

# Flip-flop chaining architecture for power-efficient scan during test application

Shantanu Gupta and Tarang Vaish  
Department of Computer Science  
and Engineering, IIT Guwahati  
North Guwahati, Assam - 781039  
gshantanu@gmail.com, tarang@iitg.ernet.in

Santanu Chattopadhyay  
Department of Electronics and  
Electrical Communication Engineering,  
IIT Kharagpur, West Bengal - 721302  
santanu@ece.iitkgp.ernet.in

## Abstract

*Power dissipation in CMOS circuits during test time poses a crucial bottleneck for circuit performance and robustness. The power consumption due to switching activity while scan-in of test vectors and scan-out of responses is of particular concern. In this paper a methodology for scan chain modification and test vector adaptation is proposed to effectively reduce the scan test power consumption by controlling this switching activity. Proposed approach, unlike the many in published literature, does not incorporate re-ordering of scan cells; thus avoiding timing and routing overheads. ATPG software ATALANTA was used for test vector generation. The algorithm was verified for ISCAS'89 benchmark circuits, where it showed as much as 27.3% of reduction in switching activity during scan operations.*

## 1 Introduction

Power dissipation is a crucial issue because of its wide ranged effects on circuit performance and life of circuit under test (CUT). Exceeding peak power limitations of any circuit can cause irreversible damages. During the test, scan-in/scan-out of vector/response causes excessive switching activity. In CMOS circuits average test power consumption is much higher than the normal mode operation and is directly proportional to this switching activity [3]. Several techniques have been proposed to reduce the same. A rather popular technique - addition of extra design for test (DfT) logic - helps in reducing the switching activity but adds area overhead. Many works [3,5] discuss heuristics for test vector re-ordering. Works involving scan cell re-ordering (or scan-latch ordering) [2,7,9] have also been prevalent in the literature; but these face the criticism for their test timing inconsistency and decoder buffer problem. Finally, methods have been explored which involve scan chain partitioning that reduce the scan chain lengths [10] but at the same time induces hardware overheads. In

a nutshell, most of the current day techniques for power aware testing are inflicted with: (1) additional DfT logic overhead, (2) test timing inconsistency due to re-ordering of scan cells, and (3) decoder buffer problems due to re-ordering of scan cells.

Evidently, an approach that does not involve the use of an additional DfT logic will be ridden of problem 1. And avoiding scan cell re-ordering will take care of problem 2 and 3. The proposed approach, named as *Optimized Scan Cell Testing* (OSCT), fits exactly in this description. And thus eliminates all of the aforementioned problems. The algorithm does not interfere with the layout decision and thus it can be optimized separately. At all times fault coverage and test time of the circuit are left unaffected. The work concentrates only upon the reduction of scan chain (flip-flop cells) switching activity as it forms a significant part of the total power consumed during scan-in. The switching activity in the combinational circuit is completely ignored. The rest of the paper is organized as follows, section 2 discusses the conventional scan cell testing method (CSCT). This sets the foundation for explanation of proposed approach named as optimized scan cell testing (OSCT) in section 3, section 4 lists the simulation results obtained for ISCAS'89 benchmark circuits. Section 5 concludes the work, followed by important references.

## 2 Conventional Scan Cell Testing : CSCT

Conventional scan cell testing is a well documented technique. [1] deals thoroughly with the subject at hand. Transitions in the scan cells result from scan-in/scan-out of test vectors/responses. Transitions also result from a *clash* that can be defined as a condition where the MSB of predecessor's response differs from the LSB of successor's vector. The total number of transition count is therefore given by (1) [12]. *Clashes* can therefore be effectively reduced in number by test vector re-ordering. We shall use this observation in section 3.3.

$$Total\ Transitions = \sum (Size\ of\ Chain - Position\ of\ Transition) + Size\ of\ Chain \times Clashes \quad (1)$$

### 3 Optimized Scan Cell Testing: OSCT

In this section we propose a new approach that effectively reduces the number of scan chain transitions during the scan cell testing. The input to the algorithm is a CUT and its corresponding test pattern set. Output is a modified scan cell architecture and a test pattern set that gives optimal reduction in test power consumption. Our approach can fundamentally be divided into three steps. The first step of the algorithm does a modification of the scan chain architecture. This modification neither involves addition of a DFT logic nor does it involve scan cell re-ordering. Second step deals with the customization of test vectors for the modified scan cell architecture. In this step, we basically harness the benefits of effectively specifying the *don't care* bits of test vectors. In third and final step of the algorithm we do a simple test vector re-ordering that does the job of reducing total number of *clashes* and consequently the overall number of transitions. The approach taken in the final step for test vector re-ordering is independent of the previous two algorithmic steps.

#### 3.1 Scan Architecture Modification (SAM)

Conventional way of connecting two consecutive flip-flops in a scan chain involves connecting the Q (output of predecessor) to D (input of successor). We make a simple modification to this approach by allowing  $\bar{Q}$  (negated output of predecessor) connection to D (input of successor). This latter type of connection is selectively done at various positions within the scan chain. During the scan-in/scan-out of vectors/responses, any two differing consecutive bits within the vector/response cause flip-flop transition at every clock tick. These transitions can be reduced in number if we set up the flip-flop interconnections optimally. Any modification of the scan cell architecture will necessitate adaption of test vectors such that after scan-in they take the original form. This is dealt with in section 3.2.2.

Our objective is to make a scan architecture for effective handling of all the given test patterns, i.e. we must consider all test vector-response pairs together and decide connections (at each flip-flop junction) that lead to best possible overall reduction in the number of transitions. The test patterns are pre-determined using a suitable ATPG (ATLANTA in our case). A significant number of bit positions in the vectors/responses remain unspecified, or are *don't*

*care* in nature. Ignoring these *don't care* bits, that are tackled later in section 3.2.1, we can compute the *cost* at every index  $i$  of the scan chain while having 1)  $Q - D$  connection and 2)  $\bar{Q} - D$  connection. The calculation given by (2) decides an optimal connection type for every scan chain index.

$$\begin{aligned} Cost_{10,01}^i &= VBitTotal_{10,01}^i \times i + RBitTotal_{10,01}^i \times (Size\ of\ Chain - i) \\ Cost_{11,00}^i &= VBitTotal_{11,00}^i \times i + RBitTotal_{11,00}^i \times (Size\ of\ Chain - i) \\ FF_{connection}^i &= Cost_{11,00}^i \geq Cost_{10,01}^i \\ &\forall i \in \{1, Size\ of\ Chain - 1\} \end{aligned} \quad (2)$$

In (2),  $i$  is the index of the flip-flop junction in the scan chain, i.e. junction between  $i^{th}$  and  $i + 1^{th}$  flip-flop.  $VBitTotal_{10,01}^i$  is the variable for number of times the consecutive bits differ at position  $i$  when considering all the test vectors.  $RBitTotal_{10,01}^i$  is the variable for number of times the consecutive bits differ at position  $i$  when considering all the test responses.  $VBitTotal_{11,00}^i$  and  $RBitTotal_{11,00}^i$  are similarly defined when consecutive bits are same. For each of these variables we leave out the count for *don't care* bits.  $Size\ of\ Chain$  is the number of flip-flops in the scan chain.  $Cost_{10,01}^i$  stands for the number of transitions that will take place due to consecutive bits at indexes  $i$  and  $i + 1$  while keeping flip-flop connection as  $Q - D$ . Similarly,  $Cost_{11,00}^i$  is similarly defined while flip-flop connection is kept as  $\bar{Q} - D$ .  $FF_{connection}^i$  is assigned a boolean value *true* or *false* whichever favors a lower value for transition cost. *True* imposes a  $Q - D$  connection, whereas a *false* imposes a  $\bar{Q} - D$  connection.

#### 3.2 Test Vector Customization (TVC)

Test vector customization is responsible for the following tasks:

- Taking the optimized scan cell architecture as a reference, it specifies the *don't care* bits in the test vectors.
- It takes the test vectors and adapts them to the optimized scan cell architecture such that after scanning in they assume desired original form.

##### 3.2.1 Specifying the *don't care* bits

ATPGs provide partially specified vectors that can be used to our advantage by customizing them for the optimized scan cell architecture. Therefore, we can do an intelligent fill-up of the *don't care* bits in the test vectors such that they incur lower transitions costs as compared to a situation where the *don't care* bits are randomly specified. The algorithm proposed handles one test vector at a time. We shall

Step	Flip Flop Q-D Connections																Description	
	0	1	1	0	0	1	1	1	0	0	1	0	1	0	0	0		
																	Original Input Vector	
1	1	X	X	0	1	X	X	X	X	0	0	X	0	X	X	X	1	Don't care group
1a	1	X	X	0	1	X	X	X	X	0	0	X	0	X	X	0	1	FF = 0, X = inv(1) = 0
1b	1	X	X	0	1	X	X	X	X	0	0	X	0	X	X	1	0	FF = 0, X = inv(0) = 1
1c	1	X	X	0	1	X	X	X	X	0	0	X	0	1	1	0	1	FF = 1, X = 1
2	1	X	X	0	1	X	X	X	X	0	0	X	0	1	1	0	1	Don't care group
2a	1	X	X	0	1	X	X	X	X	0	0	0	0	1	1	0	1	FF = 1, X = 0
3	1	X	X	0	1	X	X	X	X	0	0	0	0	0	1	1	0	Don't care group
3a	1	X	X	0	1	X	X	X	0	0	0	0	0	0	1	1	0	FF = 1, X = 0
3b	1	X	X	0	1	X	X	0	0	0	0	0	0	0	1	1	0	FF = 1, X = 0
3c	1	X	X	0	1	X	0	0	0	0	0	0	0	0	1	1	0	FF = 1, X = 0
3d	1	X	X	0	1	0	0	0	0	0	0	0	0	0	1	1	0	FF = 1, X = 0
4	1	X	X	0	1	0	0	0	0	0	0	0	0	0	1	1	0	Don't care group
4a	1	X	0	0	1	0	0	0	0	0	0	0	0	0	1	1	0	FF = 1, X = 0
4b	1	0	0	0	1	0	0	0	0	0	0	0	0	0	1	1	0	FF = 1, X = 0
	1	0	0	0	1	0	0	0	0	0	0	0	0	0	1	1	0	Don't care assigned
5	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	Pattern for shifting in

Figure 1. Illustration of Test Vector Customization

explain this algorithm with the help of a simple example, see figure 1. Alphabet X is used to represent a *don't care* bit.  $Q - D$  and  $\bar{Q} - D$  type of flip-flop (FF) interconnections are encoded as 1 and 0 respectively. Following steps summarize the algorithm:

1. Start from last bit (LSB) of the test vector and move toward the first bit (MSB).
2. While moving leftwards in the test vector, identify the first *don't care* bit, call this position **S**. See step 1 in figure 1, **X** is the first *don't care* bit.
3. Consider the right hand immediate neighbor bit of position **S**, call this bit **R**. In our example  $R = 1$  for the first *don't care* bit.
4. Check out the connection type between the flip-flops that correspond to positions **S** and **R** of the test vector.
  - If we have a  $Q - D$  flip-flop connection ( $FF = 1$ ), then assign  $S = R$ .
  - If we have a  $\bar{Q} - D$  flip-flop connection ( $FF = 0$ ), then assign  $S = \text{inverse}(R)$ .

Refer to steps 1a, 1b, 1c, 2a etc. in the figure 1 to get acquainted with this procedure. If the last bit in the test vector is *don't care* in nature, set it to a default binary value.

5. If **S** is the first bit of the test vector, stop the algorithm, otherwise go back to step 2 of this algorithm.

The test vector obtained at step 4b of the figure 1 is completely devoid of *don't care* bits.

### 3.2.2 Adaptation of test vectors

At this stage of the algorithm, we have an optimized scan cell architecture with completely specified test vectors. But the modified scan architecture demands a change in test vectors to nullify the transitions caused by  $Q - D$  connections

during scan-in. This is important as we want the test vectors to be in their original form after complete scan-in. We can handle this problem in a straightforward manner; flip a bit in the original test vector if it goes through an odd number of  $Q - D$  connections, and keep it unaltered otherwise. An example of such an adaptation of a test vector is shown in figure 1. Step 4b in the figure 1 shows a *completely specified* test vector. And step 5 shows an *adapted* test vector that is ready for a scan-in.

### 3.3 Test Vector Reordering (TVR)

In this last and final step of our algorithm, we optimally re-order the test patterns to reduce the scan chain switching transitions further. This essentially removes the *clashes*. Following steps summarize the algorithm for test vector re-ordering:

1. Attach a *tag* XY to every test pattern; where X is LSB of the test vector, and Y is MSB of the output response.
2. Four types of *tags* are possible: 00, 01, 10 and 11. Divide the complete set of test patterns into a maximum of 4 groups on the basis of these *tags*.
3. List all the test patterns in the group 00.
4. Pick test patterns from group 01 and 10 alternately. If any one of these two groups exhausts, list all the remaining patterns from the other group ignoring the alternate picking policy.
5. List all the test patterns in the group 11.

The algorithm shown above is based on a simple observation. If a test pattern A has a tag  $X_A Y_A$ , and another test pattern B has a tag  $X_B Y_B$ ; then A followed by B would cause a *clash* if and only if  $Y_A \neq X_B$ . This follows directly from the definition of a *clash*:  $Y_A$  is MSB of the predecessor's response and  $X_B$  is LSB of successor's vector. In the algorithm we minimize the number of such occurrences.

At the end of three simple steps, we have arrived at an optimized scan cell architecture and suitably customized test vectors that have been optimally ordered to reduce the number of scan chain transitions.

## 4 Simulation Results and Discussions

The complete algorithm was implemented in C and verified on ISCAS' 89 benchmark circuits [4]. The test patterns were generated using ATALANTA [6]. FSIM fault simulator [11] was used to obtain output responses for the customized test vectors. Table 1 summarizes the results obtained. CSCT column lists the number of transitions for complete testing when using approach from section2, SAM & TVC (section 3.1 and section 3.2) column lists the number of transitions obtained after scan architecture modification and test vector customization. The following column (%Imp) shows percentage improvement for this part. The

Circuit	#Gates	Fault Coverage	CSCT	SAM & TVC	% Imp.	TVR	% Imp.	% Imp. [9]
s298	119	100.00%	2926	2702	7.65%	2534	13.39%	10%
s510	211	100.00%	1115	1057	5.20%	913	18.12%	5.3%
s526	193	99.82%	12537	11383	9.21%	10774	14.06%	-
s713	393	93.46%	8826	7712	12.62%	7674	13.05%	17%
s953	395	100.00%	27905	24211	13.24%	24095	13.65%	-
s1238	508	94.91%	22625	19553	13.58%	19193	15.17%	-
s1488	653	100.00%	1953	1953	0.00%	1683	13.82%	-
s9234	5597	93.48%	9775460	7130370	27.06%	7105472	27.31%	22.4%
s15850	9775	96.68%	64559689	50710629	21.45%	50619849	21.59%	-

**Table 1. Simulation results**

second to last column TVR (section 3.3) reflects the number of transitions after application of the complete algorithm, i.e. after test vector re-ordering. The second to last column shows overall saving in scan chain power consumption for various benchmark circuits. [9] is fairly recent work that makes use of scan cell ordering approach to reduce test power and, therefore, has the problems associated with this category of approaches. A comparison with [9] hence forms an ideal ground for this work. The available values from [9] in table 1 show our method to be competitive even against a popular, but overhead ridden, scan cell re-ordering based method.

## 5 Conclusions

In this paper, we have presented a robust and efficient algorithm to reduce the scan test power consumption in CMOS circuits. Our method explores the capability of flip-flops to provide both  $Q$  and  $\bar{Q}$  as outputs simultaneously, thus allowing us to modify the scan architecture for an efficient handling of test patterns. We also exploit the unspecified or the *don't care* bits in a test vector to customize it; thus giving us better performance with the optimized scan cell architecture. Apart from these novel propositions, we also do a simple test vector re-ordering. This is independent of the rest of the algorithm and therefore gives a flexibility to make future modifications and/or to employ another test vector re-ordering scheme published in literature. Proposed method has no penalty on the fault coverage, IC test time or circuit performance. No routing issues as the scan cell remain in their original order. Finally, it is very easy to use in a classical DFT flow and has therefore a very low impact on the system design time.

## References

- [1] A. Crouch, *Design-for-Test for Digital IC's and Embedded Core Systems*, Number ISBN 0-13-08427-1 Prentice Hall, 1999.
- [2] V. Dabholkar and S. Charkravarty, *Techniques for minimizing power in scan and combinational circuits during test application*, IEEE Trans, on Computer Aided Design, 17(12):1325-1333, 1998.
- [3] Devadas and S. Malik, *A survey of optimization techniques targeting low power VLSI circuits*, In Proc. Of Design Automation Conferences, pages 242-247, 2002.
- [4] D.Bryan F.Brglez and K.Kozminski, *Combinational profiles of sequential Benchmark circuits*, IEEE ISCAS, 3:1929-1934, May 1989.
- [5] S. Gerstendorfer and H.J.Wunderlich, *Minimized power consumption for scan-based Bist*, In Proc. IEEE International Test Conference Pages 77-84, 1999.
- [6] H.K.Lee and D.S. Ha, *On the generation of test patterns for combinational circuits*, Technical Reports 12-93, Dept. of Elec. Eng. Virginia Polytechnic Institute and State University.
- [7] I. Bayraktaroglu O. Sinanoglu and A. Orailoglu. *Scan power reduction through test data transition frequency analysis*. In Proc. Of International Test Conference, pages 844-850, 2002.
- [8] C. Laundrault P. Girard, L. Guiller and S. Pravossoudovitch, *A test vector ordering technique for switching activity reduction during test operation*. In IEEE Great Lakes Symp. On VLSI, pages 24-27, 1999.
- [9] C. Laundrault, Y. Bonhomme, P. Girard and S. Pravossoudovitch, *Power driven chaining of flip-flops in scan architectures*. In Proc. IEEE International Test Conference, pages 786-803, 2002.
- [10] Ozgur Sinanoglu and Alex Orailoglu, *A Novel Architecture of Power-Efficient, Rapid Test*, International Conference on Computer-Aided Design (ICCAD '02), pages 299-303, 2002.
- [11] H. K. Lee and D. S. Ha, *An Efficient Forward Fault Simulation Algorithm Based on the Parallel Pattern Single Fault Propagation*, Proc. of the 1991 International Test Conference, pp. 946-955, Oct. 1991.
- [12] R. Sankaralingam, R. Oruganti and N. Touba, *Static Compaction Techniques to Control Scan Vector Power Dissipation*, IEEE VLSI Test Symposium, pp 35-42, 2000.