random Access Scan Architecture to Reduce Hardware Overhead

Anand S. Mudlapur, Vishwani D. Agrawal, and Adit D. Singh
Auburn University
Dept. of Electrical and Computer Engineering,
Auburn, AL 36849, USA
anand, agrawavd, adsingh@auburn.edu

Abstract

In this paper we propose an architecture for random access scan (RAS) that minimizes the signals to the RAS Flip-Flops (FF), and provide an estimate of the area overhead. Two global signals, scan-in and mode control, have been eliminated compared to previous RAS designs presented in the literature. For 'n' flip-flops, instead of routing 'n' address wires, one to each FF, we use √n wires in an 'x-y' matrix layout. A unique toggle mechanism is introduced in the RAS FF that eliminates the scan-in signal wire and reduces the vector set up time to 60% compared to traditional serial scan (SS). Serial scan induces unnecessary circuit activity during scan that causes the circuit under test (CUT) to dissipate a significant amount of power. Our design reduces this power dissipation by 99%. The problem of delay testing is highly constrained in SS and the scan-cell is often modified to assist delay testing. Any single input change delay test can be directly applied in our design. Hence all testable paths in the circuit can be effectively tested without constraints. We also propose a multistage scan-out system to observe the addressed FF avoiding a slow output bus.

1. Introduction

Testing large sequential circuits has been one of the most challenging areas in digital circuit manufacturing. Automating test generation for large sequential circuits without Design for Testability (DFT) logic has met with marginal success. Additional hardware is usually added to accomplish acceptable fault coverage in these circuits. Serial scan (SS) design has been one of the most successful methods in testing digital designs. Although it enables the application of combinational test generation algorithm, alternative methods are sought after because of some inherent drawbacks with this approach like increased test time and test power consumption. Several methods are suggested and implemented to circumvent these problems. Partial scan [1] provides a trade off between ease of testing and the costs associated with scan design. However the problem of efficiently selecting the scan registers is still open to research. At the other extreme Cross-Check methodology [2] provides a comprehensive solution to test sequential circuits and almost solves all the problems related to test application time by providing massive controllability and observability. However, the associated overheads are generally prohibitive.

Power consumption during testing is much higher than during normal circuit operation. It is often important 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 test time increases. It is a well known fact that serial scan operation may create unacceptably high activity due to frequent transitions in scan chain. To evade this problem the scan clock is slowed down [5]. Again the test application time increases, which is undesirable.

ATPG based methods have also been used to target the power issues [6]. However, this approach 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 conquer this problem modification of test vectors for power saving has also been addressed [8]. Another method researched 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 method 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 increase. A modified scan-architecture to reduce test time in full-scan circuits has been addressed [11]. The authors illustrate a reduction of test time by 50%; nevertheless power dissipation 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. An accepted method for

0-7803-9039-3/$20.00 © 2005 IEEE

overcoming this difficulty is to use a scan chain consisting of enhanced scan FFs which makes the application of arbitrary vector pairs possible. However this technique requires a hold-latch connected to each FF in addition to a "HOLD" signal that must be routed to every hold latch. This increases the area overhead and also adds some delay in the scan path [12]. Normal-scan sequential circuits can be tested for delay faults, but the vector-pairs must be specially generated [13]. Here, the first vector V1 is scanned in (usually with a slow scan clock) and is then replaced in the scan register by either (a) applying V2 which is obtained by a one-bit shift to the scan register also known as *scan-shift delay test* [14, 15], or (b) propagating V1 through the combinational logic in the normal mode, where the state portion of V2 must be justified by V1, known as *Functional broad-side delay test* [16]. However, high fault coverage is dependent on the circuit and cannot be guaranteed due to the correlation between the two vectors.

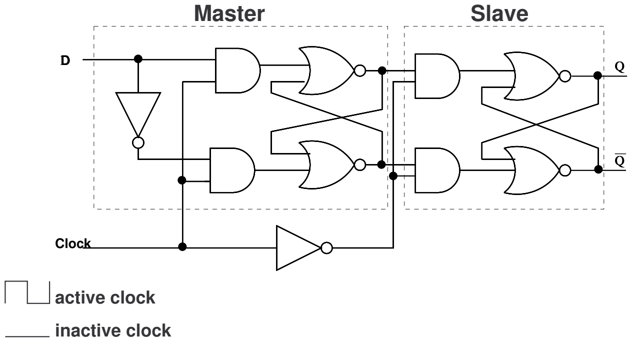


Figure 1 **Master-Slave Flip-Flop**

All the problems stated above are due to the underlying architecture used, which is traditional serial scan. Random access scan (RAS) [17, 18] offers a single solution to most of these problems. As the name implies, each scan-cell is randomly and uniquely addressable. A recent RAS architecture [19] 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 [20], although with a different name. Here, the captured responses of the previous patterns in the FFs are used as a template and modified by a circular shift for the subsequent pattern.

The concept of Random Access Scan (RAS) where every Flip-Flop is addressed uniquely has been subject to criticism from the outset because it is generally believed that the cost associated with routing is overwhelming. This argument has shelved the idea for 25 years now. In this paper, we describe an architecture for the RAS-cell, aiming at minimizing the routing complexity as compared to the architecture described in the previous work [19]. This work is based on the premise that as technology advances (more gates per die), a minimal increase in design for testability (DFT) at the cost of reduced computational resources and increased flexibility in testing will be least prohibitive.

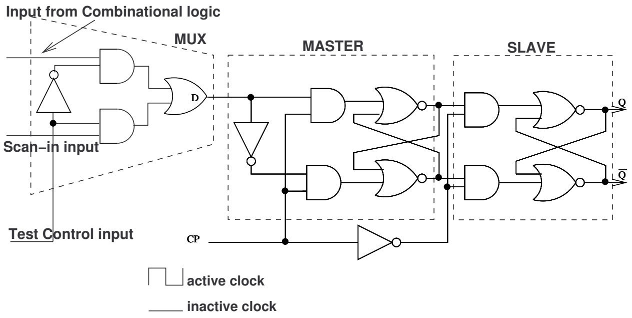


Figure 2 **Serial Scan Flip-Flop**

This paper is organized as follows. The architecture along with an optimized routing of decoder signals and the method of testing of the test-circuitry is described in Section 2. An algorithm to compact the test vectors is described in Section 3. In Section 4, delay testing using RAS is investigated. Experimental results on ISCAS Benchmark circuits are presented in Section 5. An algorithm to generate test vectors especially for RAS technique is described in Section 6.

2. New RAS Architecture

In SS, FFs form a seamless chain from the scan-in pin to the scan-out out pin in the test mode, forming a shift register structure. During normal mode of operation the input to the FFs is from the combinational logic. During scan-in/scan-out, every FF is subject to change in state. This leads to continuous activity in the FFs as well as the combinational circuits dissipating a lot of power, which is very undesirable. In RAS, a decoder is used to address every FF. Hence at any given point of time only one FF is accessed while the other FFs retain their state. This way no activity takes place in the circuit during the scan mode or the test mode. The architectures described in the literature [17, 18, 19, 21] mainly consists of a scan-in signal that is broadcasted to all the FFs, a test control signal that is also broadcasted to all the FFs, and a unique decoder signal from the decoder to every FF. The output from the FF is either fed into a MISR or the outputs are ORed to a primary output justifying the logic.

The design could become unwieldy if a unique decoder signal is routed to every FF and the scan-in signal is broadcasted. In the design that we have developed [22], we use a unique toggling scheme wherein the addressed FF toggles its present state in the test mode, there by reducing a separate globally routed scan-in signal. The outputs from the FFs are fed into a bus. Thus the addressed FF places its value on the bus in the test mode providing the necessary observability.

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

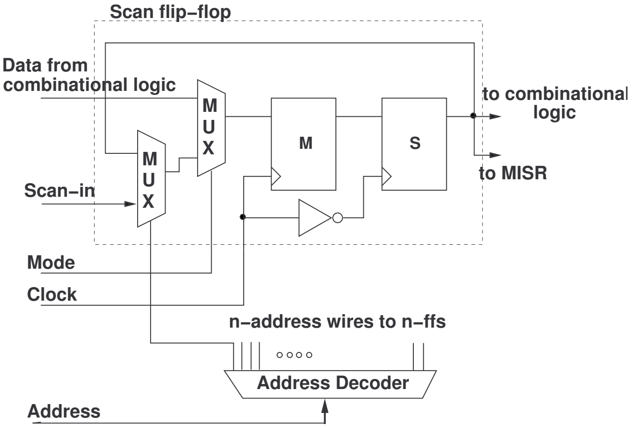
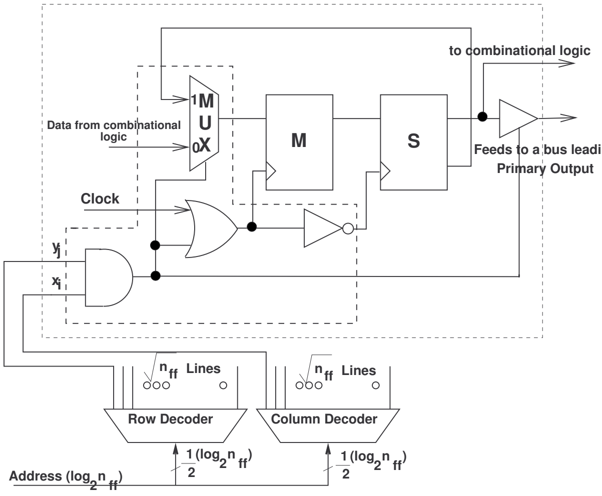


Figure 3 Design of RAS as described in [18]

The design of our RAS FF 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 FF being addressed and retrieve the contents simultaneously, and finally make sure that all unaddressed FFs hold their previous states while one FF is being accessed during test mode. The operations are summarized in the first column of Table 1. We have assumed that the FF is made up of a master and a slave latch similar to the one illustrated in Figure 1.

Every FF gets two inputs, one from the row (x) and one from the column (y) decoder. The other inputs are clock and data input from the combinational logic. The combinations used for the three defined functions are listed in Table 1. The operation of the modified scan-FF 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 FF is '0' enabling the OR gate and routing the data from the combinational logic through the multiplexer to be captured in the FFs. 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 FF at its intersection. Hence only one FF which is addressed sees a '1' at both x and y lines. The multiplexer now routes the inverted contents of the FF to the master, we call this 'toggle' mode. The signal on the x or y is then switched to '0', performing the function of a clock to load the slave latch. This operation can happen at any desired frequency. Thus the addressed FF toggles its state and at the same time the tri-state buffer is enabled to route the data previously stored in the FF to a common bus. Meanwhile, the other FFs would have to hold their previous states while the toggle operation is being performed on the addressed FF. Since the output from the AND gate is '0' at all the unaddressed FFs, 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 FF reads the contents of the FF as well as toggles its contents. Therefore the contents of the FF after a read operation would be opposite to the value that was read out. Care must be taken to avoid a race condition in the FF. This can be achieved by inserting appropriate delays.



n_{ff} = Total number of flip−flops

Figure 4 Modified Scan Flip-Flop to implement RAS

All the FFs can be cleared initially by building in reset or clear circuitry. In the clear mode each FF would be read 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 FF is read and is found to contain logic '0' state then the same FF is addressed again to toggle its present state which is logic '1'. This calls for two clock cycles. In the case when the first read is logic

‘1’, the next cycle is a dummy cycle and the FF is left unaddressed. Hence, the number of clock cycles to clear all FFs would be twice the number of FFs in the circuit.

The row and column decoders are built in such a way that the row and column lines intersect to address a FF. This design has the least area and routing overhead compared to other decoding schemes we evaluated. The total number of rows and columns depends on the number of FFs and the actual layout. The least number of horizontal and vertical lines would occur in the case when both are equal in number and numerically equal to the square root of the number of FFs 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 FFs is $m \times n$. 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 $log_2m + log_2n$.

In the Cross-Check methodology [2] an entire row needs to be addressed and a single FF can be set only if the contents of all other FFs in that row are known. And this scheme would not work if a MISR is used to capture the outputs. In our architecture we can address any FF without any constraint and read the value for it correspondingly since we do not use a MISR. This enables us to randomly set and observe any FF.

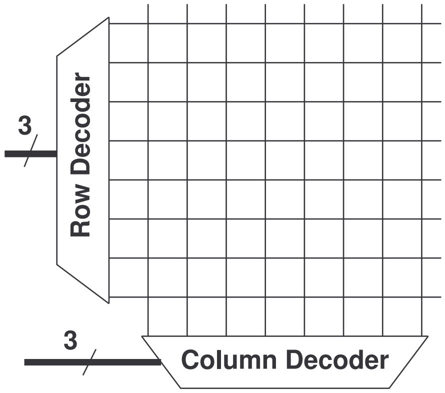


Figure 5 **Decoder design**

2.1 Routing

The architecture described in [19] used three separate signals to control any given FF 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 FF. The output from every FF is connected to a bus that leads to a primary output pin. This is analogous to the “Test-control (TC)” signal being routed in the SS, only that the TC signal is connected to every FF from a primary input pin. The scan-in signal, which forms a chain from a primary input to a primary output through all the FFs in SS, is eliminated and a signal from the decoder to each FF is added. The conventional decoder scheme used in [19] becomes very complex to implement since a single wire would have to be routed to every FF. Also the decoder complexity will grow proportionally. For 65,536 (64K) FFs, 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. A MISR is used to collect the responses, i.e., every FF feeds to a MISR in the previous RAS designs.

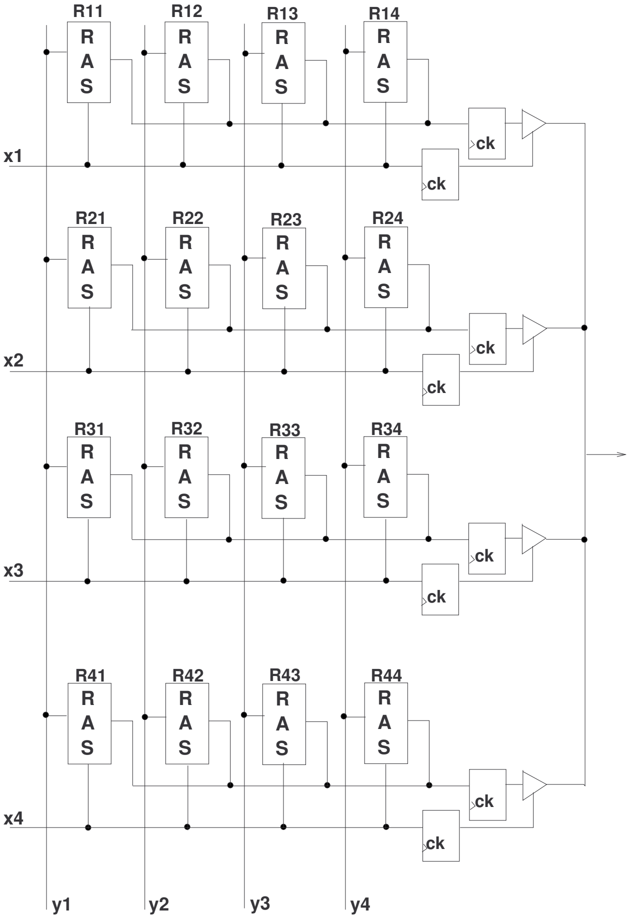


Figure 6 **Routing of decoder signals in RAS**

The grid architecture shown in Figure 5 was found to be the most efficient way to layout the decoders. The total number of extra routes added is $m + n$. With a minimum of two layers of metal routing, the row wires can be accommodated within the channel in-between the cell rows and the column wires can be routed over the cell in the next metal layer. Hence there will be an increase of one track per channel (assuming ‘*m*’ channels) and ‘*n*’ tracks that are routed on the next metal layer. Let us assume a circuit with 65,536 (64K) FFs like before. Let us also assume a square layout that has 256 routing channels. Thus every row will contain 256 FFs. i.e. $m = 256$ and $n = 256$. The total number of additional tracks will be 256 + 256 = 512, 256 tracks in the 1st layer of metal and 256 tracks in the 2nd layer of metal. Let the length of every channel be 'l' µm and assuming the vertical dimension to be a linear multiple of the channel length, i.e., $(q \times l)$ µm, the increase in length of routes is $(q + 1) \times l$ µm. Hence 65,536 wires have been reduced to 512 wires.

2.2 Scan-out Design

We have designed a novel mechanism for the scan-out of the FFs. This is a hierarchical structure that ensures there is no loading on the FFs while driving the output bus. The idea is illustrated in Figure 6. A cluster of FFs in close proximity feed to a common bus. And the contents of the bus are captured by a normal D-FF clocked by the normal clock which is suppressed in the test mode to the rest of the circuit. The row address that activates the FF is also captured in another D-FF. The contents of the FF 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. The signals feeding the tri-state buffers are ORed and captured in a FF to preserve the address for the next stage. We are also evaluating a scheme with sense amplifiers and pre-charged lines like in conventional memory to read the contents of the FFs.

2.3 Gate Area Overhead of RAS

Assume a circuit with 'n_g' gates and 'n_{ff}' FFs each consisting of 10 gates. Assume the scan FF is designed as shown in Figure 2, the gate overhead of SS [12] and RAS is given by equations (1) and (2), respectively;

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

The RAS FF has 4 gates of the multiplexer similar to scan-FF, 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 inside the FF. 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_2c' lines to 'c' lines approximately equals $2 \times c$. Let us assume that a simple gate made up of 4 transistors on an average and $n_{ff} = n$ (horizontal lines) × n (vertical lines); the gate overhead of RAS can be approximated by the equation shown below:

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

As an example let us consider a circuit with 5,120 gates and assume that there are 512 FFs in the circuit. The gate overhead of serial scan is 20% from Equation 1 and the gate overhead of RAS is 30.2% from Equation 2. Hence there is an increase of 10% in the x dimension of the layout.

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:

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

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,

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

2.4 Testing

The tests target all the stuck-faults in the CUT. Consistently dominant faults are modeled on the tri-state buffers in the circuit [23, 24]. The decoder is first tested using the MATS++ [25] test. The FFs are cleared initially. The clear operation is described above. The FFs are initialized to all zero state and then the tests are performed.

$$\{ \ \Updownarrow (w0); \ \Uparrow (r0, w1); \ \Downarrow (r1, w0, r0) \ \}$$

This test adequately tests for address decoder faults (AF) unlinked with transition faults (TF) and all AFs linked with TFs. All the stuck-at faults (SAF) are detected since a ‘0’ and a ‘1’ is read uniquely from each cell.

⇕ - Addressing order can be either increasing or decreasing

⇑ - Increasing memory addressing order

⇓ - Decreasing memory addressing order

After the test-circuitry is tested for fault free operation, the FFs are setup to perform the routine tests. The initial states are loaded into the FFs and the combinational inputs are applied at the primary input. The vector sequences required to test the decoder and FFs are linearly proportional to the number of FFs in the circuit.

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[1]. The vectors are sequenced based on the response captured by the FFs for an input vector along with the change in state of those FFs 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. Find the FFs where faults are propagated at each vector
3. While number of vectors > 0
    a) Read all the FF where the faults are detected
    b) Choose the next vector from stack that has least hamming distance from current FF states
4. End While
```

During scan-in, the CUT is subject to unnecessary activity and all the FFs are subject to change state. Various methods are presented in the literature to mask the FF transitions during test mode [28, 29]. Let us assume 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 FFs; The power dissipation in RAS is reduced drastically, since, the only activity during scan mode is the transition in state of a single FF under consideration and transitions at the primary input pins that control the decoder.

4. Delay Testing Using RAS

Delay testing in SS circuits is very constrained. In proposals for enhanced scan to support delay tests, the scan-FFs are modified and HOLD latches [30] are typically inserted between the FFs and the combinational logic. The latches insert excess delays in the path and increase area overhead due to routing of an additional control (HOLD) signal. A one bit change can be obtained very easily using RAS, which is very vital in the case of delay testing. A vector V1 is set up and vector V2 with a one bit change is applied. It is known that any testable path can be tested by a single input change vector pair [31]. These tests are easy to apply in RAS but cannot be guaranteed in SS. A change in state of a FF, only needs one clock and the circuit response is captured in the next clock cycle, thereby testing a desired path for delay. Hence delay testing, can be performed using RAS with no additional hardware, and any combinational delay test vector pair generated will work for sequential circuits using RAS.

It is also expected that fault diagnosis, which is a lengthy process for SS, can be made very efficient with RAS.

5. Results

The proposed architecture was modeled and tested on ISCAS benchmark circuits. The algorithm described above was implemented and the fault coverage seen with the transformed vectors were observed to be the same as SS. A reduction in test vectors up to 60% was observed (Table 3) for most of the circuits. Maximum reduction is achieved when the average number of faults per combinational vector is small and the number of FFs is proportionally higher. Since in these cases the setup time of scan FFs would increase compared to RAS. The reduction in test time is slightly lower than that described in [19]. This is because of the improvement that we made in the hardware efficiency of the design, by minimizing the number of signals that needs to be routed to every FF.

Relative reduction of power dissipation in the circuit was calculated assuming that the power dissipated is directly proportional to the number of transitions at the primary inputs and states of FFs. The results were obtained for both SS and RAS (Table 2). It can be observed that, as the

---

[1] Vectors were obtained from HITEC/PROOFS [26] and circuit responses and outputs where faults were detected on each vector were obtained using AUSIM [27].

size of the circuits increases, reduction in power dissipation up to 99% was achieved using RAS.

6. Potential for Customizing ATPG to Further Decrease Test Time

The results presented in this paper are based on the vectors obtained using existing ATPG algorithms. A slight modification in the form of an added constraint in the ATPG algorithms can further decrease the number of test vectors needed using RAS. The following algorithm can be employed to obtain this further compaction of test vectors;

```
1. Set the cost function of setting a FF to be the highest
2. Generate a vector to target a fault
3. Perform Fault simulation
4. While number of faults > 0
    a) Read all the FF where the faults are detected
    b) Target a fault and Generate the next vector with minimum changes to be made in the FFs from the current states considering the change of state due to a read operation.
    c) Perform Fault simulation
5. End While
```

Consider modifying PODEM's [32] back-trace algorithm, such that the pseudo-controllability[2] of the FF (pseudo primary inputs) is set very high. Thus during back-trace, a minimal set of FFs are assigned for each targeted fault. This will require the least number of flip-flops to be set at test. Furthermore, the test for the next fault is generated with minimum changes to the test response captured in the scan chain from the current test, again to minimize test application time. Early experimentation on the smaller benchmark circuits indicates that such a strategy can show a 30-40% improvement in test time. Its worthwhile noting that, better vector compaction can be achieved for larger circuits using this algorithm.

7. Conclusion

A more practically implementable architecture for RAS is proposed here. An algorithm is developed 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 FF 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 the 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, an increase in chip area for design for testability will become more acceptable.

Table 2 Power estimation based on number of transitions at the inputs of various 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

8. References

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

[2] S. J. Chandra, T. Ferry, T. Gheewala, and K. Pierce, “ATPG Based on a Novel Grid Addressable Latch Element,” in Proc. *ACM/IEEE European Design Automation Conf.*, pp. 282 – 286, June 1991.

[3] R. M. Chou, K. K. Saluja, and V. D. Agrawal, “Power Constraint Scheduling of Tests,” in *Proc. 7th International Conf. VLSI Design*, pp. 271 – 274, Jan. 1994.

---

[2] FFs are actually controllable here, but a minimum set of FF changes is targeted.

Table 3 Results of vector compaction for various benchmark circuits

Circuit	No. of flip-flops	No. of Comb. 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

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

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

[6] S. Wang and S. K. Gupta, “ATPG For Heat Dissipation Minimization During Scan Testing,” in *Proc. European Design Automation Conf.*, pp. 614 – 619, June 1997.

[7] R. Sankaralingam, R. R. Oruganti, and N. A. Touba, “Static Compaction Techniques to Control Scan Vector Power Dissipation,” in *Proc. VLSI Test Symposium*, pp. 35 – 40, May 2000.

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

[9] V. Dabholkar, S. Chakravarty, I. Pomeranz, and S. M. Reddy, “Techniques for Minimizing Power Dissipation in Scan and Combinational Circuits during Test Application,” in *IEEE Trans. On Computer Aided Design of Integrated Circuits and Systems*, vol. 17, pp. 1325 – 1333, Dec. 1998.

[10] B. Bhattacharya, S. Seth, and S. Zhang, “DoubleTree scan: a novel low-power scan-path architecture,” in *Proc. International Test Conf.*, pp. 470 – 479, Oct. 2003.

[11] I. Hamzaoglu and J. Patel, “Reducing Test Application Time for Full Scan Embedded Cores,” in *Proc. Fault-Tolerant Computing Symposium*, pp. 260 – 267, June 1999.

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

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

[14] J. Savir, “Skewed-Load Transition Test: Part, Calculus,” in *Proc. International Test Conf.*, pp. 705 – 713, Sept. 1992.

[15] J. Savir, “Skewed-Load Transition Test: Part II, Coverage,” in *Proc. International Test Conf.*, pp. 714 – 722, Sept. 1992.

[16] J. Savir and S. Patil, “On Broad-Side Delay Testing,” in *IEEE Trans. on VLSI*, vol. 2, pp. 368 – 372, Sept. 1994.

[17] H. Ando, “Testing VLSI with Random Access Scan,” in *Proc. of the COMPCON*, pp. 50 – 52, Feb. 1980.

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

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

[20] B. Arslan and A. Orailoglu, “Test Cost Reduction through a Reconfigurable Scan Architecture,” in *Proc. International Test Conf.*, pp. 945 – 952, Oct. 2004.

[21] Z. Plíva, O. Novák, and P. B. d’Aguerre, “Hardware Overhead of Boundary Scan and RAS Design

Methodologies", Accessed 15 Apr. 2005. Available at http://www.fm.vslib.cz/ kes/pub/ecms03.pdf.

[22] A. S. Mudlapur, V. D. Agrawal, and A. D. Singh, "A Novel Random Access Scan Flip-Flop Design," in *Proc. VLSI Design & Test Symposium*, Aug. 2005.

[23] T. J. Powell, "Consistently Dominant Fault Model for Tristate Buffer Nets," in *Proc. VLSI Test Symposium*, pp. 400 – 404, Apr. 1996.

[24] S. T. Chakradhar, S. G. Rothweiler, and V. D. Agrawal, "Redundancy Removal and Test Generation for Circuits with Non-Boolean Primitives," in *IEEE Trans. CAD*, vol. 16, pp. 1370 – 1377, Nov. 1997.

[25] A. J. van de Goor, *Testing Semiconductor Memories: Theory and Practice*. Chichester, UK: John Wiley & Sons, Inc., 1991.

[26] T. M. Niermann and J. H. Patel, "HITEC: A Test Generation Package for Sequential Circuits," in *Proc. European Design Automation Conf.*, pp. 214 – 218, Feb. 1991.

[27] C. E. Stroud, "AUSIM: Auburn University SIMulator – Version L2.2." Dept. of Electrical & Computer Engineering, Auburn University, Jan. 2004.

[28] S. Gerstendorfer and H. J. Wunderlich, "Minimized Power Consumption for Scan-based BIST," in *Proc. International Test Conf.*, pp. 77 – 84, Sept. 1999.

[29] X. Zhang and K. Roy, "Power Reduction in Test-Per-Scan BIST," in *Proc. International On Line Testing Workshop*, pp. 133 – 138, July 2000.

[30] S. DasGupta, R. G. Walther, and T. W. Williams, "An Enhancement to LSSD and Some Applications of LSSD in Reliability," in *Proc. International Fault-Tolerant Computing Symposium*, pp. 32 – 34, June 1995.

[31] M. A. Gharaybeh, M. L. Bushnell, and V. D. Agrawal, "Classification and Test Generation for Path-Delay Faults Using Single Stuck-at Fault Tests," *J. Electronic Testing: Theory and Applications*, vol. 11, pp. 55 – 67, Aug. 1997.

[32] P. Goel, "An Implicit Enumeration Algorithm to Generate Tests for Combinational Logic Circuits," in *IEEE Trans. on Computers*, vol. C-30, pp. 215 – 222, Mar. 1981.