

Analysis of Don't Care Bit Filling Techniques for Optimization of Compression and Scan Power

K. A. Bhavsar
Electronics & Communication Department
VPMP polytechnic
Gandhinagar India

U. S.Mehta
PG VLSI Design,EC branch
Nirma University
Ahmedabad

ABSTRACT

Test power and test time have been the major issues for current scenario of VLSI testing. The test data compression is the well known method used to reduce the test time. The don't care bit filling method can be used for effective test data compression as well as reduction in scan power. In this paper we describe the algorithm for don't care assignment like MT(Minimum Transition)-fill technique and hamming distance based technique. The selective Huffman, optimal Huffman and modified selective Huffman coding are applied on the mapping set to give the optimum Compression and weighted transition matrix is used for scan power. Using these techniques find compression and scan power parameters like average power and peak power and conclude that MT- fill technique gives low peak and average powers and Hamming distance based modified selective Huffman coding technique gives higher compression ratio compare to another methods like selective and optimal Huffman coding.

General Terms

Modified Selective Huffman Code, Selective Huffman Code, Optimal Selective Huffman Code, Weighted Transition Matrix

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 .

1. 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 ready 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.

In literature, 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 paper is organized as follows. Section 2 presents estimation of compression and scan power . Section 3 introduces the algorithm of don't care bit filling method. Section 4 describes Experimental results of compression ratio and peak and average powers. Finally conclusions and future work discussion are presented in Section 5.

2. BACKGROUND

In this section, we discuss some of the relevant background concepts used in the paper.

2.1 Estimation of Power

In CMOS circuits, the predominant fraction of power is dissipated when circuit elements switch from logic 0 to 1 or vice versa. For a circuit-under-test (CUT), controlled entirely by the test vectors applied to it, the elements will switch value when the primary inputs change value or the scan cells change values. In this paper we assume that if the primary inputs of the CUT are directly

controllable from the chip pins, then they are held constant during scan-in. Thus in this case, during scan-in, all switching activity is due to the transitions in the scan chain.

Let us consider a CUT with 4 scan cells and a vector 1001 being scanned in (see in figure 1). Let the scan cells initially be 0000. At the first clock when the first input is scanned in, the state of the scan cells will be 1000. Thus the state of the first cell has changed from 0 to 1. This will cause other gates in the CUT to switch depending on the circuit. At the second clock when the next input is being scanned in, the cells will be in a state 0100. Here both the first and the second cells have changed states. This continues until the complete test-vector has been scanned in. The test vector is then applied to the CUT and the output response is captured back in the scan chain. As the next scan vector is being scanned in, the transitions in the output response from the previous scan vector being scanned out will also cause switching activity. Thus we can divide the power dissipation during scan testing into:

- scan-in power - due to transitions in scan test vectors
- scan-out power - due to transitions in the output response being scanned out.

The best way to estimate power during scan testing would be to do actual circuit simulation to actually find the number of circuit elements that switch when a vector is scanned in. However this procedure takes a very long execution time and is thus very expensive. Instead, in this paper to estimate the scan-in or the scan out power, we use the weighted transitions metric proposed by [8]. In this paper, a widely used weighed transitions metric (WTM) introduced in [9] is used to estimate the average and peak power consumption. Test data $T = \{T_1, T_2, \dots, T_m\}$ has m patterns, and the length of the pattern is n bits. Each test pattern $T_i = \{t_{i1}, t_{i2}, \dots, t_{in}\}$, $1 \leq i \leq m$, $1 \leq j \leq n$ denotes the j^{th} bit in i^{th} pattern. weighed transitions metric WTM_j for T_j the average test power P_{avg} and peak power P_{peak} are estimated as per the formula [3]

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

$$P_{peak} = \frac{\sum_{j=1}^m WTM_j}{m}$$

$$P_{avg} = \max_{1 \leq j \leq m} WTM_j$$

Consider the example of the scan-in vector 1001. As shown in Fig.1, 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. According to [8], the weight assigned to 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 found from given above equation:

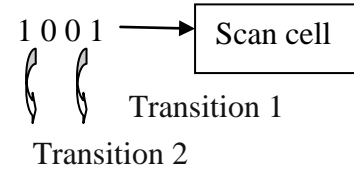


Fig. 1: Transition in Scan Vector

2.2 Estimation of Compression ratio

Using patterns of various types and various lengths, the compression algorithms exploit different features of the test set. Mapping and reordering the initial test set emphasizes these features. Therefore, the compression ratio is influenced firstly by the mapping and reordering algorithm, and then by the type of input patterns and the length of the pattern, and finally by the compression algorithm.

$$Compression\ ratio(\%) = \left(1 - \frac{Compressed\ Bits}{Original\ Bits}\right) \times 100$$

In [10] describes statistical code based methods like Huffman selective Huffman, optimal Huffman, VIHC (Variable input Huffman coding), and SVIHC (Split-VIHC) and evaluate and compare compression ratios of all methods. As per experimental result prove that SVIHC gives better compression compare to another methods. In [7] modified selective Huffman coding gives better compression ratio compare to another methods. In this paper for finding compression ratio we use selective Huffman, optimal selective Huffman and modified selective Huffman coding then compare compression ratio.

Let's understand compression methods concept with one example

Table- 1. Test Data Partitioning and Occurrence Frequencies

Test set T	Distinct Blocks	Occur. Freq
1010000010101111	1010	9/20
1111000010100001	0000	5/20
1010000000101010	1111	3/20
0000101010100000	0001	2/20
1010111110100001	0010	1/20

Table 2 Example for selective, optimal selective and Modified selective Huffman code

Distinct Blocks	Occur. Freq	Selective Huffman N=3	Optimal selective Huffman N=3	Modified selective Huffman N=3	
				Data	E/N
S0 -1010	9/20	10	0	0	1
S1-0000	5/20	110	10	10	1
S2-1111	3/20	111	110	11	1
0001	2/20	00001	1110001	0001	0
0010	1/20	00010	1110010	0010	0
Original 80 bits		57 bits	49 bits	37 bits	

From Table 2 observe that modified selective Huffman code gives higher compression ratio compare to selective and optimal Huffman coding methods.

3. 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 , an algorithms to fill don't care bits which has less computational complexity compared to other proposed algorithms .

3.1 MT(Minimum Transition)-Fill Based Technique

In this section, describe 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 fill (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 (MT-fill).

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: 100XX010X1X0. This vector, after MT-fill, would become 100000101100. 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: 1000001011XX should be 100000101111 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. 2: Test Set for Algorithm Demonstration

Table-3 Applying MT-fill algorithm and WTM

Test vectors (size=8)	Applying MT-fill	WTM
T1=X0111XXX	00111111	6
T2=X01001XX	00100111	14
T3=110111XX	11011111	11
T4=000010XX	00001000	7
T5=1101X1XX	11011111	11
T6=110X11XX	11011111	11
T7=010110XX	01011000	21
T8=X01X10XX	00111000	9

From given test set in figure 2 apply MT-fill algorithm to understand concept for vector size is 8 then in Table 3 calculate peak power and average power from equation . The peak power is 21 and average power is 11.25.

For calculating compression ratio apply different Huffman coding methods

For Block size =4 and 3-encoded distinct block using selective Huffman ,optimal Huffman and modified selective Huffman technique gives compression ratios are 12.5%, 12.5% and 34.37%.

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, an algorithm to fill don't care bits which has 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 understand this concept with one example from given test set in Figure 2

Consider the test data set with total 62 bits shown in Figure 2 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}. 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.

For calculating compression ratio we use different Huffman coding methods

Then compression ratio using selective Huffman ,optimal Huffman and modified selective Huffman code for 3 encoded distinct blocks are 28.1%,40.6% and 53.12%.

For calculate peak power and average power

Table-4 Applying Hamming distance based algorithm and WTM

Test vectors (size=8)	Applying Hamming distance based	WTM
T1=X0111XXX	10111011	18
T2=X01001XX	00100101	17
T3=110111XX	11011101	14
T4=000010XX	00001011	9
T5=1101X1XX	11010111	18
T6=110X11XX	11011101	14
T7=010110XX	01011011	23
T8=X01X10XX	10111011	18

From Table 4 calculate peak power and average power using given equation ,peak power is 23 and average power is 16.37.

4. EXPERIMENTAL RESULTS

In this paper describe don't care bit filling algorithm and calculate Weighted Transition Matrix, peak power ,average power and compression ratio using different Huffman coding methods , implementation MT-fill technique and Hamming distance based technique algorithm and Selective Huffman ,Optimal Selective Huffman and Modified Selective -Huffman code were MATLAB7.0 language. The experiments are conducted on a workstation with a 3.0 GHz Pentium IV processor and 1GB of memory. For the validation purpose, the widely cited ISCAS89 full scan benchmark circuits are used.

The test sets (with don't care) obtained from the MINTEST ATPG program are used here for experimental work. The test set is first processed with don't bit care filling algorithm, then Modified Selective Huffman compression scheme is applied to set.

Calculation of the total number of transitions for a complete set using Weighed Transition Matrix .Peak power ,Average power using above given equation.

Table 5 Comparisons of two don't care bit filling methods for scan power

Circuit	MT-fill		Hamming distance based	
	Peak Power	Avg. Power	Peak Power	Avg. Power
s5378	11524	3460	13741	11536
s9234	14107	4000	16692	8132
s13207	94887	7870	121714	17809
s15850	70909	13600	87685	24850
s38417	437998	117800	693278	578450
s38584	481182	83900	527092	108050

From Table 5 observed that MT- fill technique gives lower peak and average powers compare to Hamming distance based method.

Table 6 Comparisons of two don't care bit filling methods for Compression ratio(%)

Circuit	Selective Huffman(m=8)		Optimal Selective Huffman(m=8)		Modified Selective Huffman(m=8)	
	MT-fill	Hamm. distance based	MT-fill	Hamm. distance based	MT-fill	Hamm. distance based
s5378	28.4	29.2	45.1	46.3	53.4	54.2
s9234	22.4	30.5	38.4	51.6	47.4	55.5
s13207	42.5	45.2	65.6	69.3	67.6	70.3
s15850	31.5	39.2	50.4	62.2	56.5	64.2
s38417	29.5	34.9	47.6	55.9	53.8	59.9
s38584	32.2	36.8	51.6	58.5	57.1	61.9

From Table 6 observed that modified selective Huffman coding using Hamming distanced based bit filling method gives higher compression ratio compare to selective Huffman ,optimal Huffman code based methods

6. 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 modified selective Huffman coding method gives higher compression ratio compare to another Huffman coding like selective Huffman and optimal selective Huffman and MT-fill based gives lower peak and average powers .

7. REFERENCES

[1] N.A.Tauba (2006) Survey of Test Vector Compression Techniques :proceeding IEEE transaction 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] Mehta U, Dasgupta K, Devashrayee N (2010) Modified Selective Huffman Coding for Optimization of Test Data Compression ,Test Application Time and Area Overhead :Proceeding in Journal of Electronic Testing Theory and Applications-2010,vol.26

[8] R.Sankaralingam, R. Oruganti and N. Touba(2000) Static Compaction Techniques to Control Scan vector Power Dissipation :Proceeding in IEEE VLSI Test Symposium-2000,pp. 35-42

[9] Mehta U, Dasgupta K, Devashrayee N (2010) Run Length Based Test Data Compression Techniques:How Far From Entropy and Power Bounds?:proceeding in VLSI Design from Hindawi Publication Corporation-2010

[10] K.A.Bhavsar Mehta U(2011), Analysis of Test Data Compression Techniques Emphasizing Statistical Coding Schemes:proceeding in ACM Digital Library USA

AUTHORS PROFILE

1. Kinjal A.Bhavsar received her B.E.degree in Electronics & Communications Engineering from North Gujarat University .She is Pursuing her M.Tech. in VLSI Design from Nirma University as a sponsored candidate from VPMP Polytechnic ,EC department ,Gandhinagar

2. Usha Sandeep Mehta received her B.E. degree in Electronics & Communications Engineering from Gujarat University and Post Graduate in VLSI Design from Nirma University. She is pursuing her Ph.D. degree in VLSI Design- EC Engineering from Nirma University, Ahmedabad. She is a member of IEEE, CSI, IETE and AICTE. Currently she is Sr. Associate Professor at PG-VLSI group, Department of Electronics and Communication Engineering at the Institute of Technology, Nirma University, Ahmedabad.