A Low Power Deterministic Test Using Scan Chain Disable Technique

Zhiqiang YOU[†a)], Tsuyoshi IWAGAKI[††b)], Michiko INOUE[†††c)], *Members,* and Hideo FUJIWARA[†††d)], *Fellow*

SUMMARY This paper proposes a low power scan test scheme and formulates a problem based on this scheme. In this scheme the flip–flops are grouped into N scan chains. At any time, only one scan chain is active during scan test. Therefore, both average power and peak power are reduced compared with conventional full scan test methodology. This paper also proposes a tabu search–based approach to minimize test application time. In this approach we handle the information during deterministic test efficiently. Experimental results demonstrate that this approach drastically reduces both average power and peak power dissipation at a little longer test application time on various benchmark circuits.
key words: *low power testing, full scan testing, deterministic test, scan chain disable, tabu search algorithm*

1. Introduction

Due to the chip density increasing drastically through the last decade, power dissipation becomes one of the most important factors of very large scale integration (VLSI) design. Furthermore, power and energy consumption of digital systems are considerably higher in test mode than in normal mode [1], [2]. In particular, in the case of scan test, the power dissipation due to clocking all the scan flip-flops is so excessive that it may burn the chip. Hence, many techniques have investigated power minimization or power constraints test.

Test power dissipation depends directly on the global clock frequency and switching transitions of the circuit under test (CUT) [3]. Therefore, decreasing both the clock frequency and the switching activity can reduce test power. The method [4] reduces average power in sequential circuits by decreasing the test clock frequency. The main disadvantages of this method are that the test application time increases as the clock frequency decreases and the peak power cannot be reduced.

The main direction to reduce power is to reduce the switching activity in the circuit. Various techniques have been proposed to reduce switching activity during test. The methodologies in [5]–[8] employ test vector or scan cell reordering technique where test vectors in a test set or scan cells for a test set are reordered for minimal power consumption. The basic idea of these techniques is to find a new order of the test set such that the correlation between consecutive test patterns is increased. The methodologies in [9], [10] also explore the correlation between consecutive test patterns by filling each don't care bit in the test cubes with appropriate value 0 or 1.

There are some methods [11]–[17] that reduce power consumption by using scan chain disabling techniques. Whetsel [11] and Saxena et al. [12] proposed two schemes that divide scan chain into multiple sub-scan chains, and at any time only one sub-scan chain is activated during scan shifting to reduce power consumption. The power during scan shifting is reduced to $1/N$, where N is the number of sub-scan chains. However, these methods did not consider peak power dissipation. During capture cycle, all the sub-scan chains are active. Therefore, the peak power dissipation may be very high. The scheme in [13] employs two different clocks that work at half of the initial frequency such that these two scan chains are operated at different clock cycle during scan shifting. Though this methodology reduces average power efficiently, it suffers the same disadvantage with those in [11], [12]. Bhattacharya et al. [14] proposed a double-tree scan architecture where the scan flip-flops are organized as two complete k-level binary trees whose leaf nodes are merged pair-wise. In this scheme, during scan shifting only the scan flip-flops in a scan path are active. The average power is reduced significantly. But the power dissipation during capture cycle cannot be reduced. In [15], scan chains are grouped into two sets while the given test set T is divided into two subsets. For one test subset except for the test group boundaries, only one-group scan chains are active. Hence, this method can reduce power consumption. However, this method did not consider peak power dissipation. [16] extended the scheme in [15] by exploring a more general architecture. In this method, the test vectors are partitioned into some groups. For each group, while test vectors in the group are applied, a subset of the scan chains are disabled using a programmable scan chain disabling mechanism. This method is effective to reduce average power dissipation. However, to preserve the fault coverage some test vectors may be applied while all the scan chains are active. Therefore, the peak power reduction may not be guaranteed.

To reduce peak power dissipation, Basturkmen et al. [17] proposed a low-power pseudo-random BIST methodology for scan circuits. In this method, the scan chains are partitioned into N groups. At any time, only the scan chains in a group are active throughout scan cycles and capture cycle. Therefore, both average and peak power dissipation are reduced. However, this method, which suffers from a very long test application time, is not efficient for deterministic test.

In this paper, we present a low power deterministic test methodology for sequential circuits using scan chain disable technique. The flip-flops are grouped into N scan chains. At any time, only one scan chain is active during both scan shifting and capture cycles. Since the switch activity of logic will be confined to the fanout cone of the activated scan chain, this technique reduces peak power as well as average power dissipation.

This paper also proposes an approach to group scan flip-flops and test cubes for minimizing test application time. In this method, we group scan flip-flops considering not only the compatibility of some bits of the test cubes but also the test information which flip-flops should capture the test response. If the test for a test vector with scan chain disabling loses fault coverage, we apply this vector again, and activate another scan chain to capture the test response. Therefore, this approach achieves a short test application time while average power and peak power reduction are guaranteed.

This paper is organized as follows. Section 2 presents a low power scan test scheme. The test flow is given in Sect. 3. Section 4 shows a problem based on this scheme. Section 5 describes a flip-flops and test cubes grouping algorithm to minimize the test application time. Some experimental results using our proposed approach are reported in Sect. 6. Section 7 concludes with a brief summary and some future works.

2. Proposed Scheme

In this section, we present a low power scan test scheme shown in Fig. 1. The proposed scheme divides the flip-flops into N groups. In this paper, we do not care of the order of flip-flops in a group. The flip-flops are lined into a scan chain according to the order of appearance in the circuit de-
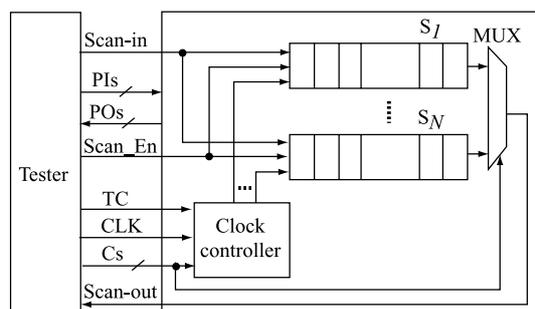
Fig. 1 Proposed scheme.

scription. In this scheme, during a test mode (TC = 1), a clock controller propagates a system clock CLK into only one scan chain selected by Cs. The selected scan chain shifts a test vector or captures a test response according to Scan_En. In a normal mode (TC = 0), CLK is propagated into all the scan chains to enable the normal operation. When TC = 1 and Scan_En = 1, a tester applies Cs and test vectors to CUT. Only the CLK for one scan chain selected by Cs is not gated by the clock controller. Test vectors are shifted into the scan chain while the MUX controlled by Cs selects the activated scan chain to shift out the test responses. When Scan_En = 0, we perform normal operations where all the scan chains are active in conventional full scan methodology. However, in this scheme, normal operations are separated into the operations of capturing test response and normal capture by the extra pin TC. When TC = 0, which means normal captures, CLK is propagated into all the scan chains. Whereas, CLK is propagated into only one scan chain selected by Cs when TC = 1. One scan chain captures the test response of the circuit. Therefore, this scheme can reduce both average and peak power dissipation during scan test. The clock controller can be implemented by a decoder and a small number of gates. The additional hardware elements are a clock controller and a multiplexer. Thus, this scheme has a little higher hardware overhead compared with conventional full scan design.

Though this scheme treats circuits with a single clock domain, it is also applicable to multiple clock domains with a little modification of both clock controller and scan chains. In the case of circuits with multiple clock domains, the clock controller and scan chains are enhanced to handle of the multiple clocks. The skew problem during scan shifting can be resolved by lockup latch insertion technique. If any scan chain consists of the flip-flops that belong to the same clock domain, the skew during shift is minimized and there is no skew during capture which is very difficult to deal with. One-hot clocking scheme, where the scan chains in a clock domain are active during capture operation, efficiently deal with skew problem in multiple clock circuits. Nevertheless, it suffers from a long test application time and high power dissipation.

A test application procedure is summarized as follows. First, some vector is shifted into the scan chains according to the following way. By setting the appropriate value of Cs, the scan chains are active in turn. The bits corresponding to the activated scan chain are shifted into the scan chain. Secondly, by setting appropriate value Cs, only one scan chain captures the test response. If the next vector has the same specified values for the disabled scan chain, Cs keeps the same value. We shift in the bits of vector and shift out test response corresponding to the activated scan chain until the bits in disable scan chain of the vector to be applied are different from that of the previous vector. We continue to apply next vector as shown in the above steps until all the test vectors are applied. Finally, the test response of the last activated scan chain is shifted out.

Notice that other power reduction techniques, such as

test vector reordering, scan cell reordering and minimum transition fill (MT-fill), adapt to this scheme. If we apply one or more these techniques to this scheme, both average power and peak power dissipation can be reduced significantly. This scheme can be enhanced by employing multiple scan trees [18] so that the test application time is reduced drastically.

Though the proposed scheme has these advantages, if the fault effect of a fault cannot propagate to the activated scan chain, the fault response cannot be captured. In this case, to achieve the same fault coverage with conventional full scan testing, we should repeat to apply this test vector while one of the scan chains that can capture its fault effect turns active.

If we apply each test vector N times, and the N scan chains are active in turn among the N same vectors, the test achieves the same fault coverage with conventional full scan design. Although it reduces average power and peak power dissipation, test application time is so long. Fortunately, there are mainly two pieces of information.

i. Usually, several flip-flops capture the fault effect of a fault when a test vector is applied. It is enough to test the fault if only one of the flip-flops is active.

ii. Usually, a fault can be detected by several test vectors in a test set. We can use any one of those test vectors.

If we handle the information efficiently, the test application time can be reduced drastically. Section 5 will describe the approach in detail.

3. Test Flow

This section shows a procedure of test. It consists of the following five steps.

The first step is to perform automatic test pattern generation (ATPG) to generate test cubes, and do static compaction to merge the compatible test cubes together.

In the second step, we obtain the detect-capture information about which primary outputs (POs) and/or pseudo-primary outputs (PPOs) capture the fault effect of a fault when applied one test cube. Figure 2 shows a simple example of a test set with test cubes. Its detect-capture information is given in Table 1. The first row shows the test cubes. The second row describes which faults can be detected by

the corresponding test cube. The flip-flops that can capture the corresponding fault are given in row 3.

In the third step, the flip-flops are grouped into N scan chains. For each test cube, only one scan chain is set to be active. According to the detect-capture information, if a fault cannot be detected, we duplicate some test cube while a scan chain that can capture its fault effect turns active. The test cubes with the information of scan chain activity are divided into some *D-compatible* subsets where the scan chains have the same activity and the bits have non-conflicting values for the disabled scan flip-flops. The detailed information to group flip-flops and test cubes will be shown in Sect. 5. Here, we give a simple example, as shown in Fig. 3 flip-flops 1 and 2 are grouped into Group 1 while flip-flops 3 and 4 are grouped into Group 2. And, the flip-flops corresponding to the bits in the rectangle field are active. For the first cube, flip-flops 1 and 2 are active. Therefore, as shown in Table 1, faults 1, 4, 8 and 9 can be detected. However, it loses fault coverage by only applying the first four cubes, since fault 3 is detected by cube 3 and 4 and its fault effect should be captured by flip-flops in group 1 whereas flip-flops in group 2 are active. We duplicate cube 3, and activate the flip-flops in Group 1 shown as the last cube in Fig. 3. According to the concept of D-compatibility, the test cubes are grouped into three D-compatible subsets shown as Fig. 3.

After that, all X's in the test cube are filled with specified value 0's or 1's. For the X's in the disabled scan chains, we fill them in the way of non-conflicting D-compatibility. For example, the X's in Cube 2 are filled by 0's because the corresponding bits of Cube 1 that is in the same D-compatible subset with Cube 2 are 0's. For the remaining X's, we randomly fill them with 0's or 1's. Therefore, we can randomly fill the X in cube 5 by 0 or 1.

Finally, fault simulation is done to drop any test vector that does not detect any new faults.

We mentioned the problem where we group flip-flops, duplicate and divide test cubes in the third step. This problem is very complex. To clarify this problem, we introduce the following proposition.

Proposition 1: The peak power of this scheme is near $1/N$ of full scan design method, and hardware overhead is a little

Flip-flops	1	2	3	4
	1	1	0	0
	0	0	x	x
	x	1	1	0
	0	1	0	1

Fig. 2 A test set with test cubes.

Flip-flops	1	2	3	4
	group1		group2	
	1	1	0	0
	0	0	x	x
	x	1	1	0
	0	1	0	1
	x	1	1	0

Fig. 3 Flip-flop and test cube groups for Fig. 2.

Table 1 Detect-capture information for the test cubes in Fig. 2.

	Cube 1					Cube 2					Cube 3			Cube 4		
Fault	1	4	6	8	9	2	4	6	7	10	3	5	7	1	3	11
Flip-flop	1, 3, 4	2, 3	4	1	2	2, 3	3	4	4	1	1, 2	3	4	1, 3, 4	2	3

higher than that of multiple scan design while keeping the same fault coverage. ∎

Therefore, the only factor that needs to be optimized is test application time, which is direct to the total test power.

Theorem 1: The test application time (clock cycles) of a scan circuit based on the proposed scheme is: $TAT = M \cdot \lceil F/N \rceil (N - 1) + (n + r + 1) \cdot (\lceil F/N \rceil + 1) - 1$, where, n is the number of original test cubes, F is the number of flip-flops, N is the number of scan chains, M is the number of D-compatible subsets, and r is the number of increased test cubes. ∎

The test application time can be calculated according to the test application procedure shown in Sect. 2 as follows. To shift in the first test vector of each D-compatible subset, it takes $N \cdot \lceil F/N \rceil$ clock cycles. There are M D-compatible subsets. Therefore, the total time is $M \cdot N \cdot \lceil F/N \rceil$. For the remaining $n + r - M$ test vectors, to shift in them it takes $(n + r - M) \cdot \lceil F/N \rceil$ clock cycles. $n + r$ clock cycles are needed to capture test response. And, finally, to shift out the test response of the last activated scan chain, it takes $\lceil F/N \rceil$ clock cycles. The test application time is the sum of the above items. That is, $TAT = M \cdot \lceil F/N \rceil (N-1) + (n+r+1) \cdot (\lceil F/N \rceil + 1) - 1$.

The problem that minimizes test application time is reduced into the problem that minimizes $M \cdot \lceil F/N \rceil (N - 1) + r \cdot (\lceil F/N \rceil + 1)$ since F, N and n are given values. Thus, we formulate the following problem.

4. Problem Formulation

Problem 1: Minimize the test application time of a scan circuit under the proposed scheme. Stating it more formally, given:

- **Input:** a sequential circuit, its detect-capture information, and the number of scan chains N.

  Task:

- **Output:** Multiple(N)-scan chains design with M compatible test sets, that achieves:
- **Objective:** minimizing test application time, i.e. $M \cdot \lceil F/N \rceil (N - 1) + r \cdot (\lceil F/N \rceil + 1)$. ∎

5. Flip-Flops and Test Cubes Grouping

In this section, we introduce the approach to group flip-flops into N scan chains and test cubes into some D-compatible subsets. In this approach, to solve the formulated problem, we use tabu search algorithm to explore the solution space.

5.1 Overview

A tabu search algorithm [19] is a heuristic to find the optimal solution. It starts with some initial solution, and repeats to select the best solution among the candidates that can be obtained by small changes (move) from the current solution.

```
Flip-flops and test cubes grouping algorithm:
/* initial solution generation */
1. extract a set MCC of MCPs
2. extract a set TCC of TCPs
3. GFF_init ← FF grouping from MCC and TCC
4. TCG_init ← Test cube grouping using MCC, TCC and
       GFF_init
5. (GFF_current, TCG_current) ← (GFF_init, TCG_init)
/* tabu-search */
6. repeat {
7.    TAT_current_best ← ∞
8.    for every pair (q_1, q_2) of flip-flops
      from different groups in GFF_current {
9.       if this pair is not in tabu list {
10.          GFF_tmp ← swap q_1 and q_2 for GFF_current
11.          TCG_tmp ← test cube grouping from MCC and
             GFF_tmp
12.          if TAT(GFF_tmp, TCG_tmp) < TAT_current_best {
13.             (GFF_next, TCG_next) ← (GFF_tmp, TCG_tmp)
14.          }
15.       }
16.    }
17.    (GFF_current, TCG_current) ← (GFF_next, TCG_next)
18.    add GFF_current into tabu list
19.    if TAT(GFF_current, TCG_current)
         < TAT(GFF_best, TCG_best) {
20.       (GFF_best, TCG_best) ← (GFF_current, TCG_current)
21.    }
22. }
23. until #iterations>Min{N_itr1, N_itr2}
```

Fig. 4 Flip-flops and test cubes grouping algorithm.

The algorithm maintains a tabu that is a list of solutions that are not allowed to be seleted as the next solution. Our algorithm uses the tabu to avoid selecting the same solution twice, and hence avoid a local optimal solution.

Figure 4 summarizes the flip-flops and test cubes grouping algorithm. Lines 1–5 generate an initial solution. First, we extract a set MCC of mandatory cube-capture pairs (MCPs) from the detect-capture information (line 1). Then, a set TCC of temporary cube-capture pairs (TCPs) to test more faults by several flip-flops with several test cubes is obtained (line 2). The flip-flops are grouped into N scan chains utilizing the MCC and TCC sets, and initial flip-flop groups GFF_{init} are obtained (line 3). After that, we use MCC, TCC and GFF_{init} to group test cubes TCG_{inti} (line 4). The detailed information of above steps is given in the following subsections. Lines 6–23 are the heart of the optimization process. $TAT_{current\_best}$ is set to ∞ (line 7). For every flip-flop pair of flip-flops from different groups, we try every possible move, which is not in the tabu list (lines 8, 9). Here, a move is a term for swapping these flip-flops. After a move, we obtain a new flip-flop group GFF_{tmp}, and a new test cube group TCG_{tmp} (lines 10, 11). If the test application time of the solution† TAT (GFF_{tmp}, TCG_{tmp}) is less than $TAT_{current\_best}$, we potentially set it to (GFF_{next}, TCG_{next}) that will be the current solution in the next generation (lines 12,

---

†A solution is flip-flop groups GFF and test cube groups TCG with established values for test application time that is given in the problem.

13). After the for loop, (GFF$_{next}$, TCG$_{next}$) is assigned to (GFF$_{current}$, TCG$_{current}$), and the corresponding move is then recorded in the tabu list (lines 18, 19). If this solution turns out to be the best one so far, we set (GFF$_{best}$, TCG$_{best}$) = (GFF$_{current}$, TCG$_{current}$). The algorithm ends when either the maximum number of iterations is reached (N_{itr1}), or the maximum number of iterations since the last obtained best solution exceeds some predetermined value (N_{itr2}).

5.2 Cube-Capture Pair Extraction

A cube-capture pair (i, q) consists of a test cube i and a flip-flop q and represents a relation that q captures the response of i. We use cube-capture pairs to get not only an initial solution but also flip-flops and test cubes grouping.

In a test, the fault effect of some fault can be captured by a PO, and some flip-flop should be active to detect some fault when a test cube is applied. This information can reduce the solution space efficiently. Starting from the fault list consists of all the detected faults by the test cubes, we first delete the following faults from the fault list.

1. For each fault f in the fault list, if its fault effect can be propagated to a PO when a test cube exercising, then the fault f is removed from the fault list. This is because even if all scan chains are disabled fault f can be detected by capturing the response at the PO.
2. For each fault f in the fault list, if there exists only one cube i that detects the fault, and only one flip-flop q that captures its fault effect, then record the cube and flip-flop as an MCP (i, q), and remove the fault from the fault list.

Here, the MCP describes the information that the flip-flop should be active once when we apply the test cube.

For example, as shown in Table 1, fault 8 only appears in the columns of Cube 1, and its fault effect can be captured only by flip-flop 1. Hence, there is a MCP $(1, 1)$. Other MCPs about Table 1, are $(1, 2)$, $(2, 1)$, $(3, 3)$ and $(4, 3)$.

3. For each fault f in the fault list, if it can be detected by an MCP (i, q), then we remove the fault from the fault list.

The tabu search algorithm will explore the solution space that is reduced by this step.

The following steps try to detect the remaining faults. The MCPs are not enough to obtain a good initial solution. We extract more cube-capture pairs called TCPs to detect more faults.

4. If only one flip-flop q captures the fault effect of a fault f, we put f into a fault set F_1, and for each test cube i that can detect f we record the cube flip-flop pairs (CFPs) (i, q) into a set CF_1 of CFPs. After that, we obtain the minimum number of CFPs in CF_1 that detect all the faults in F_1, and record them as TCPs.
5. If only one test cube i detects a fault f, we put f into a fault set F_2, and for each flip-flop q that captures its fault effect we record the CFPs (i, q) into a set CF_2 of CFPs. Then, we obtain the minimum number of CFPs in CF_2 that detect all the faults in F_2. After that, we record them as TCPs.

As shown in Table 1, both fault effects of faults 6 and 7 are captured only by flip-flop 4. There is no other fault in F_2. And CFP $(2, 4)$ can detect these faults. Therefore, we record a TCP $(2, 4)$. Fault 2 can be only detected by cube 2. And its fault effect can be captured by flip-flops 2 and 3. Therefore, the CFPs are $(2, 2)$ and $(2, 3)$. Since fault 2 is the only fault in F_2, each of the above CFPs can be the TCP. We randomly select one, such as CFP $(2, 2)$, as a TCP.

The problems in steps 4 and 5 are equivalent to the minimum prime-implicant covering problem, which is known to be NP-hard. We, therefore, use a greedy algorithm, where we select a CFP from CF_1 (CF_2) that can detect the maximum number of faults from F_1 (F_2). We repeat this step until all the faults in F_1 (F_2) are detected.

6. For each remaining fault f, if it can be detected by some TCP, we record it in fault set F_3.

Notice that, TCPs are only used to get an initial solution so that the flip-flops and test cubes grouping algorithm can obtain better solution. When we use the following flip-flops grouping algoritm and test cubes grouping algorithm to get an initial solution, we delete F_1, F_2 and F_3 from the fault list. In the tabu search part, we restore them to the fault list.

5.3 Flip-Flops Grouping

In this subsection, we employ a greedy algorithm, where we group the flip-flops into N scan chains, so that for all the test cubes we assign the maximum number of flip-flops, which have cube-capture pairs for the same test cube, to the same groups. First, we give some concepts.

Definition 1: We denote the number of test cubes where both flip-flops j and k have the cube-capture pairs as a *flip-flop relative degree* (FRD) $w_{j,k}$.

For instance, there are two cube-capture pairs about Cube 1, $(1, 1)$ and $(1, 2)$, and three cube-capture pairs about Cube 2, $(2, 1)$ $(2, 2)$ and $(2, 4)$. Both flip-flops 1 and 2 have the cube-capture pair for cubes 1 and 2. As a result, the FRD $w_{1,2} = 2$. The FRDs of pairs of flip-flops $(1, 4)$ and $(2, 4)$ are 1. Other FRDs are 0.

Definition 2: *Flip-flop test relative graph* (FTRG). Let $G = (V, E)$ be a weighted undirected graph, where each node $v_i \in V$ corresponds to a flip-flop and the weight of the edge between two nodes j and k is an *FRD* $w_{j,k}$.

The greedy algorithm, which is similar to that of [17], groups the flip-flops into N scan chains shown as Fig. 5. An example is shown for $N = 2$ in Fig. 6 to construct an *FTRG* and group the flip-flops.

```
1. for (j=1 to N){
2.     from the unselected nodes in FTRG, select
         the pair with the highest edges weight;
3.     add the pair to group j;
4.     group_j=2;
5.     while(group_j < ⌈F/N⌉){
6.         select a node such that the sum of the weights of the
             edges between the node and the already selected
             nodes in group j is maximum;
7.         Add the node to group j;
8.         group_j++;
9.     }
10.}
11.return the flip-flop groups;
```

Fig. 5   Flip-flop grouping algorithm.

Fig. 6   Flip-flop grouping.

```
1. for (i=1 to n){
2.     N_MCP(i) ← # scan chains, each of them containing at
         least one flip-flop q that is a component of an MCP (i,q);
3.     If N_MCP(i) > 0, extend cube i to CSP (i,k)
         for each k of the N_MCP(i) scan chains;
4.     Else, extend cube i to CSP (i,k), where k is the scan
         chain that the inside flip-flops have the maximum
         number of TCPs (i,q) (in the tabu search part,
         we randomly select one scan chain k);
5. }
6. remove all the detected faults;
7. repeat {
8.     find a CSP (i,k) that detect the maximum # undetected
         faults;
9.      Remove the detected faults from the fault list;
10.}
11.until the fault list is empty;
12.for (m=1 to n+r)
     for (j=m+1 to n+r) {
13.   if there exist D-compatible subsets (DSs) DS_m and DS_j
         if DS_m and DS_j are D-compatible {
14.       merge them into DS_m, remove DS_j;
15.   }
16.}
17.return the D-compatible subsets;
```

Fig. 7   Test cubes grouping algorithm.

5.4 Test Cubes Grouping

After getting the N scan chains, the final phase is to obtain the D-compatible subsets. In this phase, we use the greedy algorithm shown as Fig. 7. First, we extend each test cube i into the cube scan chain pair(s) (CSP(s)) (i,k) where k is an activated scan chain when i is applied as the following way. Let $N_{MCP(i)}$ be the number of scan chains, each of them containing at least one flip-flop q that is a component of an MCP (i,q). If $N_{MCP(i)} > 0$, the test cube i is extended into CSPs (i,k) for each k of the $N_{MCP(i)}$ scan chains. Otherwise, cube i is extended into CSP (i,k) where k is the scan chain that the inside flip-flops have the maximum number of TCPs (i,q) (in the tabu search part, in this case we randomly select one scan chain k). From the fault list, we remove all the faults that can be detected by the CSPs. If there are some remaining faults, we append such a CSP that detect the maximum number of faults among the remaining faults, and delete the detected faults from the fault list. We do the above step until the fault list is empty. Finally, we obtain the D-compatible subsets using the following steps. First, we regard each CSP as a D-compatible subset. For every D-compatible subset, we try every other D-compatible subset. If two of them are D-compatible, we merge them into a D-compatible subset. Until there are no D-compatible subsets can be merged.

For example, we can retrieve the set of test cubes by the following way for the test cubes given in Fig. 2 and the flip-flop groups shown in Fig. 6. There are two MCPs, $(1, 1)$ and $(1, 2)$, about cube 1. All the flip-flops in the MCPs, flip-flops 1 and 2 are in group 1. Hence, there is a CSP $(1, 1)$. As the same way, there are CSPs $(2, 1)$, $(3, 2)$ and $(4, 2)$ for cubes 2, 3 and 4 respectively. Fault 3 remains in

the fault list after removing the detected faults. After we apply $(3, 1)$, fault 3 is detected. That is, all the faults are detected by the five CSPs. At the end, we group the CSPs into some D-compatible subsets. $(1, 1)$ and $(2, 1)$ have the same scan chain activity. And the bits of disabled scan chain in cubes 1 and 2 are compatible. Thus, $(1, 2)$ and $(2, 1)$ are grouped into a D-compatible subset. Though $(3, 1)$ also has the same scan chain activity with $(1, 1)$ and $(2, 1)$, they are not D-compatible due to the conflict of bits of their disabled scan chain. The results are shown in Fig. 3.

Notice that, this flip-flops and test cubes grouping algorithm can also deal with the circuits that already have some sub-scan chains by regarding a sub-scan chain as a scan flip-flop.

6. Experimental Results

We have conducted experiments on full scan version of IS-CAS89 benchmark circuits. In the experiments, we use the ATPG tool "TestGen" of Synopsys to generate test cubes and do fault simulation. Table 2 shows the results of our proposed method compared with previous methods. We do not report the fault coverage in the results because it is the same as that of full scan test.

Columns Ckt. and #ff give the circuit name and the number of scan elements respectively. After them, the columns Time Red. (%) report the test application time reduction of our proposed approach when $N = 2$, $N = 3$ and $N = 4$ compared to the conventional full scan test with one scan chain, where the N scan chains are grouped using the procedure of the last section. Notice that, the test application time of the methods in [11], [12] is the same as that of the conventional full scan test. Hence, these columns also

Table 2 Results of power saving using scan disable technique.

Ckt.	#ff	Time Red. (%)			Power Red. (%) (N = 2)				Power Red. (%) (N = 3)				Power Red. (%) (N = 4)			
					vs. conv.		vs. [11], [12]		vs. conv.		vs. [11], [12]		vs. conv.		vs. [11], [12]	
		N = 2	N = 3	N = 4	Av.	Pk.	Av.	Pk.	Av.	Pk.	Av.	Pk.	Av.	Pk.	Av.	Pk.
s838	32	4.3	27.4	36.1	49.9	50.0	0.1	50.0	65.3	65.6	1.8	65.6	74.0	75.0	1.1	75.0
s953	29	22.4	35.5	37.6	51.8	31.8	6.1	16.7	64.8	54.5	2.1	44.4	72.9	63.6	7.4	55.6
s1196	18	39.0	54.8	60.9	61.8	46.7	38.0	46.7	72.7	60.0	28.2	60.0	75.6	66.7	19.6	66.7
s1238	18	35.8	54.4	60.7	53.0	42.9	11.0	42.9	70.2	57.1	23.9	57.1	75.6	64.3	16.4	64.3
s1423	74	−17.7	−18.2	−26.2	53.4	34.9	7.9	28.2	67.8	53.5	6.4	48.7	74.4	65.1	3.0	61.5
s9234	211	−15.0	−35.5	−40.6	48.6	15.2	−0.3	13.1	64.7	43.2	−0.8	41.8	73.2	57.6	−0.6	56.6
s13207	669	47.6	45.4	42.5	45.7	48.6	−5.4	48.6	63.7	65.2	−5.7	65.2	73.6	73.6	−3.1	73.6

Table 3 Comparison of test application time with random flip-flop grouping algorithm.

Ckt.	Time Red.(%)		
	N = 2	N = 3	N = 4
s838	10.1	23.0	25.2
s953	13.6	26.9	32.0
s1196	6.6	11.3	15.6
s1238	9.7	15.3	20.7
s1423	26.5	37.7	41.2
s9234	17.3	19.3	24.7
s13207	2.9	4.1	6.6

Table 4 Comparison of average power reduction with [15], [16].

Ckt.	Proposed Approach			[15]	[16]
	N = 2	N = 3	N = 4		
s9234	48.6	64.7	73.2	19.1	28
s13207	45.7	63.7	73.6	43.4	33

show the test application time reduction compared with the methods in [11], [12]. The following columns show the reduction in average power and peak power for $N = 2$, $N = 3$ and $N = 4$ compared to conventional full scan test with one scan chain, and the approach in [11], [12] separately. In this experiment, we use the technique of weight transition count described in [9] to estimate power. Therefore, the columns Power Red. (%) in Table 2 describe the percentages reduction of the weighted transitions with previous methods.

In this table, compared with the conventional full scan test, when $N = 2$, $N = 3$ and $N = 4$, the average power is reduced up to 61.8%, 72.7% and 75.6% respectively. The peak power dissipation is also reduced drastically up to 50.0% when $N = 2$, up to 65.6% when $N = 3$, and up to 75.0% when $N = 4$. In comparison with the method in [11], [12], the peak power is reduced. The maximum reduction ratios in peak power dissipation are the same with that of the conventional full scan test though they are sometimes a little smaller. And the average power is reduced or comparable. For test application time, except S1423 and S9234, our method can achieve better results than that of the other two methods. For example, applying test to S1196 the test application time is reduced by 60.9% when $N = 4$. The test application time of S1423 and S9234 are a little longer. This is because in these circuits many test vectors need to be applied several times to preserve the fault coverage of the test cubes.

Table 3 displays the test application time reduction compared with random grouping flip-flops approach. In this random grouping algorithm, the information of cube-capture pairs and test cubes grouping algorithm in our approach are utilized. We run the algorithm ten times, and select the best solution as the comparison. Given in Table 3, the test application time has up to 41.2% reduction.

Therefore, our approach is more efficient to reduce test application time compared with the random flip-flop grouping algorithm.

The comparison of average power reduction with [15], [16] is given in Table 4. Columns 2, 3 and 4 show the percentages of average power reduction for $N = 2$, $N = 3$ and $N = 4$ respectively, which are also given in Table 2. The percentages of average power reduction of the methods in [15], [16] are displayed in columns 5 and 6. The purpose of both methods in [15], [16] that employ two flip-flops and test cubes grouping algorithms is to reduce average power dissipation. As described before, our methodology is mainly for peak power reduction. However, Table 4 demonstrates our method is more efficient to reduce power compared with these methods.

From Tables 2–4, we can conclude that the proposed method is efficient in reducing both average power and peak power dissipation during test without loss the fault coverage. This method is also efficient to reduce test application time that is one of the most important factors in the test.

7. Conclusions

This paper proposes a low power scan test scheme. In this scheme, both average power and peak power reductions are achieved by activating only one scan chain during scan test. To minimize test application time as well as the total test power, this paper also formulates a problem based on this scheme. A tabu search-based algorithm is presented to solve the problem. Experimental results show that the proposed approach is more efficient in reducing average power and peak power dissipation at a little longer test application time compared to the full scan test methodology.

There are still some rooms to reduce test application time. Future work will investigate new approach based on the proposed scheme to reduce test cost. Furthermore, it will be very practical to consider layout impact to our methodology.

Acknowledgments

References

[1] Y. Zorian, "A distributed BIST control scheme for complex VLSI devices," Proc. IEEE VLSI Test symposium, pp.4–9, 1993.

[2] P. Girad, "Survey of low–power testing of VLSI circuits," IEEE Des. Test Comput., vol.19, no.3, pp.82–92, May 2002.

[3] K. Roy and S. Prasad, Low-power CMOS VLSI circuit design, John Wiley & Sons, 2000.

[4] H. Vranken, T. Waayers, H. Fleury, and D. Lelouvier, "Enhanced reduced-pin-count test for full-scan design," Proc. IEEE International Test Conference, pp.738–747, 2001.

[5] S. Chakravarty and V. Dabholkar, "Two techniques for minimizing power dissipation in scan circuits during test application," Proc. IEEE Asian Test Symposium, pp.324–329, 1994.

[6] V. Dabholkar, S. Chakravarty, I. Pomeranz, and S.M. Reddy, "Techniques for minimizing power dissipation in scan and combinational circuits during test application," IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst., vol.17, no.12, pp.1325–1333, 1998.

[7] P. Flores, J. Costa, H. Neto, J. Monterio, and J. Marq1ues-Silva, "Assignment and reordering of incompletely specified pattern sequences targeting minimum power dissipation," Proc. International Conference on VLSI Design, pp.37–41, 1999.

[8] Y. Bonhomme, P. Girard, L. Guiller, C. Landrault, and S. Pravossoudovitch, "Efficient scan chain design for power minimization during scan testing under routing constraint," Proc. IEEE International Test Conference, pp.488–493, 2003.

[9] R. Sankaralingam, R.R. Oruganti, and N.A. Touba, "Static compaction techniques to control scan vector power dissipation," Proc. IEEE VLSI Test Symposium, pp.35–40, 2000.

[10] K.M. Butler, J. Saxena, T. Fryars, G. Hetherington, A. Jain, and J. Lewis, "Minimizing power consumption in scan testing: pattern generation and DFT techniques," Proc. IEEE International Test Conference, pp.355–364, 2004.

[11] L. Whetsel, "Adapting scan architectures for low power operation," Proc. IEEE International Test Conference, pp.863–872, 2000.

[12] J. Saxena, K.M. Butler, and L. Whetsel, "An analysis of power reduction techniques in scan testing," Proc. IEEE International Test Conference, pp.670–677, 2001.

[13] Y. Bonhomme, P. Girard, L. Guiller, C. Landrault, and S. Pravossoudovitch, "A gated clock scheme for low power scan testing of logic ICs or embedded cores," Proc. IEEE Asian Test Symposium, pp.253–258, 2001.

[14] B.B. Bhattacharya, S.C. Seth, and S. Zhang, "Double-tree scan: a novel low-power scan-path architecture," Proc. IEEE International Test Conference, pp.471–479, 2003.

[15] R. Sankaralingam, B. Pouya, and N.A. Touba, "Reducing power disspation during test using scan chain disable," Proc. IEEE VLSI Test Symposium, pp.319–324, 2001.

[16] R. Sankaralingam and N.A. Touba, "Reducing test power during test using programmable scan chain disable," The First IEEE International Workshop on Electronic Design, Test and Applications, pp.159–163, 2002.

[17] N.Z. Basturkmen, S.M. Reddy, and I. Pomeranz, "A low power pseudo-random BIST technique," Proc. IEEE International Conference on Computer Design, pp.468–473, 2002.

[18] K. Miyase, S. Kajihara, and S.M. Reddy, "Multiple scan tree design with test vector modification," Proc. IEEE Asian Test Symposium, pp.76–81, 2004.

[19] F. Glover and M. Laguna, "Tabu search," in Modern Heuristic Techniques for Combinatorial Problems, ed. C.R. Reeves, McGraw-Hill Book Company, 1995.

Zhiqiang You received the B.Sc. degree in applied mathematics and M.E. degree in computer science from Hunan University, China, in 1995 and 2002 respectively, and Ph.D. degree in information science from Nara Institute of Science and Technology, Nara, Japan, in 2006. Presently, he is an assistant professor at Software School, Hunan University. His research interests include low power testing, low cost test and Boolean process. He is a member of IEEE.

Tsuyoshi Iwagaki received the B.E. degree in electronic engineering from Osaka Institute of Technology, Osaka, Japan, in 2000, and the M.E. and Ph.D. degrees in information science from Nara Institute of Science and Technology, Nara, Japan, in 2002 and 2004 respectively. Presently, he is an Associate at School of Information Science, Japan Advanced Institute of Science and Technology. His research interests are VLSI CAD, design for testability and test generation. He received the fourth IEEE Workshop on RTL and High Level Testing Best Paper Award in 2004, and IEEE Kansai Section Student Paper Award in 2005. He is a member of IEEE and IPSJ.

Michiko Inoue received her B.E., M.E, and Ph.D degrees in computer science from Osaka University in 1987, 1989, and 1995 respectively. She worked at Fujitsu laboratories Ltd. from 1989 to 1991. She is an associate professor of Graduate School of Information Science, Nara institute of Science and Technology (NAIST). Her research interests include distributed algorithms, parallel algorithms, graph theory and design and test of digital systems. She is a member of IEEE, the Information Processing Society of Japan, and Japanese Society for Artificial Intelligence.

Hideo Fujiwara received the B.E., M.E., and Ph.D. degrees in electronic engineering from Osaka University, Osaka, Japan, in 1969, 1971, and 1974, respectively. He was with Osaka University from 1974 to 1985 and Meiji University from 1985 to 1993, and joined Nara Institute of Science and Technology in 1993. In 1981 he was a Visiting Research Assistant Professor at the University of Waterloo, and in 1984 he was a Visiting Associate Professor at McGill University, Canada. Presently he is a Professor at the Graduate School of Information Science, Nara Institute of Science and Technology, Nara, Japan. His research interests are logic design, digital systems design and test, VLSI CAD and fault tolerant computing, including high-level/logic synthesis for testability, test synthesis, design for testability, built-in self-test, test pattern generation, parallel processing, and computational complexity. He is the author of Logic Testing and Design for Testability (MIT Press, 1985). He received the IECE Young Engineer Award in 1977, IEEE Computer Society Certificate of Appreciation Award in 1991, 2000 and 2001, Okawa Prize for Publication in 1994, IEEE Computer Society Meritorious Service Award in 1996, and IEEE Computer Society Outstanding Contribution Award in 2001. He is an advisory member of IEICE Trans. on Information and Systems and an editor of IEEE Trans. on Computers, J. Electronic Testing, J. Circuits, Systems and Computers, J. VLSI Design and others. Dr. Fujiwara is a fellow of the IEEE, a Golden Core member of the IEEE Computer Society and a fellow of the Information Processing Society of Japan.