International Journal of Engineering Inventions
e-ISSN: 2278-7461, p-ISSN: 2319-6491
Volume 2, Issue 8 (May 2013) PP: 41-44

Reducing Power Consumption during Test Application by Test Vector Ordering

Desai Naincy Nareshbhai
Department of Electronics Engineering, Gujarat Technological University, Ahmadabad, Gujarat, India

ABSTRACT: *In VLSI, during test, test power and test time have been the major issues. The test data compression is the well known method used to reduce the test time. In this paper, I describe the algorithm for don't care assignment and hamming distance based technique. The proposed approach is based on a re-ordering of the vectors in the test sequence to minimize the switching activity of the circuit during test application.*

KEYWORDS: *Data Compression, IP Core based SoC, Don't Care Bit Filling, MT-fill technique, Hamming distance based technique, Switching activity, Peak power, Average power.*

I. INTRODUCTION

The testing cost and testing power are the two well known issues of current generation IC testing. The test cost is directly related to test data volume and hence test data transfer time [1]. So if test data compression is applied, the problem of test cost can be solved. The extensive use of IP cores in SoC has further exacerbated the testing problem. Because of the hidden structure of IP cores, the SoCs containing large IP cores can use only those test data compression techniques and power reduction technique which do not require any modification or insertion in architecture of IP core. These methods should not also demand the use of ATPG, scan insertion or any such testing tools. They should be capable to use test data coming with IP core for data compression and power reduction. This test data may be partially specified or fully specified [2]. Thus the current research on IC testing cannot be directly applied to the SoC because of the hidden structure of IP core.So it can be inferred that the test data compression and test power reduction in context of hidden structure of IP core is the current need for SoC testing.

There are many test data compression techniques like linear decompression based, broadcast scan based and code based techniques. Considering to suitability to IP core based SoC, code based test data compression scheme is more appropriate. The don't care bit filling methods and test vector reordering further enhance the test data compression [3, 4]. The dynamic dissipation is the dominant term of power dissipation [5]. The dynamic power dissipation can be minimized by test vector set generated to minimize the frequency of switching at circuit lines during test application [6]. Test power reduction techniques involves: modification in LFSR architectures, partitioning the circuit, separate testing strategy for memory, low power ATPG algorithms, input control, test vector reordering and don't care bit filling methods. Among all these methods, the test vector reordering and don't care bit filling methods are suitable for IP core based SoC. So it can be said that don't care bit filling and test vector reordering are capable to reduce the test power as well as improve the test data compression. The reason is that it reduces dynamic power dissipation during testing through switching activity minimization in the circuit. In addition to that, they affect the correlation among data used and hence if used effectively can be helpful for further increase in test data compression.

In this paper, a don't-care assignment algorithms are analyse for compression and power modelling point of view according to experimental results, the compression ratio is high in Hamming distance based method and peak and average powers are low in MT-fill technique. The proposed technique is based on re-ordering test vectors in a given test sequence such that average and instantaneous power dissipations are minimized during testing of the circuit.

The test vector ordering problem has already been addressed in [7, 8], and a heuristic solution has been proposed. A complete directed graph in which each vertex represents a test vector and each edge represents the number of transitions activated in the circuit after application of the vector pair is first constructed. Next, a simple algorithm is used to find a Hamiltonian path of minimum cost in the graph. Unfortunately, the main problem in this approach rests in the time needed to construct the transition graph. With *n* being the number of test vectors, the construction of the graph requires *n.(n-I)* logic & timing simulations of the circuit to compute the number of transitions on each edge [8]. In the case of circuits for which a large number of test vectors are needed to ensure a high fault coverage, this approach may be not applicable (for example, 999 000 simulations of the same circuit are needed for a test sequence composed of 1000 vectors).

II. POWER DISSIPATION MODEL

The power dissipation in **CMOS** circuits is dominated by the dynamic power dissipation that occurs at a node when it switches from 0 to 1 or 1 to 0. Thus the average power is given by equation (1) that follows:

$P_{avg} = \sum_{all\ nodes\ j} \frac{1}{2} . C_{load}(j).V_{DD}^2 . a(j)$

where $C_{load}(j)$ is the load capacitance on node *j* in the CUT, V_{DD} the power supply voltage, and *a(j)* the frequency of switching or transition activity at node *j* (the average number of transitions per clock cycle). Power dissipated at a node is therefore proportional to the number of transitions at that node, which depends on the gate delays and sequence of input vectors applied to the circuit. Now if one considers the power dissipated in the CUT during application of a complete set of test vectors *Vi*, the total power can be expressed as follows (equation (2)):

$P_{test} = \sum_{all\ vectors\ vi} \sum_{all\ nodes\ j} \frac{1}{2} . C_{load}(j,vi).V_{DD}^2 . a(j,vi)$

Where *Cload(j,vi)* is the load capacitance on node *j* for vector *vi* and *a(j,vi)* the frequency of switching at node *j* when test vector *vi* is applied at the circuit inputs.

III. ALGORITHMS TO FILL DON'T CARE BITS

ATPG generated test data contains a large amount of don't care bits. Such don't care bits in test data can be manipulated to enhance the test data compression. For the statistical codes, test data is divided into equal size blocks of B bits. To improve the test data compression, the no. of distinct blocks in a given test set should be reduced and frequency of occurrence for each distinct block should be increased. In this, algorithms to fill don't care bits which have less computational complexity compared to other proposed algorithms.

3.1 MT (MINIMUM TRANSITION)-FILL BASED TECHNIQUE

In this section, descibe background of MT-fill. Consider a test vector matrix that has 0, 1 and X entries, where each row of the matrix corresponds to a test vector for the circuit. X is an unspecified value and can be filled with either 0 or 1. The conventional approach for filling the X's in the test cube is to do random fills (R-fill) in which the X's are randomly replaced by 0's or 1's. In Rfill, the idea is that it increases the chance that a single test cube would detect additional faults and hopefully the other test cubes would not be required and can be eliminated during reverse fault-simulation. However, since we are considering power, which involves the number of weighted transitions in the test vector, it is best to consider Minimum Transition Fill (MTfill). In MT-fill, a series of X entries in the test vector are filled with the same value as the first non-X entry on the right side of this series. This minimizes the number of transitions in the test vector when it is scanned in. For example, consider the test vector: 100**XX**010**X**1**X**0. This vector, after MT-fill, would become 100**00**010**11**0. If the test vector has a string of X bits that is not terminated by a non-X bit on the right side, then it should be filled by the bit value to the left of the sequence.

For example: 1000001011**XX** should be 100000101**11** after MT-fill.
Let□ s understand this concept with one example

X	0	1	1	1	X	X	X	X	0	1	0	0	1	X	X
1	1	0	1	1	1	X	X	0	0	0	0	1	0	X	X
1	1	0	1	X	1	X	X	1	1	0	X	1	1	X	X
0	1	0	1	1	0	X	X	X	0	1	X	1	0	X	X

Fig 1 test set for algorithm demonstration

Table-1 Applying MT-fill algorithm

Test vectors (Size=8)	**Applying MT-fill**
T1=X0111XXX	00111111
T2=X01001XX	00100111
T3=110111XX	11011111
T4=000010XX	00001000
T5=1101X1XX	11011111
T6=110X11XX	11011111
T7=010110XX	01011000
T8=X01X10XX	00111000

From given test set in figure 1 apply MT-fill algorithm to understand concept for vector size is 8 then in Table 1 calculate MT filling and WTM (weighed transitions metric) is used to estimate the average
and peak power consumption. Test data T= {T1, T2,.....,Tm} has m patterns, and the length of the pattern is n bits .Each test pattern Ti={ti1,ti2,...tin},1≤ i≤m, 1≤ j≤n denotes the jth bit in ith pattern. Weighed transitions metric WTMj estimated as per the formula

$$WTM_j = \sum_{i=1}^{n-1} (n - i) \times (tj, i \oplus tj, i + 1)$$

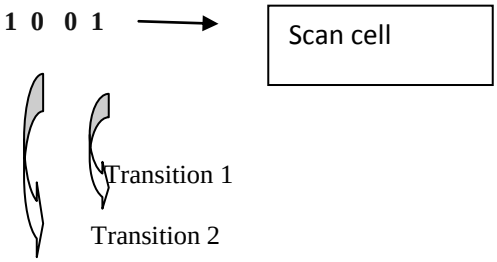

Fig 2 Transition in scan vector

Consider the example of the scan-in vector 1001. As shown in Fig.2, there are two transitions in the scan vector. While Transition 1 dissipates power at every cell in the scan chain while being scanned in; Transition 2 only dissipates power at the first scan cell. Thus when a test vector is being scanned in, the number of scan cell transitions caused by a particular transition in that vector would depend on the position of the transition in the scan vector. A transition is the difference between the size of the scan chain and the position in the vector in which the transition occurs. The number of weighted transitions is find from given above equation

3.2 HAMMING DISTANCE BASED TECHNIQUE

In this approach, the total bits in test data set are divided in to blocks of size B bits. These blocks may be completely specified or partially specified i.e. with don't care bits. For coding process, for each distinct block, the corresponding frequency of occurrence is calculated. The Hamming distance of block B1 with highest frequency of occurrence will be calculated from the B2 with the second highest frequency.

The Hamming distance is 1 if the bits on the same position of two blocks are opposite, i.e. ATPG generated test data contains a large amount of don't care bits. Such don't care bits in test data can be manipulated to enhance the test data compression. For the statistical codes, test data is divided into equal size blocks of B bits.

To improve the test data compression, the no. of distinct blocks in a given test set should be reduced and frequency of occurrence for each distinct block should be increased. In this paper, algorithms to fill don't care bits which have less computational complexity compared to other proposed algorithms. In this approach, the total bits in test data set are divided in to blocks of size B bits. These blocks may be completely specified or partially specified i.e. with don't care bits. For coding process, for each distinct block, the corresponding frequency of occurrence is calculated. The Hamming distance of block B1 with highest frequency of occurrence will be calculated from the B2 with the second highest frequency. The Hamming distance is 1 if the bits on the same position of two blocks are opposite, i.e. '1' and '0'. The Hamming distance between two blocks is summation of such bits with opposite values.

The Hamming distance between 10X1 and 010X is 2 as its first and second bits have opposite values. If the Hamming distance between B1 and B2 is more than 0, the Hamming distance with next block with descending order of frequency will be calculated. Two blocks for which the Hamming distance is 0, will be merged and a new block M1 will come into existence. The next block in the sequence will be than compared with merged block M1. This process is repeated until further merging is not possible. The process is repeated with the next highest frequently occurring still unmerged block. The merging has increased the number of specified bits. Still there is a chance that few bits are unspecified. Such bits are replaced with zeroes.

Let s understands this concept with one example from given test set in Figure 1. Consider the test data set with total 62 bits shown in Figure 1 Here the block size b=4. To make the last block of size b, at the end of test set two don't care bits are appended. Here the unique vectors are {10XX, 11XX, 1101, 01XX, X011, 1XXX, 0000, X010, X1XX, 110X, 0101, X01X} with the corresponding frequencies {3, 2, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1}. Starting with B1: 10XX. The Hamming distance of B1 from B2, B3, B4 is 1, 1, 2 respectively but with B5, it is 0. So B1 will be merged with B5. 10XX and X011 will make a merged block 1011 and frequency of this merged block is sum of the individual block i.e. 4. This merged block M1 will be further compared with B6 to B12. B6 and B12 will be merged with M1. After one cycle of merging the merged block 1101 has frequency

6. The next cycle of merging will start with B2 as it is still unmerged. The same process will continue with all unmerged blocks. For given example, the merged symbols are {1011, 1101, 0101, 0000, X010} with corresponding frequencies {6, 6, 2, 1, 1}. The last merged symbol X010 still contains a don't care bit which will be replaced by 0 and the merged symbol will be 0010.

Table-2 Applying Hamming distance based algorithm

Test vectors (Size=8)	**Applying Hamming distance based**
T1=X0111XXX	10111011
T2=X01001XX	00100101
T3=110111XX	11011101
T4=000010XX	00001011
T5=1101X1XX	11010111
T6=110X11XX	11011101
T7=010110XX	01011011
T8=X01X10XX	10111011

IV. CONCLUSION

Test power reduction and test data compression has become the essentials for today's IP core based SoC. But these two aspects are trade-off of each other. The partially specified test sets are processed with bit filling mechanism. This bit filling process affects power as well as compression. In this paper, it is observed that from above to bit filling algorithm Hamming distance based technique gives higher Compression ratio. So dynamic power can be reduced.

ACKNOWLEDGEMENT

This work was greatly supported by Gujarat Technological University and Seer Akademi by providing their guidance and encouragement.

REFERANCES

[1] N. A. Tauba (2006) Survey of Test Vector Compression Techniques: proceeding IEEE transcaction Design & Test of Computers-2006.

[2] Mehta U, Dasgupta K, Devashrayee N (2009) Survey of Test Data Compression Techniques Emphasizing Code Based Scheme : proceeding IEEE 12th Euromicro Conference on Digital System Design (DSD09).

[3] Mehta U, Dasgupta K, Devashrayee N (2009) Frequency dependant bit appending: an enhancement to statistical codes for test data compression. Proceedings of the India Conference, NDICON'09, December 2009, pp 301–304.

[4] P. Girard (2002) Survey of Low –Power Testing of VLSI Circuits: proceeding IEEE Design & Test -2002 pp.82-92

[5] N. Nicola and B. M. Al-Hashimi (2003) Power–Costrained Testing of VLSI Circuits: proceeding in Kluwer Academic Publishers-2003.

[6] P. Girard, C. Landrault, S. Pravossoudovitch and D.Severac(1998)Reducing Power Consumption During Test Application by Test Vector Ordering: proceeding in ISCAS- 1998 pp.296-299.

[7] S. Chakravarty and VP. Dabholkar, ***"Minimizing Power Dissipation in Scan Circuits During Test Application",*** Proc. of Int. Workshop on Low-Power Design, pp. 51-56, April 1994.

[8] S. Chakravarty and VP. Dabholkar, **"**Zbo *Techniques for Minimizing Power Dissipation in Scan Circuits During Test Application",* Proc. of Asian Test Symp., pp. 324-329, November 1994.