A Technique for Reducing Power Consumption by Test Vector Ordering

Shuanghua Huang, Xiaomin Li
Navy University of Engineering, Wuhan 430033, China
lxm_h@163.com

Abstract. This paper considers the issue of high power dissipation in test mode, and puts forward a novel technique based on the reordering of test vectors. This approach considers both the circuit structure and test set. The proposed approach uses a weight to quantify the relationship between the switching of each input and the internal switching activity, then reorder test set based on these weights. Results of experiments show reductions of the switching activity ranging from 17.06% to 69.22% during external test application.

Keywords: VLSI, power consumption, Test vector ordering

1 Introduction

With the development of the deep submicron and even ultra-deep sub-micron technology, the complexity of integrated circuits increases while area decreases, the frequency increases and power consumption become an important design parameter. Various low-power technologies are used in integrated circuit design, testing is an essential step in IC production, power consumption in test mode can reach twice the normal operating mode. When the tested circuit is during the testing, the power consumption is too large may cause damage to the chip, reduce the test reliability and the use of time or cause another other issues. Reducing the test consumption became an important goal in test design.

Test consumption can be divided into static and dynamic power consumption, test vectors sorting primarily affects dynamic power consumption. Dynamic power is caused by the state transitions of each gate in the circuit, the test vector sequence determines the dynamic power consumption when the test vector has been determined, and test vectors without optimization may make test consumption too large. Now there are two main methods to sort the test vectors, literature[1] proposes the method of Hamming distance reordering, which shows that the turnover number of the circuit internal state and input vectors of Hamming distance are directly proportional , the smaller the Hamming distance is , the smaller the turnover number of the circuit internal state becomes, so it is obviously not comprehensive to use the Hamming distance between test vectors to optimize the test vector sequence to make test set minimum Hamming distance without the consideration of the internal structure of the circuit; Literature [2]counts circuit internal status of all nodes when
the circuit plus test vectors, then sorted the vectors according to these states, so that the turnover number of the circuit internal will be minimum, this method is effective in smaller circuits, but now circuits become more and more complicated, the number of internal nodes (ntotal) is very large, for n test vectors, n * (n-1) times logic simulation[3] is necessary for circuits, and the time in calculating the distance between two vectors is ntotal /n times as long as which in the Hamming distance, all of those causing the test time is too long.

2 Consumption Model

Now the power consumption of CMOS circuit mainly comes from the consumption consumed by charging and discharging of the logic gate state changes in circuits, when the test vectors $V_1$ become $V_2$, the circuit power consumption $P$ [4] is

$$P = 0.5 \cdot (V_{dd}^2 / T) \sum_{i=1}^{N} [f_i(V_1) \oplus f_i(V_2)] \cdot \text{fanout}(i) = K \cdot P_0$$

where $V_{dd}$ represents the circuit supply voltage, $T$ is the circuit's clock period, $N$ represents the number of gates in the circuit, $i$ is one of the doors, $f_i(V_1), f_i(V_2)$ respectively represent for Boolean value of gate $i$ when $V_1$ and $V_2$ input, $\text{fanout}(i)$ represents for the fan-out of gate $i$, where $0.5 \cdot (V_{dd}^2 / T)$ is constant and it has nothing to do with the input vector, it can be seen that the power $P$ is proportional to $P_0$ and $P_0$ is the accumulated value of product when Boolean function value XOR all gates of circuit then multiplied by the number of fan-out of this gate when $V_1$ and $V_2$ are the input. Vector sorting with Optimization must be found to reduce $P$, thus the state of the gate in the circuit changes as little as possible.

3 Determining the Influence Coefficient

Take C17 circuit in ISCAS85 for example to illustrate the relationship between the input and the consumption.

Figure 1 represents the circuit structure of C17 with five inputs and six NAND gates, among them six gate status are related to input $c$, this means $c$ input changes will cause changes mostly in the six gate status, on the contrary, $a$ and $e$ are related to only two gates, the changing of $a$ and $e$ can lead to two gate status changes at most. From the above we know that the circuit structure is related to power relations closer.

We choose the No.1 NAND gate in C17 circuit to consider, if focused on the determined test set, the value of $c$ is 0 in the two test vectors, then changing of $a$ does not affect the state of $g$, on the contrary the value of $c$ is 1, then a changes will affect the state of the $G$. It is showed that the changing of gate status is relevant to the value of input in test set. In summary, the change of circuit status and the structure of circuit and characteristics of the test sets are closely related, these relationships can be summed up the relationship between circuit activity and the input, using the influence coefficient to express size when the variation of the each input effect on activity,
sorting test vector according to the coefficient and then began the quantitative analysis.

Consider a measured circuit with M original inputs and n gates inside, gate G presents progression in the circuit, T is the fault test set, the Vi is the test vectors in the test set. Definition: the influence coefficient which the number i original input impact on the status change of gate j is  

\[ W_{ij} = W_{x} \times P(y = key) + W_{y} \times P(x = key) \]  

In the equation, \( x \) and \( y \) are the direct inputs of gate j, \( W_{x} \) is the influence coefficient when original input i impact on the input x, \( P(y = key) \) is the probability when y take the key value during the test procedure. For AND gate, key value is 1 (because when the input of AND of gate is 1, another change in the input will lead to changes in the gate state); Similarly, the key value is 0 for OR gate; for NAND gate or the buffer gate, it only has one direct input, assuming that the input is x, then \( W_{ij} = W_{x} \). The multiple input gate is combined by the two input gate in the circuit and calculate the influence coefficient. The influence coefficient when original input impact on the gate which g=1 is equal to which another input of this gate take key value.

Definition: the influence coefficient when the number i original input effect on the whole circuit state changes is:  

\[ W_{i} = \sum_{j} \left[ \text{fanout}(j) \times W_{ij} \right] \]  

The circuit as shown in Figure 2 as an example, test set T are \{ (10100), (0100), (01110), (01010), (00111), (00100) \}, the influence coefficient which b impact on circuit status change is composed of the influence coefficient by B on gate 2, 6, 7, 9, 10 status: The influence coefficient \( W_{b} \) of b on 2: 1; The influence coefficient \( W_{b} \) of b on 6 is \( W_{6} \times p(h = 1) \); The influence coefficient \( W_{b} \) of b on 7 is \( W_{7} \times p(j = 0) \); The influence coefficient \( W_{b} \) of b on 9 is \( W_{9} \times p(i = 0) \); The influence coefficient \( W_{b} \) of b on 10 is \( W_{10} \times p(k = 0) \); Among that \( p(h = 1) \), \( p(j = 0) \), \( p(i = 1) \), \( p(k = 0) \) is the probability when h, j, i, k take their own gate value respectively, the influence coefficient \( W_{b} \) of b impact on circuit status changes is:  

\[ 2 \times W_{6} + 1 \times W_{6} + 2 \times W_{5} + 1 \times W_{5} + 1 \times W_{5} = 5 \]  

then calculate the influence coefficient which each original input changes influence on circuit state respectively.
4 Optimizing the Test Set Order

Last chapter we assign influence coefficient to each of the original input, this value includes influence degree of the original input change impact on the whole circuit state variation, we use this value to correct the Hamming distance and make it become the weighted Hamming distance, then use ordering method which is similar with Hamming distance to optimize the ordering of test set. The specific steps are as follows:

Take the circuit in Figure 2 and the test set T as examples, definite the test vectors in T: V1, V2, V3, V4, V5, V6, the individual bits in the test vector corresponding to each original input. The Hamming distance between two vectors is the number of original input of different values, for example the Hamming distance between V1 and V2 is 3; First the Weighted Hamming distance take bitwise exclusive-OR of two vectors, and then multiply it with bitwise by the influence degree of the original input, and then summed, for example the Weighted Hamming distance between V1 and V2 is Wa+Wb+Wc. The method of sorting test set is similar to the Hamming distance sorting method, both of them using the Greedy algorithm.

5 Conclusions

This method is applied to part of the ISCAS85 and ISCAS89 circuit, uses the full-scan test design and MINTEST test set, whose optimization procedure uses MATLAB software, optimized with the greedy algorithm. Each circuit activity evaluation standard uses the measured circuit internal weighted transitions. This method has an obvious advantage over the Hamming distance method on the optimization of activity. The average optimization of this method is 48.07% and maximum optimization is 69.22%, the average optimization of Hamming distance is 28.58%, on the other hand, time spent in this method is similar with which spent in the Hamming distance method.

References