IEEE 4th Workshop on RTL and High Level Testing (WRTLT'03), pp.125-127, Nov. 2003.

Test Length Minimization under Power Constraints for Combinational Circuits

Hao Wu
Nara Institute of Science and Technology,
College of Computer and Communication, Hunan University
hao-wu@is.aist-nara.ac.jp

Zhiqiang You
Nara Institute of Science and Technology
you-z@is.aist-nara.ac.jp

Michiko Inoue
Nara Institute of Science and Technology
kounoe@is.aist-nara.ac.jp

Hideo Fujiwara
Nara Institute of Science and Technology
fujiwara@is.aist-nara.ac.jp

Abstract

In this paper, we formulate a problem to obtain the shortest sequence of test vectors while keeping fault coverage and satisfying peak power constraints. There are mainly two techniques to reduce the peak power or average power for test vector reordering: one is repetition of test vectors; another is addition of new vectors. Using these techniques to reduce the peak power, we present two algorithms to obtain the solution of the problem; one is a heuristic algorithm for traveling salesperson problem (TSP) and the other is a tree travel process (TTP) algorithm.

Key words: Test length minimization, power constraints, combinational circuits, reordering

1. Introduction

Growing size of VLSI circuits, along with the high transistor density, is making minimization of power dissipation an important issue in VLSI design. The dominant component of total power dissipation is attributed to dynamic power dissipation caused by switching of the gate outputs [1]. Therefore, for combinational circuits, power is dissipated mainly when the input vector is changed. Power dissipation during test is dependent on the order of the test vectors. There is an approach for minimizing power based on test vector reordering [2].

Power dissipation during test is more than during normal operation. Hence, test application under power constraint is required. It is important to obtain the shortest sequence of test vectors such that the power does not exceed the allowable maximum power. There are mainly two techniques [3] to reduce the peak power or average power for test vector reordering: one is repetition of test vectors; another is addition of new vectors.

In this paper, we formulate a problem to obtain the shortest sequence of test vectors while keeping fault coverage and satisfying peak power constraints. Furthermore, using above two techniques to reduce the peak power, we also present two algorithms to obtain the solution of the problem.

2. Problem formulation

Power Constraint Test Length Minimization Problem:

Input

-A combinational circuit at gate level

-T: a set of test vectors

-P_{max}: A constraint on peak power

Output

-The shortest sequence of test vectors which includes all the vectors in T and satisfies the peak power constraint Pmax

3. Overview of methodology

We introduce a graph called "Power Constraint Graph (PCG)". Let PCG=(V, E) be an undirected graph, where each node $v_i \in V$ corresponds to an input vector. If the power dissipation between two vectors satisfies the allowable power, there is an edge between the nodes corresponding to the two vectors; otherwise, there is no edge between them.

Filter_Process: For a PTG,

e_{ij}: Power consumed incident two patterns T_i,T_j applied successively

v_{ij}: Test-power from T_i to T_j; $v_{ij}=v_{ji}$.[1]

If $v_{ij}>p$, then drop e_{ij},e_{ji};

A shortest path in PCG, which traverses at least all the nodes of *T*, corresponds to a shortest test sequence which includes all the vectors in T and satisfies the peak power constraint. Therefore, the power constraint test length minimization problem introduced in the previous section can be reduced to the problem of finding a shortest path in PCG which traverses at least all the nodes of T.

Since $|V|=2^p$ where *p* is the number of primary inputs of the given circuit, it is intractable to obtain the whole graph of PCG by computing all the power dissipation between all the nodes in PCG. Therefore, we deal with a sub-graph of PCG as follows.

Let $PCG_T=(T, E')$ be a sub-graph of PCG, which only contains all nodes corresponding to the test vectors set *T*.

If the sub-graph PCG_T is unconnected, we cannot obtain a test sequence which traverses all the nodes of T. In this case, we need to add some new test vectors from outside of T. Furthermore, even if PCG_T is connected, adding some new test vectors may decrease the test sequence length. Therefore, we augment the sub-graph PCG_T by adding some nodes (for new test vectors) and corresponding edges from PCG so that augmented sub-graph contains a shortest test sequence.

The procedure that finds additional new test vectors is called Find_Path. By using the Hamming distance, we can find new test vectors between the two vectors such that the peak power constraint is satisfied.

Find_Path Process:

Hd(i,j):hamming distance between vectors V_i,V_j.

$\forall V_i, V_j \in TPG_t$,by the following process:

step=2

repeat

repeat

Select $V_{k1}\ldots V_{km}$(m=step), which $Hd(k_1,i)=\ldots=Hd(k_m,j)=Hd(i,j)/step$

Calculate $e_{ik1},\ldots,e_{jkm}$

Untill $e_{ik_1} \le P, \cdots, e_{jk_m} \le P$ or all V_k which $Hd(k_1,i)=\ldots=Hd(k_m,j)=Hd(i,j)/step$ be selected

If all V_k which Hd(k,i)=Hd(k,j)=Hd(i,j)/step be selected,step=step+1

Until Hd(i,j)<step or $e_{ik_1} \le P, \cdots, e_{jk_m} \le P$

If Hd(i,j)<step then return step=∞ else return $V_{k_1}, \cdots, V_{k_m}$ and step.

After the Find_Path procedure, we can get another sub-graph of PCG described below. $PCG_{T*}=(T^*, E'')$, where T^* is a set of nodes which represent the original test vectors and the new additional test vectors mentioned above.

Two algorithms employed to find the shortest sequence are presented in this paper. One is a heuristic algorithm for traveling salesperson problem (TSP); the other is a tree travel process (TTP) algorithm.

4. Algorithm for Traveling Salesperson Problem

From the $PCG_{T*}=(T^*, E'')$, we generate a graph $PCG_D=(T, E''', W)$ with respect to T as follows. PCG_D is a complete undirected graph, where each node $t_i \in T$ represents a test vector, each edge $(t_i, t_j) \in E'''$ represents the shortest path between the pair of nodes (t_i, t_j) and the weight ($\in W$) of each edge is the length of the path.

A shortest path in PCG_D which traverses all the nodes of PCG_D is a shortest path in PCG which traverses all the nodes of T. This problem can be reduced the Traveling Salesperson Problem. Since the TSP is NP-hard, we have to consider a heuristic algorithm to solve it. There have been reported many methods to reduce the complexity [6]. Here we give a heuristic algorithm with two assumptions.

Assumption 1(m steps Markov Assumption) In TSP, one hop is only related to m frontiers.

Assumption 2 If a path from t_i to t_j is a part of the best path, it is at least the k-th best paths in all paths from t_i to t_j, where k is a given integer and t_i, t_j are different nodes.

A sequence (which visit Vi once and only once) will be called as"the best sequence" if $\sum P_{i,i+1}$ is minimized.

The algorithm is described as follows.

TSP Process

•1.Start from any V_i;(For example,V_1),L=1.

•2.Vertexes (which do not appear in front of i steps) are sorted by the weight of $\sum_{k=0}^{m} P_{i+k,i+k+1}$,select the above k vertexes.

•3.Chose k best sequence, L=L+k;

•4.repeat 2,3.until L>=n

•5.Find the best sequence from $s_i=(V_1,\ldots\ldots V_n)_i$.

•6.Find the best sequence from $(s_1,s_2,\ldots\ldots,s_k)$.

Since the algorithm starts at each node, the complexity of this algorithms is $O(n^2k^m)$, where k, m are given integer and *n* is the number of test vectors. We except that the above two assumptions will be satisfied for most cases.

5. Tree travel process algorithm

Tree travel process does not need a complete graph. It works on PCG_{T*}.

Theorem 1 :If e_{ij} is an edge of shortest routine which visit all nodes at least once. e_{ij} or e_{ji} will be passed at most twice.

Proof: If e_{ij} or e_{ji} will be passed three times or more and it's edges of the shortest routine, we only consider the last three times.

The shortest routine consists of r_0,j→i,r_1,i→j,r_2,j→i,r_3.

We can find another routine consists of r_0, r_2, j→i ,r_1,r_3. It's clear the second routine is shorter than the first one. The first one is not the shortest one. #

Corollary 1: For a connected graph, visiting all nodes at least once, the shortest routine is at most 2*m-l steps.(Where m is number of edges, l is the length of the longest path, the longest path is a path which node and edge are not repeated).

Theorem 1 and corollary 1 indicate thatfor connected graph we can reduce the problem to find the longest path and the number of edges is little enough. For unconnected graph, byFind_Path Process we can connect all nodes in one connected graph.

If a connected graph is a tree, the number of edges is the least and we can easily find the longest path. The algorithm is worked on a connected tree which is sub-graph of the connected graph. The result is not always the best, but it approaches to the best.

Even if the tree is a connected tree, adding some new test vectors may decrease the length. There are three useful corollaries to decide whether additional nodes are necessary.

i: index of sub-tree

m_i: number of edges of sub-tree i;

d_i: the maximum depth of sub-tree i;

l_i: the longest length of sub-tree i;

Tr_i:sub-tree i;

Corollary 2:Two sub-tree Tr_i,Tr_j are not first and last sub-tree, if additional nodes can be found to connect two sub-tree and satisfies P, adding them will reduce the travel steps, if it satisfies: $l_i+l_j>l_{steps}$.(which l_{steps} is the steps by Find_Path_Process).

Corollary 3: Two sub-tree Tr_i,Tr_j are first and last sub-tree, if additional nodes can be found to connect two sub-tree and satisfies P, adding them will reduce the travel steps, if it satisfies: $l_i+l_j +2>d_i+d_i+ l_{steps}$.

Corollary 4: Two sub-tree Tr_i,Tr_j ,one is first or last sub-tree, another is not. if additional nodes can be found to connect two sub-tree and satisfies P, adding it will reduce the travel steps, if it satisfies: $l_i+l_j+2 >d_i+ l_{steps}$.

The algorithm is described as follows.

TTP Process:

1. Based on the depth first search technique, travel nodes represented all original test vectors to obtain some spanning trees,Tr_1,……Tr_t.
2. Connect all trees by addition of new vectors in PCGT*, and obtain one spanning tree with some sub-trees.
3. Connect leaves of different sub-trees to reduce the length of traveling the whole tree by corollary 2,3,4.
4. Travel the whole tree and obtain a sequence of vectors.

The complexity of this algorithm is $O(t^2 * n^2)$, where *t* is the number of sub-tree and *n* is the number of test vectors.

6. Conclusions and future work

This paper formulated a problem to obtain the shortest sequence of test vectors while keeping fault coverage and satisfying peak power constraints. Furthermore, using two techniques to reduce the peak power, we also presented two algorithms to obtain the solution of the problem. We also discussed the conditions that solution exists and a method that finds additional new vectors.

Some techniques to reduce peak power dissipation for scan sequential circuits have been presented in [5, 6]. Our future work is to apply our proposed method for combinational circuits to scan designed sequential circuits.

Acknowledgement

The authors wish to thank Prof. Dong Xiang of Tsinghua University for his valuable comments.

References

[1] K. Roy and S. Prasad, "Low-power CMOS VLSI Circuit Design", John Wiley & Sons, 2000.

[2] P. Girard, L. Guiller, C. Landrault and S. Pravossoudovitch, "A test vector ordering technique for switching sctivity reduction during test operation", in Proc. IEEE Great Lakes Symposium on VLSI, pp. 24-27, 1999.

[3] V. Dabholkar, S. Chakravarty,"Minimizing Power Dissipation in Combinational Circuits During Test Application",Technical Reports at Department of Computer ScienceandEngineering,Universityat Buffalo,April 15,1994

[4] D.S.Johnson and L.A.McGeoch,"The Traveling Salesman Problem:A Case Study in Local Optimization,"in Local Search in Combinatorial Optimization,E.H.L.Aarts and J.K.Lenstra(eds.),John Wiley and Sons,1996.

[5] S. Chakravarty, V. Dabholkar, "Minimizing Power Dissipation in Scan Circuits During Test Application", in Proc. IEEE Workshop on Low Power Design, pp. 51-56, 1994.

[6] N. Nicolici and B. AI-Hashimi "Power-Constrained Testing of VLSI Circuits", Kluwer Academic Publishers, 2003.