ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 3, March 2015

Merging of Test Cubes using Test Point Insertion for Low-Power Test Compaction

S.Sindhu[1], M.Sathiskumar [2]

PG Scholar, VLSI Design, P.A. College of Engineering and Technology, Coimbatore, India[1]

Head of the Department, PG-ES, P.A. College of Engineering and Technology, Coimbatore, India[2]

Abstract: Low-power test generation procedure targets the switching activity during the fast functional clock cycles of broadside tests. The procedure is based on Test Cubes merging using Test Point Insertion that it extracts from functional broadside tests. Test cube merging supports test compaction and test point insertion improves fault coverage. The use of functional broadside tests provides a target for the switching activity of low-power tests, not exceeding the switching activity that is possible during functional operation. The use of test cubes that are extracted from functional broadside tests is a unique feature. It ensures that the low-power tests would create functional operation conditions in sub circuits that are defined by the test cubes and test point insertion reduces the complexity involved in detecting additional faults. Experimental results show that the procedure detects all or almost all the transition faults that are detectable by arbitrary (functional and non-functional) broadside tests in benchmark circuits. The simulation results are obtained using MODELSIM 6.3f and the power is analysed using XILINX 8.1 software.

Keywords: Functional broadside tests, low-power test generation, test cubes, test point insertion, transition faults

I. Introduction

Delay faults in standard-scan circuits can be detected by two-cycle tests. In a broadside test, a scan-in operation is followed by two functional clock cycles that activate delay faults and propagate them to observable outputs. The test ends with a scan-out operation. Two-cycle tests in general, and broadside tests in particular, can result in excessive switching activity and power dissipation. Low-power test generation and design-for-testability procedures were developed to address the issue [3]-[6]. These procedures address the switching activity during scan shifting and functional clock cycles. This work focuses on the switching activity during the fast functional clock cycles of broadside tests [2]. The importance of these clock cycles results from the fact that delay faults are detected during these clock cycles, and they are applied under a fast clock. It refers to the switching activity during the fast functional clock cycle of a broadside test simply as the switching activity of the test.

Test compaction procedures that are based on merging of test cubes that are able to accommodate the constraints of test data compression as they produce partially specified tests[11]. The procedure described in generates test cubes for target faults, and then merges them to achieve test compaction. The procedure described in starts from a given fully specified test set. It extracts the test cubes for target faults from this test set, and then merges them. The result is a compacted test set that detects the same set of faults [12]-[16].

The functional broadside test sets are generated by a test generation procedure that is similar to the one used. Functional broadside tests are generated under the assumption that the circuit has a known initial state, from which functional operation starts, and that functional operation consists of the application of primary input sequences. The primary input sequences are assumed to be unconstrained during functional operation. Equivalently, it is assumed that the circuit is designed to operate correctly under any primary input sequence [7],[9]. Functional broadside test generation can accommodate functional constraints on primary input sequences if they are known. To avoid the computational complexity of computing reachable states, the procedure starts from a set R of reachable states that consists only of the known initial state of the circuit. It generates functional broadside tests with scan-in states from R. It identifies additional reachable states from the state transitions that the circuit makes under the tests it generates. Specifically, let $t_i = < s_i, v_{i0}, v_{i1} >$ be a functional broadside test with scan-in state s_i and primary input vectors v_{i0} and v_{i1}. The scan-in state s_i is a reachable state. The primary input vectors $vi0$ and $vi1$ are applied in two consecutive functional clock cycles after s_i is scanned in. Suppose that the primary input vector v_{i0}, takes the circuit from state s_i to a state denoted by s_{i1}. Suppose that the primary input vector v_{i1} takes the circuit from state s_{i1} to a state denoted by s_{i2}. Then s_{i1} and s_{i2} are also reachable states. They are added to the set R of reachable states, and used for computing additional functional broadside tests. The procedure from includes conditions to limit the number of reachable states, which will be considered, yet ensure that they are varied enough to allow detectable target faults to be detected [10].

Test compaction is based on the observation that a test with a higher switching activity tends to detect more faults. The switching activity is defined as the number of lines that make a $0 \rightarrow 1$ or $1 \rightarrow 0$ transition during the second functional clock cycle of a test. The transitions are caused by changing the input patterns to the combinational logic from $s_i v_{i0}$ during the first functional clock cycle to $s_{i1} v_{i1}$ during the second functional clock cycle of the

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 3, March 2015

test t_i .The switching activity of t_i is denoted by swa(t_i). After the procedure generates a functional broadside test t_i , and before adding it to the test set, the procedure checks whether t_i can replace any test t_j that was added to the test set earlier. It performs the check only for tests t_j such that swa (t_j) < swa(t_i). The replacement of an existing test prevents the number of tests from increasing unnecessarily. It also causes the procedure to prefer tests with high switching activity. The test set is compacted further by ordering the tests from high to low switching activity, and applying forward-looking reverse-order fault simulation in this order. This process removes unnecessary tests from the test set. The order of the tests gives preference in retaining tests with high switching activity. From the previous techniques, it is observed that there is excessive switching activity and power dissipation that occurs at the output. The large number of test patterns involves large area for implementation. This in turn provides increased computational complexity.

II. Proposed Method

A. Functional Broadside Tests

The functional broadside test set generated by this procedure is denoted byT_{FB}. The maximum switching activity of any test in T_{FB} is denoted by swa(T_{FB}). During test generation, the procedure may compute functional broadside tests with switching activity that is higher than swa(T_{FB}). It discards these tests if they do not detect any target faults. The maximum switching activity of any functional broadside test that the test generation procedure considers (whether or not it is included in T_{FB}) is denoted by max_swa_func. The value of max_swa_func is used for bounding the switching activity of the tests in the low-power test set that is generated. This bound guarantees that the switching activity will not exceed the switching activity, which is possible during functional operation.

Let F be the set of target faults (transition faults in this paper). For every f ∈ F, let t(f) is the first test in T_{FB} that detects it. The test t(f) for every f ∈ F is obtained by fault simulation with fault dropping of F under T_{FB} as shown in Table I.

TABLE I
FUNCTIONAL BROADSIDE TEST SET

i	t_i	swa(t_i)
0	<010,0001,1111>	20
1	<000,0011,1101>	19
2	<000,0110,1001>	18
3	<101,0010,1011>	18
4	<010,0001,1000>	16
5	<000,1111,0001>	13
6	<000,1011,0110>	13
7	<000,1100,1011>	7

B. Functional Broadside Test Cube

The procedure considers the faults from F one at a time. It obtains a test cube c(f) for f by initially assigning c(f) = t(f). Let c(f) = < s(f), v_0(f), v_1(f) >. Since t(f) is fully specified, all the values of s(f), v_0 (f) and v_1 (f) are initially specified (0 or 1). For a circuit with n primary inputs and k state variables, there are k+2n such values. The procedure considers these values one at a time in a random order, and attempts to unspecify them (a different random order is selected for every fault). When a value is considered, the procedure replaces it with an unspecified value (an x). It then simulates f under c(f). The unspecified value is accepted if f continues to be detected, and additional values are considered with respect to the modified c(f) otherwise, the specified value is restored. After c(f) is obtained, the procedure performs fault simulation with fault dropping of F under c(f). This removes from considering all the faults that are detected by c(f). The test cube c(f) is added to a set denoted by C_{FB} . At the end of the extraction process, C_{FB} = {c0, c1, . . . , cm−1}.

TABLE II
FUNCTIONAL BROADSIDE TEST CUBES

j	f	i	c_j	swa(c_j)
0	1 : 0 → 1	0	<xxx,0xxx,11xx>	5
1	1 : 1 → 0	5	<xxx,11xx,0xxx>	5
2	2 : 0 → 1	0	<0x0,x0x1,11x1>	12
3	2 : 1 → 0	2	<x0x,011x,x0x1>	11
4	3 : 0 → 1	0	<xxx,xx0x,x11x>	1
5	3 : 1 → 0	1	<xxx,xx1x,x10x>	2
6	4 : 0 → 1	2	<x0x,0x10,x0x1>	8
7	4 : 1 → 0	4	<xx0,00x1,10x0>	8
8	5 : 0 → 1	5	<0xx,111x,x0x1>	7
9	5 : 1 → 0	3	<1xx,ox1x,x0x1>	6
10	6 : 0 → 1	6	<000,x0x1,01xx>	8
11	7 : 0 → 1	8	<xx0,x10x,x00x>	2
12	7 : 1 → 0	3	<x01,0x1x,x0x1>	11
13	10 : 0 →1	2	<xxx,x11x,x00x>	5
14	10 :1→ 0	1	<xx0,x0xx,x10x>	5
15	13 :0→ 1	6	<0x0,10x1,01xx>	11
16	13 :1→ 0	0	<01x,0xxx,11xx>	15
17	15 :1→ 0	7	<xxx,x10x,xx1x>	2
18	16 :0→ 1	6	<000,x0x1,0xx0>	5
19	16 :1→ 0	4	<x10,00xx,10x0>	10
20	17 :1→ 0	0	<x1x,0xxx,11x1>	10
21	22 :0→ 1	2	<x0x,011x,10x1>	15

Table II shows a set of test cubes that are obtained through this process for s27 from the test set TFB shown in Table I. The a → a_ transition fault on a line g is denoted by g: a → a_. For 0 ≤ j < 22, column f of Table II shows the fault f, for which the test cube $c_j$was derived and column i shows the index of the test t_i, from which it was derived. Thus, t(f) =t_i.Column swa(c_j) shows the switching activity of the test cube.

Figure 1 shows the values obtained under the two functional clock cycles of the test cube c_2 from Table II, which was extracted from t_0 for the 0 →1 transition fault on line 2. The specified values obtained under c_2 are also obtained under t_0 . Thus, c_2 creates the signal-transitions as t_0. This includes the 0 →1 transition on line 2, and the 1 →0 transitions on line 9.

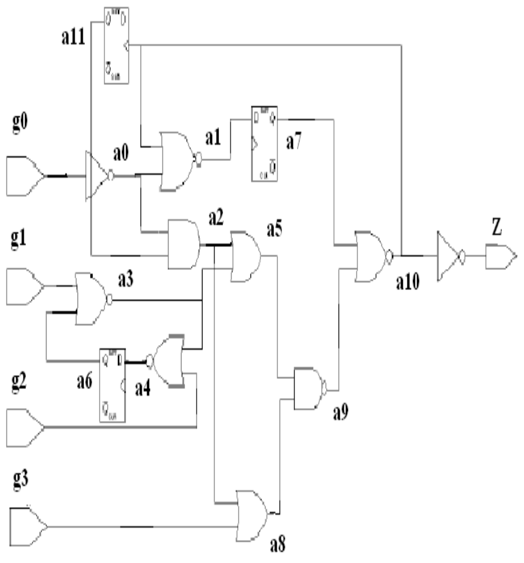

Fig. 1. ISCAS-89 Benchmark s27 circuit

Other lines, such as lines 4, 5, and 19, are prevented from making signal-transitions by c_2 as well as by t_0. All these values can also occur together during functional operation, and they ensure the detection of the 0 →1 transition fault on line 2. When c_2 is merged with other test cubes to form a new test, the test will preserve these values ofc_2, which also occur under t_0 and can occur during functional operation. The switching activity under a test cube c_j is denoted by swa(c_j). It is defined by assuming optimistically that unspecified values will not result in 0 → 1 or 1 → 0 transitions. Thus, swa (c_j) is the number of lines that have specified 0 →1 or 1 →0 transitions. Based on Figure 3.1, swa(c_2)=12.
Therefore, it does not exceed the maximum switching activity that is possible during functional operation. Extracting a test cube for fault in a circuit with n primary inputs and k state variables requires logic simulation of k+2n test cubes that are obtained by unspecifying input values one at a time. Event-driven simulation and structural analysis of the circuit can be used to identify inputs that do not affect the activation or propagation of the fault and the values of these inputs can be unspecified together without logic simulation. These techniques are expected to make the extraction process feasible even when n and k are large. The computation of a set of test cubes C_{FB} for a set of faults F requires at most |F| test cubes to be extracted. In addition, it requires fault simulation with fault dropping of F under C_{FB} . The number of test cubes in C_{FB} is typically significantly lower than |F| since each test cube detects several faults.

The same procedure is applied for s208 and s298 Benchmark circuits.

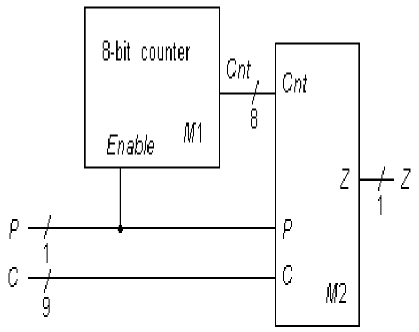

Fig. 2. s208 Benchmark Circuit

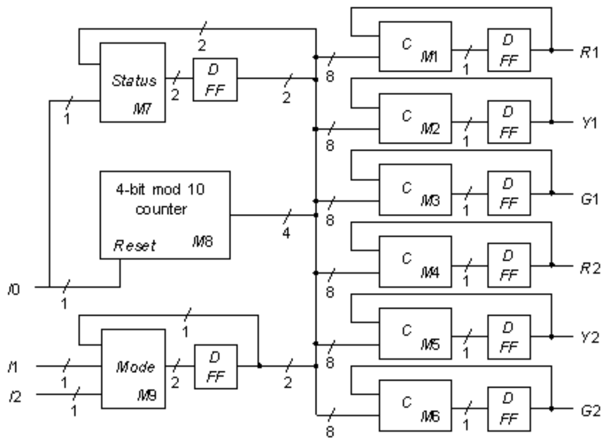

Fig. 3. s298 Benchmark Circuit

C. Test Cube Merging For Faults That Are Detected By Functional Broadside Test

TABLE III
TEST SET OBTAINED BY MERGING OF TEST CUBES

i	t_i	Swa(t_i)
0	<010,0001,1111>	20
1	<xxx,111x,010x>	7
2	<010,0110,1001>	20
3	<x10,00x1,10x0>	11
4	<0xx,111x,x0x1>	7
5	<000,10x1,0100>	14
6	<xx0,x10x,x00x>	2
7	<xxx,x10x,xx1x>	2

i	t_i	Swa(t_i)
0	<010,0001,1111>	20
1	<110,1111,0101>	8
2	<010,0110,1001>	20
3	<010,0001,1010>	17
4	<000,1110,1001>	10
5	<000,10x1,0100>	15
6	<110,1101,0001>	8
7	<101,0100,1010>	9

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 3, March 2015

Removing test cubes that have been marked, the procedure obtains the test set shown in the first part of Table III. The fully specified test set obtained for *s*27 is shown in the second part of Table III.

D. Test Cube Merging Using Test Point Insertion

A heuristic design-for-testability method based on observation point insertion in the Circuit Under Test (CUT) is proposed to increase the error detection ability of Concurrent Checkers (CC). The selection of test points in the proposed scheme is driven by probabilistic fault simulation. Observation points facilitate the propagation of faults, whereas control points change the signal probabilities of gates in their fanout cone and hence change the excitation and propagation of faults. Test point insertion reduces the complexity involved in detecting additional faults.

TABLE IV
TEST CUBE MERGING USING TEST POINT INSERTION

jp	ˆt	Swa(ˆt)	prop
3	<xxx,0xxx,11xx>	11	0
20	<xxx,11xx,0xxx>	10	0
16	<0x0,x0x1,11x1>	15	0
17	<x0x,011x,x0x1>	2	1
15	<xxx,xx0x,x11x>	9	0
2	<xxx,xx1x,x10x>	3	0
4	<x0x,0x10,x0x1>	2	1
21	<xx0,00x1,10x0>	7	0
9	<0xx,111x,x0x1>	3	1
1	<1xx,ox1x,x0x1>	3	1
13	<000,x0x1,01xx>	3	1
6	<xx0,x10x,x00x>	5	0
11	<x01,0x1x,x0x1>	4	1
12	<xxx,x11x,x00x>	4	0
14	<xx0,x0xx,x10x>	4	1
7	<0x0,10x1,01xx>	4	1
8	<01x,0xxx,11xx>	4	1
0	<xxx,x10x,xx1x>	4	1

Table IV shows that for every test cube c_{jp}, the test ˆt obtained by merging c_{jp} with t, the switching activity of ˆt, and whether or not it creates a propagation path. The test t changes only when the switching activity does not exceeds max_swa_func = 23 and a propagation path exists. The final test is fully specified and it detects the fault 6:1 → 0.

The worst-case computational complexity of the procedure is determined as follows. For a fault f ∈ F, the procedure considers m test cubes for merging.

Each test cube requires logic simulation of the fault free and faulty circuits to determine the switching activity and the existence of a propagation path. Thus, an iteration of the procedure requires O(|F|m) passes of logic simulation. The size of F is reduced as additional iterations are performed.

III. SIMULATION RESULTS

A. MERGING OF TEST CUBES FOR S27 BENCHMARK CIRCUIT

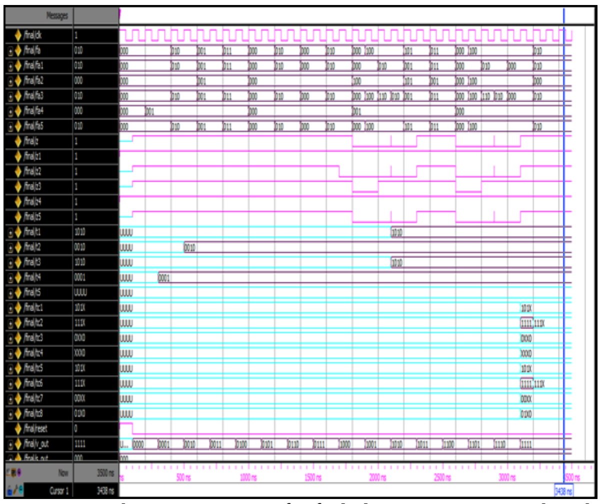

Fig. 4. Test cube merging output for fault detection in s27 Benchmark circuit

Figure 4 shows the merging of test cubes for faults that are detectable by functional and nonfunctional broadside tests for s27 Benchmark circuit. The extraction of test cubes from functional broadside test provides low-power test procedure. The use of test cube merging supports test compaction, and it can be used for accommodating the constraints of test data compression.

B. MERGING OF TEST CUBES USING TEST POINT INSERTION FOR S208 BENCHMARK CIRCUIT

Figure 5 shows the merging of test cubes using test point insertion for faults that are detectable by functional and nonfunctional broadside tests for s208 Benchmark circuit. The extraction of test cubes from functional broadside test provides low-power test procedure. The use of test cube merging supports test compaction and can be used for accommodating the constraints of test data compression.

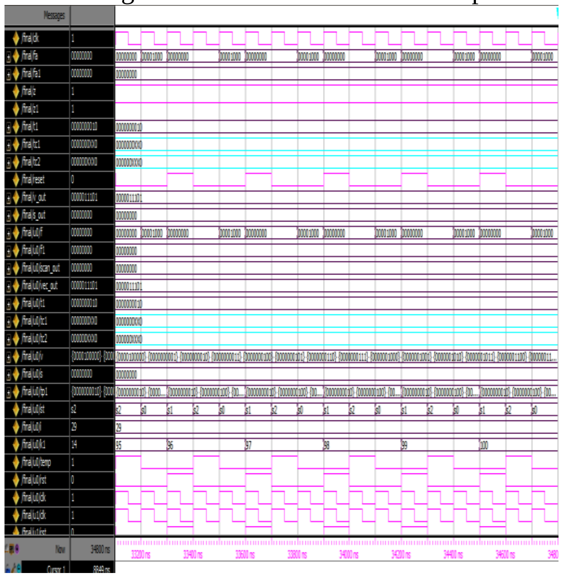

Fig. 5. Test cube merging output for fault detection in s208 Benchmark circuit

C. Merging Of Test Cubes Using Test Point Insertion For s298 Benchmark Circuit

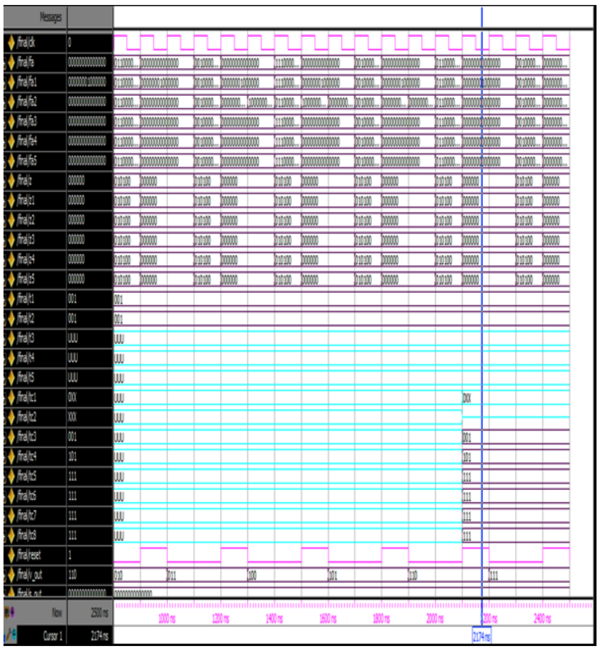

Fig. 6. Test cube merging output for fault detection in s298 Benchmark circuit

Figure 6 shows the merging of test cubes using test point insertion for faults that are detectable by functional and nonfunctional broadside tests for s298 Benchmark circuit. The extraction of test cubes from functional broadside test provides low-power test procedure. The use of test cube merging supports test compaction, and it can be used for accommodating the constraints of test data compression.

D. Comparison Table

TABLE V

COMPARISON OF POWER ANALYSIS FOR DIFFERENT METHODS

TECHNIQUES	Power analysis for Benchmark circuits(mw)		
	s27	**s208**	**s298**
Low-power test vector compaction **[Chandra A. and Chakrabarty K. et al (2002)]**	244	346	385
Scan clock splitting **[Wang S. and Gupta S.K. et al(1994)]**	208	298	364
Shift control techniques **[Xiang D. and Gu S. et al(2003)]**	225	312	392
Power aware test scheduling **[Pomeranz I. (2011)]**	226	345	321
Existing method **[Functional Broadside Tests]**	293	325	383
Proposed method **[Merging of Test Cubes using Test point insertion]**	186	186	228

Table V shows the comparison of power analysis for various methods of low-power testing which involves Test vector compaction, Shift control techniques, Scan clock splitting and Power aware scheduling. Large amount of power is consumed by these methods because of excessive switching activity during test pattern generation.
It is evident that the power consumption during testing is reduced after applying the technique of Merging of test cubes for different benchmark circuits. The use of functional broadside tests provides a target for the switching activity of low-power test. The existing method is 16% more efficient than Shift control techniques and Power aware test scheduling methods and the Merging of Test Cubes Technique is 40% efficient than other low-power testing methods.

IV. Conclusion

This work describes a test generation procedure that extracts test cubes from functional broadside tests, and merges them to form low-power tests. The use of test cube merging supports test compaction and can be used for accommodating the constraints of test data compression. The functional broadside tests provides a target for the switching activity of low-power tests, not exceeding the switching activity that is possible during functional operation. In addition, the use of test cubes, which are extracted from functional broadside tests, ensures that the low-power tests would create signal-transitions that are also possible during functional operation in sub circuits that are defined by the specified values of the test cubes. The procedure generated low-power tests using two merging procedures. The first procedure merged non - conflicting functional broadside test cubes to obtain a smaller test set that detects the same faults as the functional broadside test set. The second procedure is merging of test cubes using test point insertion method in order to reduce the complexity involved in detecting additional faults. Experimental results showed that the procedure detects all or almost all the transition faults, which are detectable by arbitrary (functional and non-functional) broadside tests in benchmark circuits.

References

[1] Chandra A. and Chakravarty K. (2002),'Reduction of SOC test data volume, scan power and testing time using alternating run length codes', in Proc. IEEE Trans.on Design Autom. Conf.,Vol. 34 ,pp. 677-678.
[2] Chou R.M. and Saluja M.M. (1997), 'Scheduling Test for VLSI Systems under Power Constraints', IEEE Trans. on Proc. VLSI Systems, Vol. 5, No. 2, pp. 175-185.
[3] Dabholkar V. and Chakravarty S. (1998),'Techniques for minimizing power dissipation in scan and combinational circuits during test application', IEEE Trans.on Comput.-Aided Design, Vol.17, pp. 1321-1333.
[4] Devanathan V.R. and Ravikumar C.P. (2007),'On power profiling and pattern generation for power-safe scan tests', in Proc. IEEE Trans. on Design Autom.Test Europe Conference, Vol. 33,pp. 1–6.
[5] Kajihara S. and Ishida K. (2002),'Test vector modification for power reduction during scan testing', in IEEE Trans. on VLSI,Vol. 45, pp. 160–165.
[6] Lee K. and Hsu S. (2004), 'Test power reduction with multiplecapture orders', in IEEE Trans. on Design and Autom. Vol. 21, pp. 26–31.

[7] Lee K.J. and Huang T.C. (2000), 'Peak-power reduction for multiple-scan circuits during test application', in Proceedings IEEE Trans. on Comput.-Aided-Design, Vol. 23,pp. 453–458.
[8] Pomeranz I. (2002), 'Scan Shift Power of Functional Broadside Tests', IEEE Trans. on Comput.-Aided Design, Vol. 30, No. 9, pp. 1416-1420.
[9] Pomeranz I. (2011),'Augmenting functional broadside tests for transition fault coverage with bounded switching activity', in Proc.17th IEEE Pacific Rim Int. Symp.Vol. 45, pp. 38–44.
[10] Pomeranz I. (2011),'Signal-transition patterns of functional broadside tests', IEEE Trans. on Computation,Vol. 45, pp.78-89.
[11] Pomeranz I. (2013), 'Functional broadside templates for low- power test generation', IEEE Trans. on VLSI Design, Vol. 11,pp.123-132.
[12] Sankaralingam V. and Oruganti R.R. (2000), 'Static compaction techniques to control scan vector power dissipation', in Proc. 18th IEEE VLSI Test Symp.Vol. 32, pp. 35–40.
[13] Touba N.A. (2006), 'Survey of test vector compression techniques',,IEEE Trans. on Design Test Computation, Vol. 23, No. 4, pp. 294–303.
[14] Wang S. and Gupta X.L. (1994), 'ATPG for heat dissipation minimization during test application', IEEE Trans.on Comput.-Aided Design,Vol. 21, pp. 250–257.
[15] Whetsel D. (2000),'Adapting scan architectures for low power operation',in IEEE Trans. on Testing of VLSI circuits,Vol. 34, pp. 863–872.
[16] Xiang D.and GU S. (2003), 'A cost-effective scan architecture for scan testing with non-scan test power and test application cost',in IEEE Trans. on Design Automation, Vol. 34,pp. 234-238.