TEST VECTORS REDUCTOIN FOR INTEGRATED CIRCUIT
TESTING USING HORIZONTAL HAMMING DISTANCE

ARBAB ALAMGIR

A project report submitted in partial fulfilment of the
requirements for the award of the degree of
Master of Engineering (Computer and Microelectronic Systems)

Faculty of Electrical Engineering
Universiti Teknologi Malaysia

JUNE 2016

To my loving parents and grandfather.

Thank you, your support and prayers has always been my strength.

ACKNOWLEDGEMENT

In the name of ALLAH, the most Beneficial and the most Merciful. Sholawat to Prophet Mohammed (PBUH). Thanks to my ALLAH for giving me unbelievable strength to accomplish my thesis and research.

Professor Dr. Abu Khari Bin A'ain has been a motivational and most helping factor in completing my research. I would thank him for being my thesis adviser, for his patience and encouragement. His ideas and suggestions improved my skills and helped me whenever I was in a difficult situation. His valuable feedback and innovative ideas contributed greatly to this dissertation.

I would specially thank my parents for their prayers and their support throughout the period of my studies.

I would also thank to my colleagues at "Computer and Microelectronic Systems" department of "Universiti Teknologi Malaysia" who had been a helping factor for me. I have great regard for them from depth of my heart.

I am indeed grateful to Almighty Allah for providing me the opportunity and strength to commence and conclude my Master of Engineering at "Universiti Teknologi Malaysia".

ABSTRACT

In testing digital combinational logic for stuck-at faults, it is required to determine the most appropriate test sequence needed to detect the required number of possible faults. The exhaustive test pattern generation method is the simplest approach to implement as it produces test patterns consisting of all possible input combinations of the circuit under test. However, a consequence of this approach is that it results in a large test set when the number of circuit inputs is large. This can take an unnecessarily long time to apply on the circuit under test as during the test process, only a small fraction of all possible test vectors is actually required to produce high percentage of fault coverage. As an alternative, random test pattern generation applies a random set of test patterns which can be used to reduce the number of test patterns compared to exhaustive test. However, both test pattern generation approaches generate unnecessary test vectors to apply to the circuit as multiple patterns typically detect the same fault. Antirandom testing on the other hand ensures that the identified test vectors to use do not detect the same fault by introducing the concept of Hamming distance between test vectors and this distance is be maximized. This results in a reduction in the number of required test vectors when compared to an exhaustive test. However, the algorithm for Antirandom test vector generation is computation intensive and vague in its definition when there are more than one possible next test vectors. In this study, efficient calculation of Hamming distance has been proposed, moreover the choice of the next test vector is addressed by using the proposed Horizontal Hamming distance method which has not yet been explored. The approach effectively detects faults at a much faster rate and produces a much higher fault coverage than the existing Antirandom method.

ABSTRAK

Untuk menguji logik gabungan digital yang ada kegagalan, adalah perlu memastikan set corak ujian yang paling sesuai. Kaedah ujian corak yang menyeluruh adalah pendekatan yang paling mudah untuk dilaksanakan kerana ia menghasilkan corak ujian untuk semua kemungkinan gabungan input litar yang diuji. Namun, hasil daripada pendekatan ini menyebabkan set ujian yang besar apabila bilangan input litar bertambah. Ini mengambil masa yang terlalu lama untuk di aplikasikan kepada litar yang sedang diuji dan hanya sebahagian kecil daripada ujian vektor diperlukan untuk menghasilkan peratusan liputan kerosakkan yang tinggi. Dengan ujian yang menyeluruh, ujian vektor yang tidak diperlukan sebenarnya digunakan dan ini mengambil masa ujian yang lebih lama daripada yang sepatutnya. Sebagai alternatif, kaedah generasi corak ujian rawak digunakan untuk mengurangkan bilangan corak ujian berbanding dengan ujian lengkap. Ujian Antirandom memastikan bahawa ujian vektor telah dikenalpasti supaya ia tidak mengesan kerosakkan yang sama dengan memperkenalkan konsep jarak Hamming yang mana ujian vektor dan jarak akan dimaksimumkan. Ini mengurangkan bilangan ujian vektor yang diperlukan berbanding dengan ujian lengkap. Walau bagaimanapun, algoritma bagi ujian vektor kaedah generasi Antirandom adalah kabur dan tidak dikenal pasti apabila ada lebih daripada satu kemungkinan ujian vektor selanjutnya untuk dipilih. Dalam kajian ini, pengiraan yang cekap jarak Hamming telah dicadangkan. Ujian vektor yang seterusnya ditangani dengan menggunakan kaedah jarak mengufuk Hamming yang masih belum diterokai. Pendekatan yang efektif mengesan kegagalan pada kadar yang lebih cepat dan menghasilkan liputan kesalahan yang lebih tinggi daripada kaedah Antirandom yang sedia ada.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

AR	-	Antirandom
BIST	-	Built In Self-Test
CUT	-	Circuit Under Test
FC	-	Fault Coverage
HD	-	Hamming Distance
HTHD	-	Horizontal Total Hamming Distance
IC	-	Integrated Circuit
TCD	-	Total Cartesian Distance
TPG	-	Test Pattern Generation
VTHD	-	Vertical Total Hamming Distance
ATPG	-	Automatic Test Pattern Generation
ORA	-	Output Response Analyzer
CD	-	Cartesian Distance
MTHD	-	Maximum Total Hamming Distance
MTCD	-	Maximum Total Cartesian Distance
LFSR	-	Linear Feedback Shift Register

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1   Introduction

Race of innovation and technology development has shifted the trends from System on Board (SoB) to System-on-Chip (SoC) and System-in-Package (SiP). Embedding millions of logical operations on a single platform with efficient utilization of resources has resulted in extremely complex integrated circuits (ICs). Moreover, indulge design and manufacturing has become a challenge to produce required functionality at an affordable price. Stuck-at-faults, delay faults and manufacturing defects have made verification and testing a vital step in formulation of VLSI realization process, increasing production cost by 40%. In a short span of time it is impractical to synthesize and propagate faults on each node of an embedded circuit. Furthermore, minimizing yield loss and defect level can cause a delayed availability of devices to consumers. Whereas timely organized testing with in minimum duration may preserve time and cost of testing.

Rapid testing even throughout the production life cycle is not enough to maintain modern quality standards [1, 2]. Now-a-days quick digital circuit testing during operation is critical for reliable electronic services. Avoiding expensive and delicate probe testing in automatic test equipment, Built-in-Self-test (BIST) has solved problem in periodic testing of automotive electronics. BIST, comprising of Automatic Test Pattern Generator (ATPG) and Output Response Analyzer (ORA) is

mounted on device itself to tests an IC on regular intervals based on its requirement. ATPG uses space efficient Linear Feedback Shift Register (LFSR) to generate test patterns for application on Circuit Under Test (CUT). Moreover, signature analysis and comparison is carried out by ORA.

Relation of stuck-at-faults with delay, short and open faults suggests that exposing all stuck-at-faults in logical circuits gives 99.9% of confidence in an IC. Therefore, study on testing stuck-at-faults has been carried out minimizing the number of test patterns required for testing of an IC. The simplest approach to test stuck-at faults in digital combinational circuits is to use an exhaustive testing where the test set comprises of all the possible input combinations for CUT [1]. For a circuit with a large number of inputs, the test process would take a substantial amount of time to complete. Test time would be longer than necessary because many of the test vectors used are not actually needed; as some test vectors may detect more than one faults. Random testing is another alternative for test pattern generation which picks random test vectors from input space and tests the IC till required Fault Coverage (FC) is obtained. However, both test pattern generation approaches generate unnecessary test vectors to apply on circuit, as multiple patterns typically detect the same fault. Antirandom (AR) gives a selection criteria using Cartesian Distance (CD) and Hamming Distance (HD) ensuring that test vectors doesn't target same fault sites. Whereas AR is compute intensive and vague in its definition having no selection criteria when more than one test vectors are equally eligible for next selection [3]. Therefore, in the course of this study, definition of HD has been revised to facilitate selection criteria maximizing FC.

1.2   Horizontal and Vertical Hamming Distance

HD is count of different bits when two vectors are compared. For example, having a first vector v1 = {0000} and a second vector v2 = {1011} results in HD of 3 as the 1st, 3rd and 4th bits switch from 0 to 1 while changing input pattern from v1 to v2. This concept has been used to maximize the distance between two vectors. It can be observed that bits not only differ from one vector to another, but they also differ

from one bit to another within that same vector. Simulations show that a different bit count within a single vector is of big importance from FC point of view. In this study it is proposed that the HD can be in two directions, vertical and horizontal. The one defined above is termed as vertical HD and second one in the horizontal direction gives the count of bits switching within a test vector. For example, v1 = {0000} doesn't have any bit switching within this vector so its Horizontal Total Hamming Distance (HTHD) will be zero whereas in v2 = {1010}, there are three transitions. The first transition is from the 1st bit to the 2nd bit, the second transition is from the 2nd bit to the 3rd bit and third transition is from the 3rd bit to the 4th bit, resulting in HTHD of 3 for this vector. Figure 1.1 shows the calculation of Vertical Total Hamming distance (VTHD) (in the gray box) and HTHD (in the blue box) for a 4-bit sequence.
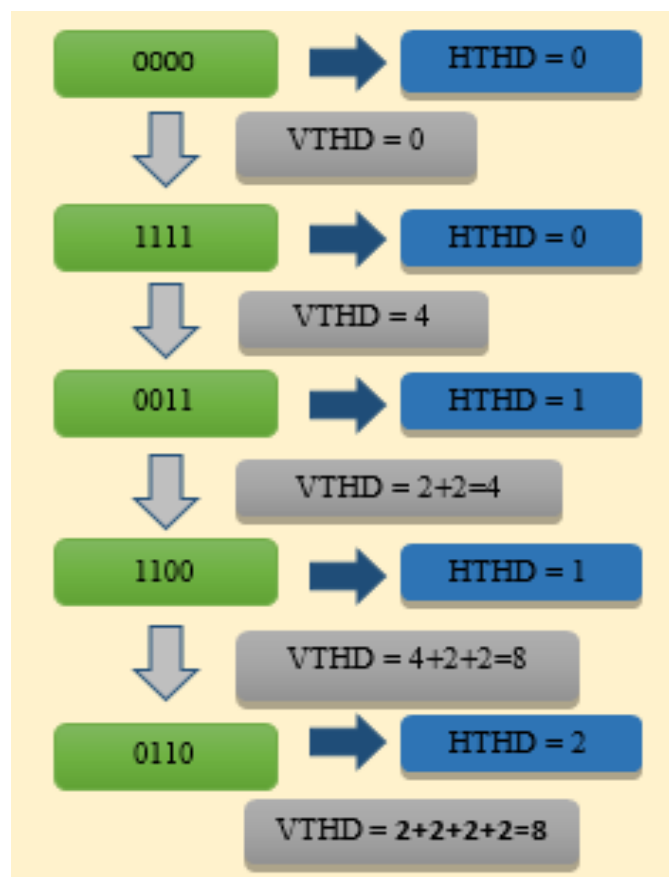

Figure 1.1: Difference of VTHD AND HTHD

1.3   Problem Statement

Removal of randomness in TPG is effective from a FC point of view. AR testing concept plays an important role in this perspective with application of two powerful filters. HD and CD are used to maximize the distance between preceding and subsequent test vectors. The randomness still prevails in cases when more than one test vectors have same Maximum Total Hamming Distance (MTHD) and Maximum Total Cartesian Distance (MTCD). For a 4 bit input CUT, starting with a seed value of {0000}, only one option is available for the next test vector (i.e. {1111}) having MTHD of 4 and MTCD of 2 among all other test vectors. For selection of the next test vector, Table 1.1 shows that there are six test vectors which have the same THD and TCD (shown in bold face). With AR testing approach, all of these six test vectors are equally eligible to be chosen as subsequent test vectors. The criteria to choose between these six vectors has not been given due consideration in AR testing method. Choosing randomly among these six test vectors introduces randomness in the selection procedure which if done deterministically, can lead to higher FC.

Table 1.1:   Test Vectors with THD and TCD

Candidate next test vector	THD with {0000,1111}	TCD with {0000,1111}
0001	4	2.7320
0010	4	2.7320
0011	**4**	**2.8284**
0100	4	2.7320
0101	**4**	**2.8284**
0110	**4**	**2.8284**
0111	4	2.7320
1000	4	2.7320
1001	**4**	**2.8284**
1010	**4**	**2.8284**
1011	4	2.7320
1100	**4**	**2.8284**
1101	4	2.7320
1110	4	2.7320

Distance maximization requires two types of distance calculations differing widely in calculation time and computational complexity. Prioritizing the test vectors based on distance approach can be compute intensive if selection procedure is not strategized properly. Computational complexities of HD and CD can be compared

considering their formulae. Although CD can be a function of HD but TCD cannot be formulized easily in form of THD.

$$CD = \sqrt{HD} \text{ where as } TCD \neq \sqrt{THD} \tag{1}$$

While selecting test vectors from input space, test vector with MTHD and MTCD is selected. With "M" number of selected test vectors and "N" number of available choices, M x N computations are required to calculate all VTHD's and test vectors with maximum VTHD's are shortlisted for CD calculations.

For a 4-bit input IC, prioritizing whole input space with respect to VTHD requires 680 computations and with a 10 bit input IC, number of computations required to prioritize whole input space are 178956800. Higher number of computations required to calculate VTHD makes AR a compute intensive algorithm for TPG. CD calculations are only applied on the test vectors that have maximum VHD. Therefore, efficient VTHD calculations can lead to quick TPG.

There has been no criteria to choose when more than one test vectors have same MTHD and MTCD. Moreover, FC has always been compromised for reduction of HD calculations which makes AR an inefficient algorithm. AR has also overlooked the HTHD while selecting test vectors from input space. This study will explore the following questions.

> Is there any relationship between Horizontal Hamming Distance and Fault Coverage?
> Does Horizontal Hamming Distance increase Fault Coverage?
> How to reduce computational complexity in calculating Vertical total hamming Distance without compromising on Fault Coverage?

1.4   Objectives

The objectives of this research are

➢ Analyze Horizontal Hamming distance of all test sets generated by ATLANTA for each ISCAS'85 benchmark circuits from Fault Coverage point of view.

➢ Employ Horizontal Hamming distance in Antirandom concept to increase Fault Coverage.

➢ Formulate an algorithm to reduce computational complexity of Vertical Total Hamming Distance calculations without compromising on Fault Coverage.

1.5   Scope of Research

Area of this study is bounded to detection of all type of stuck-at faults in combinational circuits. Standard ISCAS'85 combinational circuits has been used for the testing purposes [4]. They include bench circuits with a vast range of inputs from 5 to 207 inputs. The following Table 1.2 gives critical overview of all ISCAS'85 benchmark circuits.

Table 1.2:   List of ISCAS'85 Benchmark Circuits

ISCAS'85 Circuit	No. of input pins	No. of Output pins	No. of Gates	No. of faults
c17	5	2	6	22
c432	36	7	160	524
c499	41	32	202	758
c880	60	26	383	942
c1355	41	32	546	1574
c1908	33	25	880	1879
c2670	233	140	1193	2747
c3540	50	22	1669	3428
c5315	178	123	2307	5350
c6288	32	32	2416	7744
c7552	207	108	3512	7550

1.6   Thesis Organization

The rest of the thesis explains background of study, strategy to accomplish the objectives and results obtained with the help of this study. Chapter 2 introduces the topic of LFSR reseeding, white box and black box test pattern generations. This section gives an overview of VTHD calculations required for selection procedure of test vectors. Chapter 3 is for methodology showing a flow graph of the project along with a brief description on Atlanta and ISCAS'85 benchmark circuits. Chapter 4 contains the results obtained from all simulations and a critical analysis on the obtained results. This project ends with a conclusion and highlights the major contributions.   The thesis ends with a few suggestions for future work.

REFERENCES

[1]   L. Mariani, M. Pezzè, and D. Zuddas, "Chapter Four-Recent Advances in Automatic Black-Box Testing," *Advances in Computers,* vol. 99, pp. 157-193, 2015.

[2]   S. Anand, E. K. Burke, T. Y. Chen, J. Clark, M. B. Cohen, W. Grieskamp, *et al.*, "An orchestrated survey of methodologies for automated software test case generation," *Journal of Systems and Software,* vol. 86, pp. 1978-2001, 2013.

[3]   H. J. Hasan, M. Alshraideh, and B. A. Mahafzah, "Branch Coverage Testing Using Anti-Random Technique," *i-Manager's Journal on Software Engineering,* vol. 8, p. 7, 2013.

[4]   D. Bryan, "The ISCAS'85 benchmark circuits and netlist format," *North Carolina State University,* p. 25, 1985.

[5]   S. Xu and P. Xu, "A Quasi-best Random Testing," in *Test Symposium (ATS), 2010 19th IEEE Asian*, 2010, pp. 21-26.

[6]   E. J. McCluskey, "Built-in self-test techniques," *Design & Test of Computers, IEEE,* vol. 2, pp. 21-28, 1985.

[7]   D. P. Vallett, "IC failure analysis: The importance of test and diagnostics," *IEEE Design & Test of Computers,* pp. 76-82, 1997.

[8]   L. Balamurali, S. P. Sagar, K. Indumol, S. T. Engineer, and A. Kumar, "Test Optimization Using Adaptive Random Testing Techniques."

[9]   T. Y. Chen, F.-C. Kuo, R. G. Merkel, and T. Tse, "Adaptive random testing: The art of test case diversity," *Journal of Systems and Software,* vol. 83, pp. 60-66, 2010.

[10]   T. Chen, A. Bai, A. Hajjar, A. K. A. Andrews, and C. Anderson, "Fast anti-random (FAR) test generation to improve the quality of behavioral model verification," *Journal of Electronic Testing,* vol. 18, pp. 583-594, 2002.

[11]   W.-C. Lien, K.-J. Lee, T.-Y. Hsieh, and K. Chakrabarty, "A new LFSR reseeding scheme via internal response feedback," in *Test Symposium (ATS), 2013 22nd Asian*, 2013, pp. 97-102.

[12]   A. Coyette, B. Esen, R. Vanhooren, W. Dobbelaere, and G. Gielen, "Automatic generation of lightweight controllability and observability structures for analog

circuits," in *Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design (SMACD), 2015 International Conference on*, 2015, pp. 1-4.

[13]   M. Venkatasubramanian, V. D. Agrawal, and J. J. Janaher, "Quest for a quantum search algorithm for testing stuck-at faults in digital circuits," in *Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFTS), 2015 IEEE International Symposium on*, 2015, pp. 127-132.

[14]   S. Wu, Y. Wu, and S. Xu, "Acceleration of Random Testing for Software," in *Dependable Computing (PRDC), 2013 IEEE 19th Pacific Rim International Symposium on*, 2013, pp. 51-59.

[15]   A. Khalili, M. Narizzano, A. Tacchella, and E. Giunchiglia, "Automatic test-pattern generation for grey-box programs," in *Automation of Software Test (AST), 2015 IEEE/ACM 10th International Workshop on*, 2015, pp. 33-37.

[16]   Y. K. Malaiya, "Antirandom testing: getting the most out of black-box testing," in *Software Reliability Engineering, 1995. Proceedings., Sixth International Symposium on*, 1995, pp. 86-95.

[17]   M. E. Khan and F. Khan, "A comparative study of white box, black box and grey box testing techniques," *Int. J. Adv. Comput. Sci. Appl,* vol. 3, 2012.

[18]   H. Liu, X. Xie, J. Yang, Y. Lu, and T. Y. Chen, "Adaptive random testing through test profiles," *Software: Practice and Experience,* vol. 41, pp. 1131-1154, 2011.

[19]   M. S. Sahari, A. K. A'ain, and I. A. Grout, "Scalable Antirandom testing (SAT)," *SAT,* vol. 1355, p. c3540, 2015.

[20]   Z. Liu, X. Gao, and X. Long, "Adaptive random testing of mobile application," in *Computer Engineering and Technology (ICCET), 2010 2nd International Conference on*, 2010, pp. V2-297-V2-301.

[21]   Leeba Varghese, Suranya G.," Test Pattern Generation Using LFSR with Reseeding Scheme for BIST Designs" ,International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, Dec,2014

[22]   Wei-Cheng Lien, Kuen-Jong Lee, Tong-Yu Hsieh and Krishnendu Chakarabarty, "A new LFSR Rseeding Scheme Via Internal Response Feedback*, Asian Test Symposium* , 2013.

[23]   M Kalaiselvi, K.S Neelukumari, "LFSR-Reseeding Scheme for Achieveing Test Coverage", International Journal of computer Trends and Technology, 2013.

APPENDIX A

Atlanta – M 2.0 Guidelines

Commands for operations in Atlanta, c432 bench file is used as an example

Generating test patterns for a bench file *"atalanta-M -t c432.pat -W 1 c432.bench"*
Generating test patterns with the fault list. *"atalanta-M -t c432.pat -W 1 -F c432.flt c432.bench"*
Generating test patterns and correct test vectors to detect faults *"atalanta-M -t c432.pat -W 2 c432.bench"*
Generating test patters required to test CUT. *"atalanta-M -D 1 -t c432.pat -W 2 c432.bench"*
generating test vectors only for faults specified in bench file. *"atalanta-M -t c432.pat -f c432.flt -W 1 c432.bench"*
simulating test vectors in .pat file and writing the resulted fault coverage in .rep file *"atalanta-M -S -t c432.pat -P c432.rep c432.bench"*
Simulating test patterns in .pat file and writing all the undetectable faults in .ud file. *"atalanta-M -S -t c432.pat -P c432.rep -U c432.ud -v c432.bench"*