Low Power Test Methodology for SoCs: Solutions for Peak Power Minimization

A Thesis

Submitted For the Degree of

Master of Science (Engineering)

in the Computer Science and Engineering

by

Jaynarayan Thakurdas Tudu

Department of Computer Science & Automation

Division of Electrical Sciences

Indian Institute of Science

BANGALORE – 560 012

June 2010

TO

My Beloved Parents

Acknowledgements

I am very much grateful to Prof. Mathew Jacob and Dr. Virendra Singh for their guidance in various aspect of my research work. For all the official matter of my academic program Prof. Mathew Jacob has helped me whenever there was a need. The amount of time Dr. Singh has spent on me is more than what a student can expect from his supervisor. Dr. Singh has trained me in various aspect of the research. I was really in difficult situation in the beginning of my dissertation work. However, the affectionate help of Dr. Singh enabled me to overcome those difficulties gradually.

I would like to extend my gratitude to Prof. Erik Larsson of Linkoping University, Sweden for the guidance he has provided me in whatever possible way he could. Prof Erik has helped me in many ways by providing me a technical feed back on each of the paper which we have published, specially in choosing the proper sentences and grammatical correctness was due to him only.

I would like to thank Prof. Sunil Chandran of Computer Science and Automation department at Indian Institute of Science, Bangalore. Prof. Sunil has helped me in understanding the graph theoretic part of my research work. Infact, for my first work on vector reordering, Prof. Sunil only gave me a hints to tackle the problem.

I would like to thank to Prof. Vishwani D. Agrawal and Prof. Adit Singh of Auburn University, USA for their valuable suggestions in formalizing the problem. I also thank to Prof. Fujiwara of NAIST, Japan for his comments on scan chain reordering issue for power minimization. The suggestion of Prof. Fujiwara has really enhanced the quality of the paper. My experimental work was some what become lighter as Prof. Seiji Kajihara of Kyusu Institute of Technology, Japan kindly provided me the test patterns for some

of the ISCAS99 benchmark circuits.

I am thankful to my friends Pawan and Pramod for their supports in tools, providing me various informations about the research, and spending time in every joyous moment. I thank to Soheil Sammi of Linkoping University, Sweden for helping me in carrying out the experimental work.

My family members particularly my parents are the ones who has encouraged me to take up the research career, without their support it would not have been possible to come to the Indian Institute of Science. I respectfully offer my thanks to them.

Lastly, I humbly bow down my head in front of my most worshipable The supreme personality of Godhead, Lord Jagannatha.

Vita

Currently I am MSc(Engg) student at department of Computer Science and Automation, Indian Institute of Science, Bangalore. My MSc(Engg) dissertation work is in the area of Low Power Test Methodology of System on Chip (SoC). My thesis is guided by Prof. Matthew Jaboc and Dr. Virendra Singh. I explored the graph theoretic technique to formulate the low power scan testing methodology.

I did my BE in computer science and engineering from Biju Patnaik University of Technology, Rourkela, Orissa, India in the year of 2005. I worked as a lecturer at Silicon Institute of Technology, Bhubaneswar, Orissa for one year from 2006 to 2007. I joined MSc(Engg) program at IISc in the year of 2007.

Publications based on this Thesis

1. Jaynarayan Tudu, Erik Larsson, Virendra Singh, and Vishwani D. Agrawal, *On Minimization of Peak Power during SoC Test*, Proc. of European Test Symposium (ETS) 2009, Seville, Spain, May 25-29, 2009.

2. Jaynarayan Tudu, Erik Larsson, Virendra Singh, and Adit Singh, *Capture Power Reduction for Modular System-on-Chip Test*, Proc. of VLSI Design and Test Symposium (VDAT) 2009, Bangalore, India, July 8-10, 2009.

3. Jaynarayan Tudu, Erik Larsson, Virendra Singh, and Hideo Fujiwara, *Scan Cells Reordering to Minimize Peak Power during Scan Testing of SoC*, IEEE Workshop on RTL & High Level Testing (WRTLT) 2009, Hong Kong, China, November 27-28, 2009.

4. Jaynarayan Tudu, Erik Larsson, Virendra Singh, and Hideo Fujiwara, *Scan Cell Reordering to Minimize Peak Power during Test Cycle: A Graph Theoretic Approach*, Proc. European Test Symposium (ETS), 2010, *Poster session*, Paragues, Czech Republic, May 24-28, 2010.

Abstract

Power dissipated during scan testing is becoming increasingly important for today's very complex sequential circuits. It is shown that the power dissipated during test mode operation is in general higher than the power dissipated during functional mode operation, the test mode average power is around 3x and the peak power is around 30x of normal mode operation. The power dissipated during the scan operation is primarily due to the switching activity that arises in scan cells during the shift and capture operation. The switching in scan cells propagated to the combinational block of the circuit during scan operation, which in turn creates many transition in the circuit and hence it causes higher dynamic power dissipation. The excessive average power dissipated during scan operation causes circuit damage due to higher temperature and the excessive peak power causes yield loss due to IR-drop and cross talk. The higher peak power also causes the thermal related issue if it last for sufficiently large number of cycles. Hence, to avoid all these issues it is very important to reduce the peak power during scan testing. Further, in case of multi-module SoC testing the reduction in peak power fascilitates in reducing the test application time by scheduling many test sessions parallely. In this dissertation we have addressed all those stated issues. We have proposed three different techniques to deal with the peak power problem.

The first solution proposes an efficient graph theoretic methodology for test vector reordering to achieve minimum peak power supported by the given test vector set. Three graph theoretic problems are formulated and corresponding algorithms to solve the problems are proposed. The proposed methodology also minimizes average power for the given minimum peak power. Further, a lower bound on minimum achievable peak

power for a given test set is defined. The results on several benchmarks show that the proposed methodology is able to reduce peak power better than the industrial solution.

To address the peak power problem during test cycle (the cycle between launch and capture pulse) we have proposed a scan chain reordering technique. A new formulation for scan chain reordering as TSP (Traveling Sales Person) problem and a solution is proposed. The experimental results show that the proposed methodology is able to minimize considerable amount of peak power compared to the earlier proposals. To minimize the peak power dissipated during scan shift operation we have extended the same technique to consider the scan shift power. The proposed methodology consists of a graph theoretic problem formulation and corresponding algorithm. The experimental results show the effectiveness of the proposed methodology over the previously proposed and the industrial solutions.

The capture power (power dissipated during capture cycle) problem in testing multi chip module (MCM) is also addressed. We have proposed a methodology to schedule the test set to reduce capture power. The scheduling algorithm consist of reordering of test vector and insertion of idle cycle to prevent capture cycle coincidence of scheduled cores. The experimental results show the significant reduction in capture power without increase in test application time.

Keywords

VLSI Testing, SoC testing, Scan testing, Power aware test, Peak power, Capture power, Vector reordering, Scan reordering, and Graph algorithm

Contents

List of Tables

List of Figures

Chapter 1

Introduction

This dissertation work focuses on the reduction of peak power for circuit with scan architecture during scan operation. The power dissipated during the test mode operation is significantly higher than the power dissipated during normal mode of operation of circuit under test [55]. Recently it is reported by Shi Cheng and Rahit Kapur that the average power consumed during scan mode operation is around 3x the functional mode power and the peak power of scan mode operation is around 30x the functional mode peak power [8]. The excessive power consumption during scan testing is a majore concern particulary for modern high density and high frequency design [25].

Many solutions addressing power problem have been proposed over the last two decades. Most of the solutions proposed so far are aimed at reduction of average power [19, 31]. Although most of the proposals on average power achieved the reduction in peak power a special attention to high peak power is still needed. The reasons for exclusive consideration of peak power are; to assure the accurate at-speed test, reliability of circuit during scan shift operation, and test time reduction by scheduling more cores during test. A very few proposals have been proposed to address the peak power problem. Keeping these observation in mind, in this dissertation we are proposing a set of solutions to deal with the excessive peak power during scan operations.

The objective of this chapter is to provide a brief history and background concept of VLSI testing. The Chapter is divided into three different sections. The Section

1.1 discusses about the role of testing in VLSI design flow, and importance of scan testing and various scan architecture. Section 1.2 provides the detail concept of CMOS power dissipation, power dissipation during test, estimation of power and excessive power problem during scan testing. The Chapter is concluded in the Section 1.3 with description about the dissertation's chapters and contributions.

1.1 Testing of VLSI Circuit

Following the so-called Moore's law, the scale of ICs has doubled every 18 months. A simple example of this trend is the progression from LSI to VLSI devices. In the 1980s, the term "VLSI" was used for chips having more than 100,000 transistor and that has continued to be used over time to reffer to chip with millions of transistors. In 1986, the first megabits random access memory(RAM) contained more than 1 million transistors. Microprocessors produced in 1994 contained more than 3 million transistors. VLSI devices with many million of transistors are commonly used in the modern computers and electronic devices. This is a direct result of the steadily decreasing dimensions, referred to as **feature size**, of the transistors and interconnecting wires from tens of microns to tens of nanometers, with current submicron technologies based on a feature size of less than 100 nanometers. The reduction in feature size has also resulted in increased operating frequencies and clock speeds; for example, in 1971, the first microprocessors runs a clock frequency of 108 KHz, while current commercially available microprocessors commonly runs at several gigahertz.

The reduction in feature size increases the probability that a manufacturing defect in the IC will result in a faulty chip. A very small defect can easily result in a faulty transistor or interconnecting wire when the feature size is less than 100 nm. Furthermore, it takes only one faulty transistor or wire to make the entire chip fail to function properly or at the required operating frequency. Yet, defects created during the manufacturing process are unavoidable, and, as a result, some number of ICs is expected to be faulty; therefore, testing is required to guarantee fault-free products, regardless of whether the

product is VLSI device or an electronic system composed of many VLSI devices. It is also necessary to test components at various stages during the manufacturing process. There is general agreement with the **rule of ten**, which says that the cost of detecting a faulty IC increases by an order of magnitude as we move through each stage of manufacturing, from device level to board level to system level and finally to system operation in the field.

Electronics testing includes IC testing, PCB testing, and system testing at the various manufacturing stages and, in some cases, during system operation. Testing is used not only to find the fault free devices, PCBs, and system but also to improve production yield at the various stages of manufacturing by analyzing the cause of defects when faults are encountered. In some systems, periodic testing is performed to ensure fault-free system operation and to initiate repair procedures when faults are detected. Hence, VLSI testing is important to designers, product engineers, test engineers, managers, manufacturers, and end-users.

1.1.1 VLSI Design and Test Flow

Testing of a VLSI circuit typically consist of applying a set of test vector at the input of the circuit while the response at the circuit output are analyzed. Circuit that gives the wrong response is considered to be the faulty chip where as the circuit with correct response is considered as the fault free circuit. The Figure 1.1 illustrates the basic test procedure. The input test patterns are generated either by automatic test pattern generator (ATPG) or by random process. The response at output are compared with the predetermined response, if the response are correct for all the test then the circuit under test is declared as fault-free circuit otherwise the circuit is faulty.

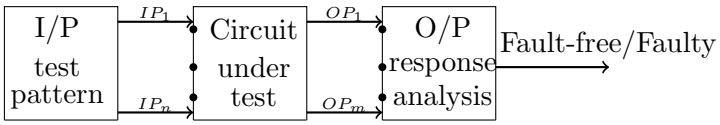


Figure 1.1: Basic test procedure

Testing plays an important role in VLSI design flow. The design has to be tested and verified after each design phase, hence the testing is performed at each phase of design flow. VLSI design flow consist of mainly following phases: 1. Design specification 2. Design 3. Fabrication 4. Packaging 5. Quality assurance. In Figure 1.2 the overall design flow is illustrate with corresponding test requirement. The figure show that the testing in some form is needed in each phase of design flow. The first phase of design is to specify the design requirement. Once the specification is documented the next phase is to design the circuit and synthesize it to produce the netlist for further processing. At this stage of design the functionality of the circuit has to be verified (modern formal verification technique are used in general) according to the specification.

Once the design is verified as correct the next phase is to fabricate in the wafer. Also at this time the test engineer develop the fault model and tests. Once the fabrication is completed the ICs are tested to determine the defective devices. The chip that passes the wafer test is then extracted and packaged. To eliminate some of the chip with defective package a testing is performed again. Finally the chip which passes both wafer test and package test are tested for quality assurance. This final testing includes the important parameter like input/output timing, voltage, and current. In addition to above described testing, chip is also tested in high temperature which is known as burn-in test which accelerated the test by stressing the device.

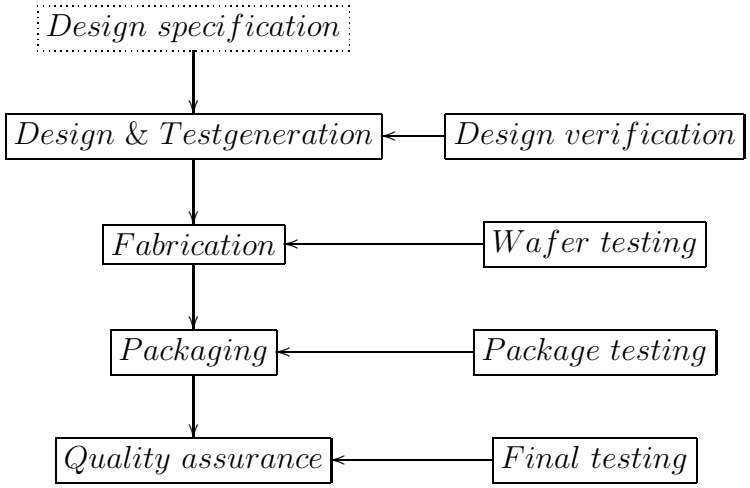


Figure 1.2: VLSI design flow [52]

1.1.2 Scan Architecture and Scan Testing

Scan testing is widely used DFT technology. The basic motivation behind the scan design is to improve the testability of the circuit by improving the controlability and observability. In scan design the sequential circuit is converted in to scan design by designing the normal flip-flop as scan flip-flop and connecting them each other in a shift register fashion. Now, the test pattern for the given sequential circuit can be generated using combinational ATPG. Scan design operates in three different modes: **normal mode**, **shift mode**, and **capture mode** [52].

In normal mode, all the test related signals are disabled and the circuit is operated in functional mode. In shift mode, scan test vector are shifted in to the scan chain and the test response are shifted out simultaneously. In capture mode, the shifted in test vector are applied to the combinational block and the response are captured.

Scan Architecture

Among the many varieties of scan architecture proposed, the *full scan design*, *partial scan design*, and *random access scan design (RAS)* are the basic scan desing.

In full-scan architecture all the sequential cells are used as scan cells. The full-scan architecture can be implemented using *multiplexed D flip-flop, clocked-scan cells, or LSSD (Level sensitive scan design) scan cells*. Full-scan design provides the advantage of achieving highest possible fault coverage for a given ATPG algorithm compared to any other variation of its. The main disadvantage of full scan design is longer test application time and excessive toggling in scan cells as it requires serial shift operation. Some of the alternative design some what better in test application time and power dissipation are partial scan design and RAS.

Unlike full-scan design where all the sequential cells are used as scan cells, partial-scan design requires only a subset of the sequential cells to be used as scan cells in scan chain. It can also be implemented using *multiplexed D flip-flop, clocked-scan cells, and LSSD (Level sensitive scan design) scan cells*. The test pattern for partial-scan design can either be generated by combinational ATPG or by sequential ATPG.

Full-scan and partial-scan design are based on serial shift operation. The serial shift operation creates the switching in scan cells which causes power dissipation. To avoid excessive switching a RAS is proposed. The idea of RAS is to access only one cell at time. Thus, RAS reduces the power drastically. The organization of RAS is similar to RAM (random access memory). An address decoder consist of column address decoder and row address decoder is used to address each of the scan cells.

Scan Testing

Stuck-at fault Testing: The stuck-at fault testing is performed to test the stuck-at fault. This type of testing can be categorized into two types: internal testing and external testing. In case of internal testing the test related operation is performed using BIST (Built In Self Test) whereas in case of external testing ATPG and ATE (Automatic Test Equipment) are used.

The stuck-at testing for scan design (circuit with scan chain) consist of four basic phases: *Test generation* carried out by ATPG, *test application*, *response capture* and *response comparison*, all are performed by ATE. The test application step involves the shifting in and launching of test pattern. During shift operation the SE (Scan Enable) signal is kept high so that the circuit can operate at scan mode. The shift operation is performed at slow clock rate generally half the functional clock. This slow shifting is performed to guarantee the average power consumption below the specified limit. Once the test patterns are shifted in, the primary input patters are applied and the SE signal is disabled so that the circuit can now operate in normal mode. In the normal mode operation the responses from the combinational circuit are captured in scan flip flop. The capture operation of stuck-at fault testing is performed at slow clock rate. Once the responses are captured they are shifted out while shifting in another test pattern. The shifted out responses are compared with the previously generated gloden ration by ATE for circuit to be declared as faulty or fault free.

Skewed-load Delay Testing: Unlike the stuck-at testing, the delay testing requires a

test vector pair $\langle V_1, V_2 \rangle$ to be applied to test a set of delay faults. The first vector V_1 of the pair is used to set the circuit to an initial state and the second vector V_2 is used to launch the transition. The second vector V_2 of the delay test pair is one bit shift over the first vector V_1 in the pair. The capture cycle for this test application is now divided into two pulse: launch and capture pulse. In launch pulse the transitions are launched by the application of V_2. The SE signal is now disabled and the capture operation is performed at functional clock speed [41]. Once the responses are captured they are shifted out while shiftiing in new test pattern. Finally the shifted out responses are compared with the gloden responses.

One of the difficulty in skewed-load scan methodology is timing critical consideration of SE signal [29]. The power related problem that arises during the at-speed testing is discussed in Section 4.

Broad-side Delay Testing: A broad-side delay testing is a form of a scan-based delay testing, where the first vector V_1 of the pair $\langle V_1, V_2 \rangle$ is applied to the scan chain to initialize the chain and the test vector V_2 is applied to launch the transitions. The vector V_2 of the pair is the combinational circuits response to the first vector V_1. This form of delay testing is called as broad-side since the second vector of the delay test pair is provided in a broad-side fashion, namely through the logic [42]. The broad-side test is advantageous over the the skewed-load test because in this case the signal SE is no more a timing critical due to an additional dead cycle which allow SE to settle [29].

1.2 Power Dissipation During Scan Testing

Power dissipation in digital CMOS circuit is caused by four sources [33]:-

- The *leakage current*, which is primarily determined by the fabrication technology, consists of two components: 1) reverse bias current in the parasitic diodes formed between source and drain diffusions and the bulk region in a MOS transistor, and 2) the subthreshold current that arises from the inversion charge that exists at the

gate voltages below the threshold voltage

- The *standby current*, which is the DC current drawn continuously from V_{DD} to ground
- The *short-circuit* (rush-through) current, which is due to the DC path between the supply rails during output transitions
- The *capacitance current*, which flows to charge and discharge capacitive loads (C_L) during logic changes

The above source of power dissipation are explained briefly in the following sections.

1.2.1 Dynamic Dissipation

The term *dynamic power* dissipation refers to the sum of *capacitive* and *short-circuit* dissipations.

Each time the capacitor C_L gets charged through the PMOS transistors, its voltage rises from 0 to V_{DD}, and a certain amount of energy is drawn from the power supply. Part of this energy is dissipated in PMOS device, while the remainder is stored on the load capacitor. During the high-to-low transition, this capacitor is discharged, and the stored energy is dissipated in the NMOS transistor.

The energy E_{VDD}, taken from the supply during the transition, as well as the energy E_C, stored on capacitor at the end of the transition, can be derived by integrating the instantaneous power over the period of interest. Equation 1.1 and 1.2 show the energy drawn from V_{DD} and the actual energy stored in C_L respectively. For these equations it is assumed that the NMOS and PMOS devices are never on simultaneously.

$$E_{VDD} = \int_0^\infty i_{VDD}(t) V_{DD} dt = V_{DD} \int_0^\infty C_L \frac{dv_{out}}{dt} dt = C_L V_{DD} \int_0^{V_{DD}} dv_{out} = C_L V_{DD}^2 \tag{1.1}$$

and

$$E_C = \int_0^\infty i_{VDD}(t) v_{out} dt = \int_0^\infty C_L \frac{dv_{out}}{dt} v_{out} dt = C_L \int_0^{V_{DD}} v_{out} dv_{out} = \frac{C_L V_{DD}^2}{2} \quad (1.2)$$

From above equations it can be observed that the energy stored at C_L, E_C is half the energy supplied at V_{DD}, E_{VDD}. This means that only half of the energy supplied by the power source is stored on C_L. The other half has been dissipated by the PMOS transistor. During the discharge phase the charge is removed from the capacitor, and its energy is dissipated in the NMOS transistor. In summary, each switching cycle (consisting of an low-to-high and high-to-low transition) takes a fixed amount of energy, equal to $C_L V_{DD}^2$. In order to compute the power consumption, we have to take into account how often the device is switched. If the gate is switched on and off $f_{0\to1}$ times per second, the power (dynamic) consumption is given by

$$P_{dyn} = C_L V_{DD}^2 f_{0\to1} \quad (1.3)$$

For a complex circuit, the equation for dynamic power, Equation 1.3, can be re-written as

$$P_{dyn} = C_L V_{DD}^2 P_{0\to1} f \quad (1.4)$$

where f is the maximum possible event rate of the inputs (which is often the clock rate) and $P_{0\to1}$ the probability that a clock event results in a $0 \to 1$ event at the output of the gate [34].

Power Dissipation due to Direct Path Current: The direct-path power is dissipated when both PMOS and NMOS transistor are conducting the current. This power is directly proportional to the switching activity, similar to the capacitive power dissipation [34].

The set of methodologies proposed in this dissertation are concentrated on the dynamic power dissipation.

1.2.2 Static Dissipation

The term *static power* (or steady-state power) dissipation refers to the sum of leakage and standby dissipations. The static power dissipation is expressed by the relation

$$P_{stat} = I_{stat} V_{DD} \tag{1.5}$$

where I_{stat} is the current that flows between the supply rails in the absence of switching activity.

Ideally, the static current of the CMOS inverter is equal to zero, as the PMOS and NMOS are never on simultaneously in steady-state operation. However, there is a leakage current flowing through the reverse-biased diod junctions of the transistor, located between the source and drain of the substrate. This contribution is, in general, very small and can be ignored.

An emerging source of leakage current is the subthreshold current of the transistor. The subthreshold current increases as the threshold voltage approaches towards zero volt and the larger the static power consumption [34].

The emerging DSM (deep sub micron) technology has brought the concern for static power dissipation. The static power dissipation is now becoming a major concern for devices at 0.1 micron or below [26].

1.2.3 Metrics to Measure Power

Energy: The total switching activities generated during the period of test application. Energy affects the battery lifetime during power up or periodic self-test of battery-operated devices.

Average power: Average power is the total distribution of power over a time period. The ratio of energy to test time gives the average power. Elevated average power increases the thermal load that must be vented away from the device under test to prevent structural damage (hot spots) to the silicon, bonding wires, or package.

Instantaneous power: Instantaneous power is the value of power consumed at any given instant. Usually, it is defined as the power consumed right after the application of a synchronizing clock signal. Elevated instantaneous power might overload the power distribution systems of the silicon or package, causing brown-out.

Peak power: The highest power value at any given instant, peak power determines the components thermal and electrical limits and system packaging requirements. If peak power exceeds a certain limit, designers can no longer guarantee that the entire circuit will function correctly. In fact, the time window for defining peak power is related to the chips thermal capacity, and forcing this window to one clock period is sometimes just a simplifying assumption. For example, consider a circuit that has a peak power consumption during only one cycle but consumes power within the chips thermal capacity for all other cycles. In this case, the circuit is not damaged, because the energy consumed will not be enough to elevate the temperature over the chips thermal capacity limit (unless the peak power consumption is far higher than normal). To damage the chip, high (not only highest) power consumption must last for several cycles [19, 33].

1.2.4 Scan Shift and Capture Power

Scan testing involve the shift and capture operation as it is discussed in Section 1.1.2. The shift operation shift in and out the test patterns and responses. While shifting in and out the test patterns and responses the scan cells changes their data in each shift clock. The continuous changes of data in scan cells result into the large number of switching in combinational part of the design. This excessive switching causes the proportionate amount of dynamic power dissipation. The power dissipated during the scan operation can be called as "scan power". The scan power can be classified into two classes based on the scan period: 1) Shift power and 2) Capture power.

Shift Power: The power dissipated during the shift operation is know as *shift power*.

In general the shift power includes the power dissipated over all the shift cycle including the last shift cycle. In literature shift power is studied in terms of peak and average power [19]. The peak shift power is computed as the maximum power dissipated over all the shift cycle for all the vectors (both test and response) [14, 49]. The average power is the total energy consumed for all the vectors over the total shift periods [39].

Capture Power: The power dissipated during the capture cycle is known as *capture power*. For stuck-at fault testing and skewed-load delay testing there can be only one capture cycle assuming the scan design as full scan design and for broad-side testing there are two capture cycle known as launch pulse and capture pulse[1]. The capture power dissipated due to application of a test vector in case of stuck-at and skewed-load testing is the Hamming distance between the test vector and its corresponding response. The capture power for broad-side test can be categorized into two: power dissipated during launch pulse and power dissipated during capture pulse. The power dissipated during launch pulse is due to the application of second test vector V_2 of test vector pair to launch the transition. The power dissipated during the capture pulse is due to the difference between test vector and its response which is approximated by Hamming distance between them.

1.3 Thesis Contributions and Organization

The primary goal of the dissertation is to provide the solutions for peak power minimization during scan testing. Overall, the dissertation consists of three original solutions. Each of the solutions target the excessive peak power during scan operations. First two proposals assume the full scan circuit without considering the multiple module test scheduling. The last proposal however, takes the test scheduling into consideration. The contribution of this dissertation work in nutshell is summarized in following bullets.

- A graph theoretic approach to formulate the test vector reordering methodology

[1] refer Section 1.1.2

- Formulation of scan chain reordering methodology for test-cycle peak power reduction as a graph theoretic problem
- Developement of efficient scheduling algorithm for capture power reduction in SoC testing

The previous work on minimization of average power and peak power are explained in Chapter 2. The Chapter 2 also discusses the motivation behind this dissertation work to reduce the peak power during scan testing which has not been explored yet by any of the previously proposed methodology. The Chapter is organized in two sections based on the motivation and direction of each of the previously proposed work. Each section are again divided into separate subsections to classify the different approaches.

Chapter 3 consists of the work on peak power reduction during scan testing using test vector reordering methodology. Section 3.1 gives the introduction about the problem of peak power and motivation. The proposed problem formulation is presented in Section 3.2 and the corresponding algorithms for each problem formulated are presented in Section 3.3. For each algorithm the corresponding experimental results on benchmark circuit are provided in Section 3.4. The Chapter is concluded in Section 3.5 with conclusion and possible future direction.

Chapter 4 is on reducing peak power during test cycle of scan operation. The Chapter starts with introduction and motivation for targetting the peak power particularly at test cycle. In Section 4.2 brief introduction about the peak power during test cycle for at-speed test, and the basic understanding about scan chain reordering for power minimization are explained. A graph theoretic problem is formulated in Section 4.3. Section 4.4 provides the proposed algorithm for construction of Hamiltonian path. Time and space complexity are analyzed in Section 4.5. The experimental results on ISCAS89 and ITC99 benchmark circuits is provided in Section 4.6. The Chapter is concluded in Section 4.7 with conclusion and further work.

The proposal on capture power reduction for modular SoC test is presented in Chapter 5. The introduction and motivation are given in Section 5.1. Section 5.2 presents the

formulation of problem and proposes two technique to deal with capture power. The experimental results are provided and discussed in Section 5.3. The Chapter is concluded in Section 5.4.

Finally, Chapter 6 summarizes the dissertation work. This chapter discusses the advantages and limitations of the proposed approach and shows the future direction to overcome those limitations. The primary contributions of this dissertation work is discussed in this chapter. Possible further work and new problems are also discussed.

Chapter 2

Motivation and Related Works

2.1 Motivation

In Introduction chapter the importants of peak power reduction is highlighted. Due to variation in design parameters like feature size, operating frequency, and complex functionality the peak power consumption during testing is gradually increasing along with the change in these parameters [8, 25, 55]. The excessive peak power during scan testing causes unnecessary yield loss and burn out of chip. The excessive peak power means the huge requirement of current which causes the IR-drop. The IR-drop inturn increases the transistor switching delay. Due to this delay the at-speed capture operation of the scan testing will not be able to capture the correct response in specified capture cycle. Hence, this incorrect response causes the chip to be identified as faulty whereas in actual the chip is fault free [43]. More over this excessive peak power during shift operation also puts limit in scan shift speed. It is discussed by P. Girard [19] that the peak power also determines the thermal and electrical limits of circuit under test. If the peak power dissipation remains high for consecutively sufficient number of cycle then it may cause the thermal related issues. Hence to avoid all these problems it is required to investigate deeper into the possible solutions.

The problem of peak power is also equally important incase of the modern multi-module system on chip (SoC) test. The multi-module SoCs are generally tested in

parallel to reduce the over all test time. The parallel testing of SoCs are constraint by two parameters: the area and the power. Hence in this case also the reduction in excessive peak power can results into reduction in test application time.

Many solutions have been proposed to address the power problem during testing and particularly for scan based circuit because the scan testing is most popular DFT methodology for industrial practice. In this chapter some of the earlier low power scan methodologies are discussed. The chapter is divided into two primary sections for categorising the work those deals with average power and peak power separately. Each of the section is further divided into four subsections to classify the different work based on the particular methodology.

2.2 Minimization of Average Power

Several solutions have been proposed for average power reduction in various direction such as, scan masking [18], scan partitioning [32, 44, 53], circuit modification [27, 46], ATPG [7, 39], vector reordering [12, 22], and scan cells reordering [5, 20]. Along with those above mentioned technique the average power can also be controlled during the shift operation by shifting the scan patterns with low frequency test clock. However, this low frequency test clock affects the test application time. Following sections explains the methodologies proposed in above mentioned work.

2.2.1 Circuit Modification Techniques

The work on scan masking approach to reduce the power consumption during scan shift operation is proposed by Gerstendorfer et al. [18]. This approach could able to reduce the scan shift power (both the average and peak) significantly. However, the scan masking methodology has its own limitation, it will not be able to control the power dissipated during *capture cycle* [1] and it increases the path delay because it inserts the masking circuit to check the toggling effect.

[1] refer Section 1.1.2 for definition of *capture cycle*

The approach proposed by Whetsel [53], Nicola et al. [32] and Saxena et al. [44] have explored the scan partitioning methodology for average power reduction. The scan partitioning technique partitions the scan chain into two or more segments. Each of the segment are operated individually while shifting in the scan vector. Since only a portion of the scan chain is active at a time the power dissipated during the shift operation is inverse proportionate to the number of partition. This technique however, require extra area for implementing the clock distribution circuit.

The methodology proposed in [46] modifies the scan cells by inserting some additional logic to control the transition that arises during the shift operation. This methodology does not degrades the circuit performance because the additional logic are not added on the functional path rather on scan path. This methodology is capable of reducing both the average and peak power. The methodology proposed in [27] divides the circuit into multiple partition and test each of the partition independently. The independent testing of individual partition is the key in reducing the average power. The division of the circuit is performed at RTL level.

2.2.2 ATPG Based and Test Vector Modification Techniques

A pattern generation and DFT based technique is proposed by Butler et al. [7] to deal with power during scan testing. The proposed methodologies reduces both the average as well as peak power. In pattern generation technique, the X-filling heuristic known as "fill adjacent" is used to fill the unspecified X-bits in the order of scan chain from scan-out to scan-in. And in DFT based technique, the design partition methodology is used to reduce the switching in untested module.

Another work by LI et al. [28] explored the same idea of X-filling to reduce the average power and capture power. The idea in this proposal was that, the minimum possible number of Xs can be used to minimize the peak power only upto the required threshold limit while rest of the available Xs can be used to reduce the average power. The static test vector compaction technique is explored by Sankaralingam et al. [39], this work is also based on the X-filling methodology.

Another work by V. Devanathan et al. [13] proposed an ATPG based low power methodology. They have proposed a new pattern generation technique which consists of the technique of statistically profiling the scan patterns and rejecting the scan patterns those does not satisfy the power safety criterion.

2.2.3 Methodologies on Test Vector Ordering

Reordering of test vector is another technique that is suitable to reduce the power. The advantages of this approach is: it does not add any additional area and it can be done without interfering the design and test flow. P Girard et al. [21, 22] have formulated a graph theoretic problem for test vector reordering to reduce average power.

Another work by Dabholkar et al. [12] has achieved average power reduction using test vector and scan latch reordering methodology. The problem of test vector reordering and scan latch reordering was formulated as graph problem and solved by greedy ATSP heuristics, simulated annealing, and Christofied algorithm.

2.2.4 Methodologies on Scan Chain Ordering

Scan chain reordering is one of the efficient technique for reducing power; both, the average and the peak power. However, this technique comes up with power and routing area tradeoff. The reordering of scan chain also puts impact on transition fault coverage, the impact can be either increase or decrease in fault coverage. Following are some of the proposals based on scan chain reordering.

The proposal by Girard et al. [20] and Bonhomme et al. [5] formulated the scan reordering problem as a graph theoretic problem. The problem of finding the best ordering of scan chain is combinatorial problem which is NP-complete. As the problem is NP-complete they have solved it using a greedy based heuristic algorithm. Furthermore in [20] a methodology to fix the tradeoff between power and routing area is proposed.

2.3 Minimization of Peak Power

Most of the proposals on average power also achieved reduction in peak power as a by-product. However, the emergence of deep submicron technology has brought a deeper concern for peak power during shift and capture cycle of scan operation. Particularly in case of at-speed testing the peak power becomes an important issue. Till date various direction have been explored for peak power reduction. Some of the earlier proposals are explained in following subsections.

2.3.1 Circuit Modification Techniques

Sankaralingam et al. [40] have proposed a technique based on circuit modification to minimize peak power during shift and capture cycle. The technique is based on the insertion of test point at the out put of scan flip-flops. The function of the test point is to block the transitions from propagating to the combinational block. The main idea of this work is to selectively choose the few numbers of flip-flops on which the test point can be inserted. The proposed methodology is designed to avoid inserting the test point in critical path. Further, the interesting contribution of this method is that it can also control the peak power during capture cycle. However our proposed methodology, the test vector ordering, the scan chain ordering, and the test scheduling can be complement to this kind of methodolgy because the test point that is inserted in the out put scan flip flops is indepenedent of test patterns and scan chain architecture.

Rosinger et al. [35] have proposed a scan partition technique to deal with the peak power during shift cycle and capture cycle. The idea behind this proposed methodology is similar to the original idea that was proposed by Whetsel et al. [53] and latter extended by J. Saxena et al. [44]. The main idea is to partition the scan chain in to different segments and operate each of the segment in non-overlapping fashion. In the earlier works of L. Whetsel and Saxena et al. the scan chain was splitted into length-balanced segments and enabling only one in each shift clock. Both of these earlier works performed well in reducing the average power and peak power during shift operation. However,

in both of these approaches the capture clock is applied simultaneously to every scan segments which limits the technique from reducing the capture power. Based on this observation Rosinger et al. proposed an idea extended in this line to overcome the limitation of simultaneous capture operations. The idea proposed by Rosinger et al. is to split the scan chain in to length-balanced segments, and only one segment is enabled during shift as well as capture cycles. The idea is realized by exploring the clock gating technique. This technique is shown effective in reducing both the power dissipated during shift operation and the capture power. More over this technique does not depend on ATPG algorithm. It can reuse the patterns generated by ATPG without altering fault coverage. The minor drawbacks of this methodology is test time overhead and area overhead.

Our proposed test vector reordering methodology can still be applied to these kind of scan architecture. Now the idea of test vector reordering may not be as simple as in the case of plain full scan architecture but with small modification the test patterns can be reordered to further minimize the peak power. The test pattern reordering has to be applied to each segment of the scan chain individually. Similarly the scan chain reordering technique can be applied to each of the scan segments where the scan cells of a particular segment can be reordered irespective of the other segments. The segmented scan architecture enhances the perfomance of the test scheduling approach because now each of the segment can be scheduled simultaneously and at the same time the technique to avoid the capture cycle coincidence can also be applied to each segment separately.

2.3.2 ATPG Based and Test Vector Modification Techniques

Corno et al. [10] have proposed an ATPG technique for sequential circuit to deal with peak power. The technique proposed is based on the idenfication of primary inputs which can be set to don't care bit without affecting the fault converage and specifying these inputs to minimize the power consumption. Based on this idea they have proposed a two steps, test sequence relaxation and test sequence reconstruction methodology. In the first step a set of primary inputs are indetified which can be set to don't care bit

and in the second step the identified unspecified inputs are assigned to appropriate bit. Although the authors proposed this technique for non-scan circuits, this can also be applied to circuit with scan dft. However, this kind of approach does not limit the test vector reordering and scan chain reordering technique. Both of these technique can still be applied without putting any side effect on this kind of techniques. Simialarly the test scheduling also can be applied along with this kind of techniques.

A post ATPG methodology to reduce peak power during shift and capture operation is proposed by Sankaralingam et al. [14]. In this work four type of power violation problems are identified; Scan capture, Order dependent, Scan-in, and Scan-out. Each of the problem are then solved by proposed bit-stripping, X-filling and test vector swapping heuristic. The scan capture power violation can occur if the peak power dissipated during the capture operation is higher than the threshold limit. The scan capture power as it is defined by author is approximated by hamming distance between the applied test vector and its response. The order dependent power problem is due to the particular ordering of test vectors. The scan-in power violation occurs during the scan in of test patterns. The scan-out power violation occurs during the scan out of test reponses. To eliminate the power violation they have used three techniques. The bit-stripping technique is used to identify the particular bits in test vector which can be set to X value without affecting the fault coverage. Once the X bits are identified next a technique called as minimum transition fill [1] is applied to fill the Xs which minmizes the total number of transitions. This technique is applied to eliminate the scan-in, scan-out, and scan capture power violation. The order dependent power violation is eliminated by vector swapping method.

Our proposed test vector reordering methodolgy is targetted to minimize the order dependent power violation. In our methodology the reordering of test vector is done in more formal and exhustive way by formulating the problem as a travelling sales person problem (TSP) on complete directed weighted graph.

Badereddine et al. [1, 2] have proposed a scan chain stitching and X-filling heuristic known as adjacent fill also known as MT-fill (Minimum Transition-fill) for peak power

reduction. In this section we will discussing about the scan X-filling technique and the scan stitching is discussed in the Subsection Methodology for Scan Chain Ordering. The X-filling technique proposed here to reduce the peak power is one of the effective methodology to minimize the power. Here, the idea is to use a test generation process during which non-random filling is used to assign values to don't care bits (Xs) of each test pattern of the deterministic test sequence. For example, it is possible to apply the following non-random filling heuristics:

- Don't care '0': all don't care bits in a test pattern are set to '0'
- Don't care '1': all don't care bits in a test pattern are set to '1'
- Adjacent filling: all don't care bits in a test pattern are set to the value of the last encountered care bit (working from left to right).When applying adjacent filling, the most recent care bit value is used to replace each X value. When a new care bit is encountered, its value is used for the adjacent Xs

For example, consider a single test pattern that looks like the following: 0XXX1XX0XX0XX. If we apply each one of the three non-random filling heuristics, the resulting pattern will be:

- 0000100000000 with '0' filling
- 0111111011011 with '1' filling
- 0000111000000 with adjacent filling.

The X-filling technique is very much compatible with the test vecor reordering and scan chain reordering technique. Similarly the test scheduling technique which uses test vector swapping and idle cycle insertion methodology can also be applied on the MT-filled test patterns.

2.3.3 Methodologies for Test Vector Ordering

A proposal by Girard et al. [21] was primarily formulated for average power reduction, however this could also able to reduce the peak power as by-product. A vector swapping approach is proposed by Sankaralingam et al. [14] (this technique is discussed in detail

in Section 2.3.2) to minimize peak power during scan shift operation.

Hence from all these literature survey we observe that the test vector reordering technique has not been explored to fullest extent. This forms our basic motivation behind formulating the test vector reoordeing technique as graph theoretic problem ro deal solve the peak power problem.

2.3.4 Methodologies for Scan Chain Ordering

The scan chain reordering is another domain to explore for peak power minimization. However this domain was left unexplored till now, very few proposals have been made in peak power reduction during *test-cycle*. Badereddine et al. have explored this domain in [2, 3]. They have formulated the scan reordering problem as a constraint global optimization problem. The objective of formulation is to reduce the peak power during *test-cycle*. To solve the problem they have proposed a heuristic based on simulated annealing approach. Although the simulated annealing can provides near optimal solution - if it is allowed to run for sufficient number of iteration - the graph theoretic formulation will wider the solution space for scan reordering problem. The scan reordering methodology may alter the delay fault coverage for skewed-load testing, however it is reported in [3] that the change is very nominal.

Wu et al. [54] proposed a graph theoretic formulation of scan cells reordering for average power reduction. In this work, the Xs are preserved during the time of scan cells reordering for further optimization using MT-fill. This work also reduces peak power consumption along with average power as by-product.

2.4 Test Scheduling Approaches on SoC Test Power Minimization

Chou et al. proposed the usage of a single power value for each block of logic [9]. In the test scheduling process, the objective is to schedule the testing of blocks such that the

total test application time is minimized while the power consumption at any time is below a given power constraint. Zorian proposed for BIST cores also a scheduling technique under the peak power model [55]. However, the single power model is pessimistic and makes the schedules unnecessarily longer in test time. Rosinger et al. therefore have proposed a double-value test power model (one value representing a constant low power consumption, and the other value representing a constant high power consumption for a test) [36]. The power model was made further accurate by the cycle-accurate power model proposed by Sami et al. [37, 38]. Instead of one or two values per test, a power value per cycle is used in order to find tighter test schedules with lower test application time. Sami et al. also have proposed a test architecture and test scheduling technique that defined the test access mechanism (TAM) and assigned tests to TAMs such that the total test time is minimized.

Chapter 3

Test Vector Ordering to Minimize Peak Power

3.1 Introduction and Motivation

Power consumption during testing of scan circuit is an important issue to address for today's very complex sequential circuits. It becomes especially important when chips are designed with small feature size and higher frequency; hence, at-speed test becomes necessary. Excessive average power results into burn out of chip whereas excessive peak power results into power droop problem which can falsely classify a good chip as a faulty chip. Average power can be reduced by reducing clock frequency. Reduction of peak power during test becomes very important for two main reasons: 1. Higher peak power causes loss of yield due to power droop and cross talk, 2. If the scan circuit is a module in an SoC then multiple module can be scheduled together to minimize the test time. Apart from these two reasons the peak power can also induce thermal related damage to the chip if the peak power remain high for sufficient number of cycle [20].

The problem of test power reduction is an active area of research for quite sometime. Most of the solutions [9, 12, 39] are aimed at average power reduction. However, they have also achieved small amount of reduction in peak power as a by-product. Test vector reordering methodology proposed in [12] was meant for average power reduction.

Bonhomme et al. [5] used scan chain reordering technique to minimize the total number of transitions to reduce average power. Scan chain reordering technique is one of the effective way of minimizing the power consumption during scan operation. However, the scan reordering methodology has its inherent shortcoming of consuming extra interconnect area. In [23] and [18], additional logic is added to hold the output of the scan cells at a constant value during scan shifting thereby reducing the power dissipation. This approach greatly reduces the average power, and will avoid peak power problems during scan shifting. However, this category of approaches lead to extra area overhead. Moreover, these kind of approaches degrade circuit performance because its add extra logic in the functional paths.

Another set of solutions which have been proposed in the literature are the low activity pattern generation using ATPG. Corno et al. [10] proposed a test pattern generation technique which modifies test sequence for sequential non-scan testing for reducing peak power. The approaches proposed by Shankaralingam et al. [14] and Wen et al. [15] specifies the don't care bits of the generated test cubes in such a way that it can reduce the peak power. Badereddine et al. [1] proposed a solution based on a power-aware assignment of don't care bits, an experiment is carried out to verify the effect of MT-fill algorithm for peak power minimization. We see that the test vector reordering can still be applied to the X-filled patterns. Hence further reduction in power can be achieved.

As most of the current designs are getting increasingly complex and timing sensitive, peak power consumption during scan testing should be under the allowable threshold peak power limit. Although the problem is important enough to address at this point, there is not much work been done to address. In this work we are proposing a test vector reordering methodology to minimize the peak power consumption during scan shift and launch/capture cycle. The test vector reordering methodology is formulated in more formal way as a graph theoretic problem. The graph theoretic formulation consists of three different problems. The first problem is formulated as bottle-neck TSP problem on directed complete weighted graph and solved using a simple algorithm. The second

problem is formulated as a minimum weight directed walk problem on a complete directed weighted graph. The formulated minimum weight walk problem is solved using a linear time heuristic. And the third problem is formulated as an extended walk problem on an extended graph. The extended walk problem is then solved using a heuristic based on path cover approach. At the end, we are also defining a lower bound on minimum achievable peak power.

The rest of the Chapter is organized as follows. Section 3.2 presents the formulation of three different problems and describes the lower bound on peak power. In Section 3.3 the corresponding algorithm for each problem are presented. Section 3.4 provides the experimental results and analysis . The Chapter is concluded in Section 3.5 with concluding remarks and future work.

3.2 Construction of Directed Weighted Graph and Problem Formulation

This section describes the procedure for constructing the directed weighted graph from a given scan test information and it describes the formulation of graph theoretic problem on the constructed graph. The abstract level problem for the test vector sequence which minimizes the peak power consumption is stated below.

Problem Statement: Determine the minimum achievable peak power during application of test for a given test set and find a test vector sequence which can support the minimum achievable peak power.

The directed weighted graph is constructed by representing each of the scan test elements such as test vectors, order of applying the test vectors and peak power into the corresponding graph elements. The procedure for constructing the weighted directed graph D_c is given below.

- Each test pattern T_i corresponds to the node N_i in the graph
- The application of test vector T_i followed by T_j corresponds to the directed edge

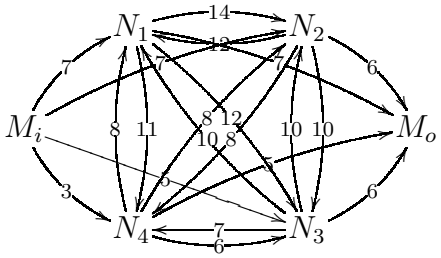


Figure 3.1: A complete weighted digraph for pattern set in Example 3.1

E_{ij}, from node N_i to node N_j

- The maximum number of transitions that occur in scan chain per clock cycle for a complete scan shift operation while applying test pattern T_i followed by T_j is represented by the weight EW_{ij} of the edge E_{ij}.

Since any pattern can follow any other pattern during the scan shift operation, the constructed graph would be a *complete directed weighted graph*. Example 3.1 help understanding the construction of graph. The Example 3.1 is used as a running example in this Chapter.

Example 3.1. Let length of scan chain be 15 and total number of patterns be 5. Let following be test patterns and responses.

T_1	1 1 1 1 1 1 1 0 1 0 1 0 1 1 1	R_1	1 1 1 0 1 0 1 0 1 1 1 1 0 0 0
T_2	1 1 1 1 1 1 1 1 1 0 1 0 1 0 1	R_2	1 0 1 0 1 0 1 1 1 1 1 1 1 1 1
T_3	1 1 1 1 1 1 1 1 1 1 1 0 1 0 1	R_3	1 0 1 0 1 1 1 1 1 1 1 1 0 0 1
T_4	1 1 1 1 1 1 1 1 1 1 1 1 1 0 1	R_4	1 1 0 1 1 1 1 1 1 1 1 1 0 1 1 0

Figure 3.1 shows the complete directed weighted graph constructed using the above given test set. In this figure two additional dummy nodes are introduced. The dummy node M_i is introduced to represent the scan in of first test vector while the scan chain is in reset state and M_o is introduced to represent the scan out of last response.

Three problems are formulated to obtain the minimum peak power and the order of applying the test vectors for a given test vector set. The first problem, minimum peak power Hamiltonian path, is formulated to provide the test vector sequence without test

application time penalty. The second problem, minimum peak power walk, is formulated to reduce the peak power below the peak power achieved by the first method with reapplication of some test vectors, hence, this may consume extra test cycles. And the third problem, minimum peak power extended-walk, is formulated to reduce the peak power below the peak power achieved by first and second method. The formulation of third problem introduces new low activity test vectors, hence, this also consume extra test cycles.

3.2.1 Minimum Peak Power Hamiltonian Path Problem

The minimum peak power Hamiltonian path is formulated as a bottle-neck TSP problem [17]. The detail formulation is followed from the following definition and problem statement.

Definition 3.1. ***Path-weight:*** *Path-weight is defined as the maximum of weight of each edge in the path in a digraph.*

Problem Statement: Given a complete weighted directed graph D_c, find a Hamiltonian path which has minimum *path-weight*.

The following illustrates the above problem statement. Consider three different Hamiltonian paths, Path1, Path2, and Path3 from Figure 3.1,

Path1: $M_i \xrightarrow{7} N_1 \xrightarrow{12} N_3 \xrightarrow{7} N_4 \xrightarrow{8} N_2 \xrightarrow{6} M_o$

Path2: $M_i \xrightarrow{7} N_2 \xrightarrow{8} N_4 \xrightarrow{6} N_3 \xrightarrow{10} N_1 \xrightarrow{7} M_o$

Path3: $M_i \xrightarrow{5} N_3 \xrightarrow{10} N_1 \xrightarrow{14} N_2 \xrightarrow{8} N_4 \xrightarrow{5} M_o$

Let PW_1, PW_2 and PW_3 be the corresponding *path-weight*. Then we have PW_1 = Max((N_0,N_1),(N_1,N_3),(N_3,N_4),(N_4,N_2)) = 12 and similarly PW_2 = 10 , PW_3 = 14. Thus *Minimum Peak Power* = Min(PW_1, PW_2, PW_3) = 10.

In order to solve this problem, the directed graph problem is transformed in to an unweighted directed graph problem. The transformation of a weighted digraph for a given peak power value, P_{th}, to an unweighted digraph is carried out as follows:

- Every node in weighted graph are preserved in corresponding unweighted graph

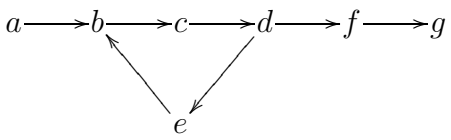


Figure 3.2: An unweighted digraph

- All the edges in weighted graph whose weight is greater than P_{th} are removed
- All other edges whose weight, $EW_{ij} \leq P_{th}$ are replaced with unweighted edges

The above stated problem statement can be restated for unweighted graph as follows: For a given complete weighted directed graph D_c, find the minimum peak power P^1_{th}, such that its corresponding unweighted graph D_u has at least one Hamiltonian path.

The above problem formulation will obtains the minimum possible peak power P^1_{th}. There may be more than one possible Hamiltonian paths in the unweighted graph. Although minimization of peak power also minimizes average power, it can be further minimized under peak power P^1_{th} by choosing a path that provides minimum average power. Note that the Hamiltonian path visits each and every node once and only once and hence the test application time will remain unchanged.

3.2.2 Minimum Peak Power Walk Problem

The minimum peak power walk problem is formulated to over come the barrier in the minimum peak power Hamiltonian path problem. The barrier in that problem was the constraint of one time visit of a node. The following observation is the basis of formulating this problem.

A directed graph, D, which does not contain any Hamiltonian path may have a possible *walk* which visits each and every node at least once [4]. This is illustrated by the scenario shown in Figure 3.2

It can be observed from Figure 3.2 that the Hamiltonian path does not exist in this graph, however, there is a walk from node a to node g which visit each and every node at least once with revisiting nodes e, b, c and d.

Definition 3.2. ***Walk-weight:*** *The walk-weight for a walk in a directed weighted graph is defined as the maximum edge-weight in that walk.*

Problem Statement: For a given complete weighted digraph D_c, find a walk which visits each and every nodes at least once with minimum *walk-weight*.

The problem can be redefined for unweighted graph as follows: For a given weighted directed graph D, find the minimum peak power value, P^2_{th}, such that its corresponding unweighted graph has at least one walk.

The possible walk will revisit few nodes; hence, it will increase the test application time. The increase in test time is proportional to the number of revisited nodes.

3.2.3 Minimum Peak Power Extended-walk Problem

The first problem is formulated to obtain a Hamiltonian path having minimum peak power P^1_{th} and the second problem is formulated to find a walk having minimum peak power P^2_{th} (less than P^1_{th}) at the cost of revisiting few nodes. If we further reduce P_{th} (minimum achievable peak power) the corresponding unweighted graph, D_u, will not have a walk to apply complete test set. Further reduction will result into disjoint paths. These paths can be further joined together to form a single walk by introduction of some extra low activity nodes (currently we consider only all 0s patterns and all 1s patterns whose corresponding nodes are N_{a0} and N_{a1}) to D_u. We call this walk with these additional nodes as *extended-walk* and the graph D_u with these additional nodes as *extended-graph*. Note that we only scan in these patterns and will not apply to the combinational block so that the scan out would have same pattern. The above stated scenario is illustrated in the Figure 3.3.

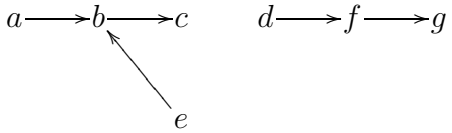


Figure 3.3: Disconnected unweighted digraph

In Figure 3.3 the possible disjoint paths would be $a \longrightarrow b \longrightarrow c$, $d \longrightarrow f \longrightarrow g$, and

e. After interconnecting these paths the resultant path will be as follows,

$a \longrightarrow b \longrightarrow c \longrightarrow x \longrightarrow d \longrightarrow f \longrightarrow g \longrightarrow y \longrightarrow e$, where x and y are the new nodes. Therefore, the problem can be defined in following way.

Problem statement: For a given complete weighted directed graph D_c, find the minimum peak power value, P^3_{th} such that its corresponding unweighted *extended-graph* has at least one *extended-walk*.

3.2.4 Lower Bound on Minimum Achievable Peak Power

This section will explain the minimum achievable peak power by the proposed methodology. Problem-3 is formulated to achieve the minimum possible peak power by inserting additional nodes N_{a0} (all 0's) and N_{a1} (all 1's). The minimum achievable peak power is defined in the following Theorem:

Theorem 3.1. *The lower bound on minimun achievable peak power for a test set is*

$$Max\forall_i[Max\{Min(EW_{a0i}, EW_{a1i}), Min(EW_{ia0}, EW_{ia1})\}]. \tag{3.1}$$

Proof. In order to achieve the minimum peak power the graph should be augmented by additional nodes N_{a0} and N_{a1} to construct an *extended-graph*. In *extended-graph*, the edge weight, EW_{ij}, is greater than or equal to $Max[Min(EW_{ia0},\ EW_{ia1}),Min(EW_{a0j},\ EW_{a1j})]$. Therefore, it is guaranteed that visiting the node j directly from the node i would be more or equal to visiting the node through either node N_{a0} or N_{a1}. In worst case, we may always need to travers through extended nodes while going from node i to node j. The minimum power consumed to traverse the node j from the node i would be $Max[Min(EW_{ia0},\ EW_{ia1}),\ Min(EW_{a0j},\ EW_{a1j})]$. Hence, the minimum achievable peak power for a given test set would be the maximum of minimum peak power computed for all nodes, that is $Max\ \forall_i\ [Max\ \{Min(EW_{a0i},\ EW_{a1i}),\ Min(EW_{ia0},\ EW_{ia1})\}]$. □

Lemma 3.1. *Let there be a test set of n vectors, and scan chain length be l. The worst case test time to achieve the lower bound is* $(3n-1)\times l + (n-1)$.

Proof. According to Theorem 3.1, it is always possible to achieve minimum power by traversing through N_{a0} or N_{a1}. Hence, it adds one extra node in traversal from node i to node j. However, it can be observed that, to achieve minimum power, node j can be reached from node N_{a1} or N_{a0}, and N_{a0} or N_{a1} can be reached from node i. Therefore, in the worst case we need to take the following path to reach node j from node i: N_i to N_{a0} to N_{a1} to N_j. We walk through additional $(2n-1)$ nodes in worst case. Therefore the worst case time would be: $(3n-1) \times l + (n-1)$. □

Corollary 3.1. *The minimun achievable peak power under maximum test time* $(2n-1) \times l + (n-1)$ *for a test set is* $Max\ \forall_i$ *[*Max *(*EW_{a0i}*,* EW_{a1i}*,* EW_{ia0}*,* EW_{ia1}*)]*

Proof. A walk from node i to node j under power constraint Max (EW_{a0i}, EW_{a1i}, EW_{ia0}, EW_{ia1}) can be made by the following traversal: V_i to V_{a0} to V_j, or V_i to V_{a1} to V_j. Hence, the worst case test time under above stated power constraint would be $(2n-1) \times l + (n-1)$. □

Note that the cost of walks from N_{a0} to N_{a1} to N_j, and from N_i to N_{a0} to N_{a1} to N_j will differ by at the most one transition as $Max\{Min(EW_{a0i}, EW_{a1i}), Min(EW_{ia0}, EW_{ia1})\}$ - $Max\ (EW_{a0i}, EW_{a1i}, EW_{ia0}, EW_{ia1}) = 1\ \}$. Therefore, it can be inferred from Corollary 3.1 that the test time can be reduced significantly on the cost of additional one transition in the worst case.

3.3 Proposed Algorithms

3.3.1 Minimum Peak Power Hamiltonian Path Algorithm

The following algorithm computes the Hamiltonian path and minimum peak power value P^1_{th}.

AlgoMinHam(Input:D_c, Output:HamPath,P^1_{min})

1 Let A be nondecreasing sorted array of edge-weights;

2 index = $SelectInitialIndex$;

3 while(TRUE)

4 $UpLimit = A$(index);
5 RemoveEdge(D_c, $UpLimit$, D_u);
6 found = SearchHam(D_u, HamPath);
7 if(found == TRUE)
8 then
9 $P_{th}^1 = UpLimit$;
10 return HamPath and P_{th}^1;
11 else
12 index = $SelectNextIndex$;
13 CONTINUE ;
EndofAlgo

Description of procedures:

AlgoMinHam(): This algorithm returns Hamiltonian Path and minimum peak power value P_{th}^1.

RemoveEdge(): This procedure removes all the edges of D_c having *edge-weight* > $UpLimit$ and returns the resultant unweighted digraph D_u.

SearchHam(): This procedure checks for at least one node having no incoming edges or at least two node having no outgoing edges. If this is the case then its return FALSE, otherwise it searches for a Hamiltonian path and if found it return TRUE else return FALSE.

Following example illustrates the functionality of algorithm AlgoMinHam.

Example 3.2. Let D_c be the digraph from Figure 3.1. The algorithm AlgoMinHam will return Hamiltonian path:

$M_i \xrightarrow{3} N_4 \xrightarrow{8} N_2 \xrightarrow{10} N_3 \xrightarrow{10} N_1 \xrightarrow{7} M_o$

and minimum peak power value $P_{th}^1 = 10$.

3.3.2 Heuristic for Minimum Peak Power Walk

The following algorithm find a minimum-weight *walk* if exist which visits every nodes of given digraph at least once.

AlgoMinWalk(Input:D_c, Output:$Walk$, P^2_{th})

```
1   Let A be nondecreasing sorted array of edge-weights;
2   index = SelectInitialIndex;
3   while(TRUE)
4      UpLimit = A(index);
5      RemoveEdge(D_c, UpLimit, D_u);
6      Success = AlgoConDag(D_u, DAG);
7      if(Success == TRUE)
8         AlgoConIpath(DAG, Ipath);
9         AlgoConWalk(Ipath, Walk);
10        P^2_th = UpLimit;
11        return Walk and P^2_th;
12     else
13        index = SelectNextIndex;
14        CONTINUE;
EndofAlgo
```

Description of procedures:

AlgoMinWalk(): This is the main procedures which returns $Walk$ which have *walk-weight* of value P^2_{th}.

AlgoConDag(): This procedure returns TRUE if a DAG is successfully constructed from given digraph D_u by forming each cycle into corresponding *supernode* [16] else it returns FALSE indicating that DAG cannot be constructed.

AlgoConIpath(): This procedure construct an intermediate path $Ipath$ from a given DAG by using the concept of acyclic ordering [4].

AlgoConWalk(): This procedure construct a walk $Walk$ from a given $Ipath$ by first

unrolling those *supernode* into cycle and then inter connecting each of the nodes.

The complexity of the algorithm is O($nodes^3$ + $edges^2$), which is polynomial. Following example illustrates the functionality of heuristic AlgoMinWalk,

Example 3.3. Let D_c be complete digraph from Figure 3.1. The unweighted digraph D_u after RemoveEdge() is shown in Figure 3.4. After obtaining this digraph, by executing rest of the methods, resultant $Walk$ and P^2_{th} will be, $Walk$:

$M_i \xrightarrow{7} N_2 \xrightarrow{8} N_4 \xrightarrow{6} N_3 \xrightarrow{7} N_4 \xrightarrow{8} N_1 \xrightarrow{7} M_o$

and $P^2_{th} = 8$ which is $< P^1_{th}$.

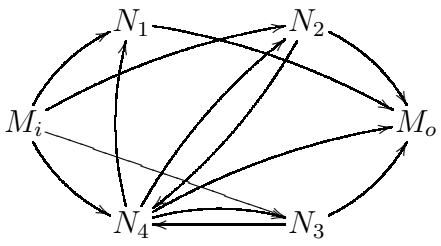


Figure 3.4: Unweighted digraph having edge-weights ≤ 8

3.3.3 Path Cover Heuristic for Minimum Peak Power Extended-walk

This algorithm find a minimum-weight, P^3_{th}, *walk* in an extended digraph.

AlgoMinEWalk(Input:D_c, Output:$EWalk$, P^3_{th})

1 Let A be nondecreasing sorted array of edge-weights;

2 index=*SelectInitialIndex*;

3 while(TRUE)

4 $UpLimit = A$(index);
5 RemoveEdge(D_c, $UpLimit$, D_u);
6 FindDisjointPath(D_u, C);
7 JoinPath(C, $EWalk$);
8 $TempWalk = EWalk$;
9 $P^3_{min} = UpLimit$;
10 if($LowerBound < P^3_{th}$)
11 index=$SelectNextIndex$;
12 CONTINUE;
13 else
14 return $TempWalk$ and P^3_{th};
EndofAlgo

Description of procedures:

AlgoMinEWalk(): This is the main procedure which returns a *walk* having minimum peak power P^3_{th}

FindDisjointPath(): This procedure finds the directed path cover.

JoinPath(): This introduces the new nodes, N_{a0} and N_{a1} to interconnect each path. Following criteria is applied to introduce the node N_{a0} and N_{a1}.

Let $Path_i$ be one path and $Path_j$ be another to be connected in order, $Path_i \longrightarrow Path_j$. Let first node and last node of $Path_i$ be F_i and L_i respectively and let for $Path_j$ be F_j and L_j. Let corresponding test vector for F_i, L_i, F_j, and L_j be T_{fi}, T_{li}, T_{fj}, and T_{lj}.

if(LSB(Response(T_{li})) == 1 & MSB(T_{fj}) == 1)

then connect $Path_i$ with $Path_j$ using N_{a1} node;

if(LSB(Response(T_{li})) == 0 and MSB(T_{fj}) == 0)

then connect $Path_i$ with $Path_j$ using N_{a0} node;

If both of the above condition become false then $Path_i$ and $Path_j$ are connected through randomly selected node.

Following example illustrates the functionality of heuristic AlgoMinEWalk.

Example 3.4. Let D_c be the digraph from Figure 3.1. The unweighted digraph D_u

obtain after RemoveEdge() is shown in Figure 3.5,

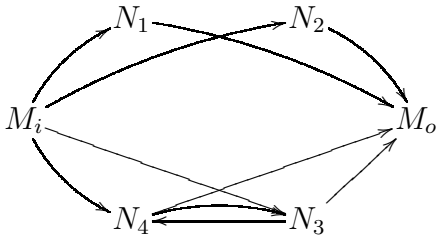


Figure 3.5: Unweighted digraph having edge weights ≤ 7

In this example the possible disjoint path would be $M_i \xrightarrow{3} N_4 \xrightarrow{6} N_3$, N_1 and $N_2 \xrightarrow{6} M_o$. Then the resultant *extended-walk* would be,
$M_i \xrightarrow{3} N_4 \xrightarrow{6} N_3 \xrightarrow{6} N_{a1} \xrightarrow{6} N_1 \xrightarrow{7} N_{a1} \xrightarrow{6} N_2 \xrightarrow{6} M_o$
and $P_{th}^3 = 7$ which is $< P_{th}^2$.

3.4 Implementation and Experimental Results

All the three algorithms are implemented in C. For experimental evaluation we have used ITC99 and ISCAS89 benchmark circuits. Scan insertion and synthesis is carried out using "DFT Compiler" and test patterns are generated using "TetraMAX" of synopsys. The experimental results are shown in Table 3.1, 3.2 and 3.3 respectively for AlgoMinHam, AlgoMinWalk, and AlgoMinEWalk. For each benchmark circuit, the peak transition due to given ordering of test patterns (given by TetraMax), peak transition due to proposed algorithms and percentage of improvement achieved are shown in tables. In Table 3.1, the average transition due to given order and due to AlgoMinWalk are shown with percentage of improvement. For algorithm AlgMinHam the test application time is not reported because it remains unchanged. For algorithm AlgoMinWalk and algorithm AlgoMinEWalk the test application time is reported in Table 3.2 and Table 3.3 in column named "testtime overhead".

3.4.1 Analysis of Experimental Results

The experimental results shown in Table 3.1 indicates that algorithm AlgoMinHam can achieve upto 27% reduction in peak power and 16% reduction in peak power at average in compared to given order. It is important to note that it also reduces average power upto 4.45%. The algorithm AlgoMinHam was aborted for s838, and b22 due to its NP-completeness. However, the algorithm AlgoMinWalk was used for these circuit to obtain the results without any test time penalty as it is polynomial time algorithm.

Algorithm AlgoMinWalk achieved about 8.33% improvement over algorithm AlgoMinHam while incurring nominal test-time overhead of 1.33%. Algorithm AlgoMinWalk was not able to reduce the peak power of other circuits below the limit achieved by algorithm AlgoMinWalk because either algorithm AlgoMinHam itself achieved the lower bound or the circuit did not have any cycle that can make walk different than Hamiltonian path. *However, for those benchmarks which achieved lower bound in algorithm AlgoMinHam, algorithm AlgoMinWalk was able to achieve the same result without increase in test time (without revisiting the nodes) quickly in comparison to algorithm AlgoMinHam.* For rest of the benchmarks (which are not mentioned in Table 3.2) reported in Table 3.1, algorithm AlgoMinWalk also achieved the same results (in terms of peak and average power reduction without test time overhead) as obtained by algorithm AlgoMinHam and are reported in Table 3.1.

For benchmarks b19 and b22, algorithm AlgoMinEWalk has marginal improvement over algorithm AlgoMinHam and algorithm AlgoMinWalk. It could not further reduce the peak power for the other benchmarks which are not reported in Table 3.3, because they already achieved the lower bound by algorithm AlgoMinHam or algorithm AlgoMinWalk. It is important to note that algorithm AlgoMinWalk and algorithm AlgoMinEWalk are polynomial time complexity algorithms. Therefore, these algorithm can be used for large industrial circuits.

Table 3.1: Experimental results for algorithm AlgoMinHam

Benchmark	#Testpattern	Scanlength	Peaktrans RandomOrder	Peaktrans AlgoMinHam	%Improvement	Avgtrans RandomOrder	Avgtrans AlgoMinHam	%Improvement
s9234	105	211	68	56	17.64	27.02	26.37	2.40
s38584	110	1426	444	379	14.63	120.99	119.45	1.24
s15850	95	534	120	96	20.00	37.09	36.42	1.80
s35932	12	1728	106	89	16.03	34.61	33.38	3.55
s38417	68	1636	473	342	27.69	161.35	158.84	1.55
s1196	136	18	17	14	17.64	9.21	8.80	4.45
b04	43	66	49	44	10.20	29.17	28.58	2.03
b07	52	45	34	31	08.82	19.37	18.95	2.17
b10	47	17	16	13	18.75	8.26	7.85	4.96
b14	627	215	163	150	7.97	98.10	98.00	0.11
b19	628	215	171	162	5.26	98.12	97.96	0.17
b21_1	488	215	144	129	10.41	99.34	99.20	0.15

Table 3.2: Experimental results for algorithm AlgoMinWalk

Benchmark	#Test pattern	Scan length	Peaktran RandOrder	Peaktran AlgoMinWalk	%Improvement	%Improve over AlgoMinHam	Testtime overhead	Avgtran RandOrd	Avgtran AlgoMinWalk	%Improvement
s838	75	32	16	12	25.00	0.00	0.00	4.61	4.59	0.43
s838	75	32	16	11	31.25	08.33	1.33	4.61	4.57	0.80
b19	628	215	171	162	5.26	0.00	0.00	98.12	97.96	0.17
b22	622	215	172	158	6.97	0.00	0.00	98.23	98.07	0.16

3.5 Concluding Remarks

This Chapter presented a graph theoretic approach to formulate the problem of test vector ordering for peak power minimization during scan testing, which is a concern for today's complex scan based design. This Chapter has presented an approach that can achieve minimum peak power without increase in test application time. It also presented two approaches that can further reduce the peak power on the cost of marginal increase in test application time. Finally, a theorem on lower bound on the minimum achievable peak power for a given test set is presented. The results indicate that the proposed approach can achieve upto 27% of peak power reduction without increase in test time. The proposed approach also reduces average power upto 4.96%. The algorithm AlgoMinWalk and algorithm AlgoMinEWalk are polynomial complexity algorithms, hence the methodology is scalable. In the future, algorithm AlgoMinWalk can be implemented as an approximation algorithm to reduce the time complexity. Note that the proposed method can be applied with the existing X-fill based approaches because it uses fully specified vectors. Hence, it can achieve more power saving in combination with the

Table 3.3: Experimental results for algorithm AlgoMinEWalk

Bench mark	#Test pattern	Scan length	Peaktran RandOrder	Peaktran AlgoMinEWalk	%Improve ment	%Improve over AlgoMinWalk	Testtime overhead	Avgtran RandOrd	Avgtran AlgoMinEWalk	%Improve ment
b19	628	215	171	161	5.84	0.61	0.63	98.12	97.50	0.63
b22	622	215	172	157	8.72	0.63	0.64	98.23	97.61	0.63

existing techniques.

Chapter 4

Minimization of Peak Power During Test Cycle

4.1 Introduction and Motivation

Scan circuit is widely practiced DFT technology. The scan testing procedure consist of state initialization, test application, response capture, and response comparison process. During the state initialization process the scan vectors are shifted in to the scan cells and simultaneously the responses captured in last cycle are shifted out. During this shift operation the transitions that arise in the scan cells are propagated to the combinational circuit, which inturn creates many more toggling activities in the combinational block and hence increases the dynamic power consumption. The dynamic power consumed during scan shift operation is much more higher than that of normal mode operation [55].

Due to change in design characteristics the dynamic power dissipated during scan operation becomes an important issue [25]. The average and peak power are the standard metrics for dynamic power. During scan testing it is very much important to keep both the average and peak power under certain threshold limit to assure the safe and reliable testing of chip. Average power causes excessive heat dissipation where as peak power causes IR drop and cross talk problem [19]. Particularly, the excessive peak power during

test-cycle (the dark gray period in Figure 4.2) of at-speed testing is vulnerable. The excessive peak power causes high rate of current in the power and ground rails which decreases the supply voltage and causes ground bounce, this phenomenon is known as IR-drop. The larger IR-drop means the worse speed performance of circuit. This degradation in performance grows if circuit is operated at high frequency which is the case during at-speed test [43]. This degradation in performance leads to incorrect capture of responses and this results in to undesired yield loss.

Hence, to avoid yield loss the peak power minimization is necessary specially in case of narrow *test-cycle*. More over the minimization of peak power is also advantageous for parallel testing of multiple cores to reduce test time.

This Chapter presents a methodology to minimize peak power consumption during *test-cycle* for at-speed test. The methodology proposed is based on scan cells reordering. The overall methodology consists of graph theoretic formulation of problem and a greedy based linear time heuristic to solve the problem.

The rest of this Chapter is organized as follows. Section 4.2 provides background on scan reordering and test power consumption. Section 4.3 presents the problem formulation. Proposed heuristic is described in Section 4.4. Time and space complexities for proposed algorithm is explained in Section 4.5. Section 4.6 presents the experimental results and Section 4.7 concludes the Chapter.

4.2 Background

4.2.1 Scan Operation and Test Cycle Peak Power

The scan operation in general consists of four basic processes: scan initialization, test launch, response capture, and response comparison. The initialization process initializes the scan chain with known data called as test pattern which are shifted in to the scan chain in serial shift manner. In the process of test launch the shifted in patterns are applied to the combinational block and the responses are captured next during the response capture process. The captured responses are shifted out while shifting in another

Clock	Scan chain				Operation
	$\longrightarrow$SF$_1$	$\longrightarrow$SF$_2$	$\longrightarrow$SF$_3$	$\longrightarrow$SF$_4$ $\longrightarrow$	
–	1	1	0	1	last response
C_1	0	1	1	0	shift
C_2	1	0	1	1	shift
C_3	0	1	0	1	shift
C_4	1	0	1	0	shift/launch

Figure 4.1: Scan operation

pattern. The shifted out responses are now compared with the golden responses to see whether a chip is faulty or faultfree. The golden responses are the predefined correct responses determined by the simulation process. Figure 4.1 demonstrates the standard scan procedure. The bits colored in light gray are responses of last scan vector and the bits without color and dark colored bits are current scan vector. In every scan clock each bits are shifted towards scan out. In the last shift cycle the response bits are completely shifted out and the current scan in bits are launched. Once the bits are launched, next a capture pulse is applied at system frequency for at-speed capture of responses. togglings in the combinational block cause invalid response capture.

Figure 4.2 illustrates the timing of at-speed scan test. The timing diagram shown is of skewed-load testing. In this figure L_p and C_p are launch and capture pulse, TCl and SEn are test clock and scan enable respectively. The dark gray area in Figure 4.2 shows the vulnerable period known as *test-cycle*. The excessive power dissipated during the *test-cycle* is particularly important to control for at-speed testing to assure the correct capture of responses. The excessive peak power during any moment in scan operation can leads to IR-drop and ground bounce problem. The problem of IR-drop may leads to the incorrect capture of responses during *test-cycle* which inturn causes a good chip to fail the test leads to yield loss.

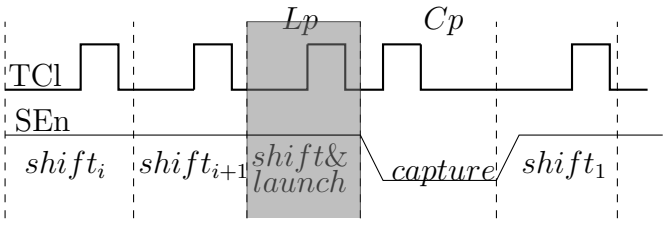


Figure 4.2: Timing digram for at-speed skewed-load scan testing

4.2.2 Power Estimation: Considering only Tests

As the circuits size is growing with the complexity of design the estimation of power by exhaustive simulation becomes a time consuming process. Hence the alternative power estimation models like test data analysis [45] and weighted transition metric [39] methods are more preferable. These kind of power estimation methods are based on toggle count in the scan cells. In this proposed methodology the power (*test-cycle* peak power) is estimated based on the toggle counts in scan cells. The proposed methodlogy is focused on the *test-cycle* peak power. The power that is dissipated during the *test-cycle* is contributed by the togglings due to shifting of current scan vector and responses of previous scan vector. However, during the test-cycle only a response bit of previous scan vector will remain in the scan chain as the test-cycle is the last shift cycle. In Figure 4.1 the dark gray area (clock C_3 and C_4) shows the transitions caused by the scan vector and the last response bit. Hence, for approximate power estimation only scan vectors would be sufficient. However, the exact estimation needs to consider the last bit of response as well. With this assumption we have formulated the problem considering only scan vectors.

4.2.3 Power Reduction by Scan Chain Reordering

The scan chain reordering mechanism used for peak and average power minimization is basically rely on reduction of *intra pattern transitions*. *Intra pattern transions* are transitions caused due to the difference between any consecutive bits in a pattern. For example, in Figure 4.3 the pattern P1 in default scan chain ordering have three intra

pattern transitions. By looking at the differing bits in a pattern, each scan cell can be reconnected in a suitable order to reduce the *intra pattern transitions*. The reordering of scan cells will bring all 1s close to each other as well as all 0s while performing the scan shift operation. For example, in Figure 4.3 the pattern P1 after reordering of scan chain have only one transition between bits in SF_3 and SF_4 and similarly the pattern P2 has only one transition.

Although the scan chain reordering is one of the efficient methodology to reduce the test power, it does have some inherent limitations in terms of other parameters like routing area and delay fault coverage. The power efficient scan reordering may result into routing area overhead and may also alter delay fault coverage. The routing area is affected if it will not be considered as a constraint during the process of scan chain reordering. However, the routing area problem can also be solved by considering the routing area as one of the parameter along with the power. The work by Girard et al. [20] deals particularly with this problem where the tradeoff between average power and routing area is fixed.

The scan chain reordering may also alter the delay fault coverage for skewed-load testing. In skewed-load testing two test vector $\langle V_1, V_2 \rangle$ are applied in consecutive cycles for scan initialization and transition launch respectively [1]. The vector V_2 is single shift of vector V_1. In normal case the vector V_1 will initialize the scan chain properly, however, once the scan chain is reordered the reordered version of vector V_1 has to be applied. The reordered version of vector can not be in proper position to the corresponding scan cells at the time of scan initialization and this makes the scan initialization improper and hence leads to alteration in fault coverage.

[1] refer Section 1.1.2

	Default scan chain ordering				Reordered scan chain			
	$SF_1 \rightarrow SF_2 \rightarrow SF_3 \rightarrow SF_4$				$SF_1 \rightarrow SF_3 \rightarrow SF_4 \rightarrow SF_2$			
P1	1	0	1	0	1	1	0	0
P2	1	0	1	1	1	1	1	0

Figure 4.3: Basic principle of scan chain reordering

4.3 Construction of Graph and Problem Formulation

Objective: To find an optimal ordering of scan cells which minimizes the peak power consumption during *test-cycle*.

Problem formulation consist of two steps. In first step a complete vector-weighted graph is constructed with respect to scan chain. In second step a TSP (travelling salesman problem) like problem, Vector-TSP, is formulated on the constructed graph.

4.3.1 Construction of Vector Weighted Graph

The important information that are needed to be reflected in the final graph are, scan cells, possible scan paths among cells and the peak power information. Following is the procedure for vector-weighted graph construction:-

- For every scan cell SF_i there is corresponding node N_i in graph.
- The scan path between any two possible scan cells is represented by an undirected edge between respective nodes.

  Note: The edges are undirected because, the pairing of scan cells is symmetric with respect to the number of transition. For example, in Figure 4.3 a reverse ordering of the reordered scan chain will also results into one transition only.
- The instantaneous power consumed due to any possible pair of scan cells for all scan

vectors is represented as a weight of the edge between respective nodes of the scan cells. The edge weight is a vector quantity, we name this weight as *vector-weight*. The *vector-weight* is computed as follows:

Computation of vector-weight: Each *vector-weight* keeps the instantaneous power information for all scan test vectors. The computation process does not consider the scan responses because during the last shift operation only one response bit will remain in the scan chain [2]. The instantaneous power information for an individual scan vector on particular scan cell pair is either 1 or 0 based on the bit difference at original scan cells position.

Table 4.2 shows the *vector-weights* for scan test pattern set given in Table 4.1. The rows in Table 4.1 are scan test patterns T_i and columns are scan flops SF_i. In Table 4.2 the scan cell pairs are given in column labeled with "Scan cell pairs" and corresponding *vector-weights* are given in columns t_1 to t_6. The columns t_1 to t_6 indicate the peak power information for each scan test pattern. The rectangular box in Table 4.1 shows the bit difference due to scan pair SF_1 and SF_2 and the corresponding power consumption information in Table 4.2 in the form of transition is inscribed in a square box in t_1 column. Figure 4.4 shows the complete vector-weighted graph constructed from the data given in Table 4.1 & Table 4.2.

Table 4.1: Scan patterns and original scan order

	$\rightarrow SF_1 \rightarrow$	$SF_2 \rightarrow$	$SF_3 \rightarrow$	$SF_4 \rightarrow$
T_1	1	0	1	0
T_2	0	1	0	1
T_3	1	0	1	0
T_4	1	0	1	1
T_5	0	1	0	1
T_6	1	0	0	0

[2] refer Figure 4.1

Table 4.2: Computed vector-weight

Scan cell pair	Vector-weight					
[SF_i , SF_j]	t_1	t_2	t_3	t_4	t_5	t_6
[SF_1 , SF_2]	1	1	1	1	1	1
[SF_2 , SF_3]	1	1	1	1	1	0
[SF_3 , SF_4]	1	1	1	0	1	0
[SF_4 , SF_1]	1	1	1	0	1	1
[SF_1 , SF_3]	0	0	0	0	0	1
[SF_4 , SF_2]	0	0	0	1	0	0

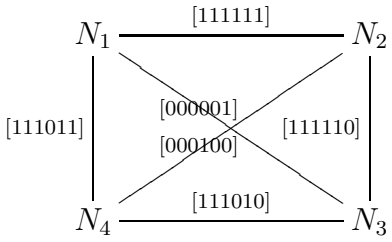


Figure 4.4: A complete vector-weighted graph

4.3.2 Formulation of Vector TSP problem

This section will explain the formulation of TSP like problem called as Vector TSP on complete vector-weighted graph.

The instantaneous power consumed by scan pattern during *test-cycle* due to scan cell pair [SF_i, SF_j] is represented as the *vector-weight* of edge, edge(N_i, N_j), in the graph. For example, the instantaneous power consumed by scan pair [SF_4, SF_2] is represented as *vector-weight* ([0 0 0], [1 0 0]) shown in Table 4.2. As the graph is complete graph it is always possible to find an order of scan cells by constructing a *Hamiltonian path* (which visits each node of the graph). Each possible Hamiltonian path in the graph has a corresponding cost, *cost(N_s, N_e)*, of traversing the path N_s—∼— N_e, where N_s and N_e are start and end nodes of path. The cost function *cost(N_s, N_e)* is computed in the following way,

- Sum up all the *vector-weights* of path N_s—∼— N_e in vector addition manner
- Find maximum value on summation result. This maximum value corresponds to the peak power consumed by scan order SF_s—∼— SF_e

In more formal way, the peak power can be expressed as,

$$PeakPower = cost(N_s, N_e) \tag{4.1}$$

where, $cost(N_s,\ N_e) = Max\left(\sum_{i,j\epsilon(1,n)} vector\text{-}weight(N_i, N_j)\right)$, where N_i, N_j are adjacent nodes in path N_s—∼— N_e, $(1, n)$ is set of integer from 1 to n, and n is length of scan chain.

Following example illustrates the above described procedure.

Example 4.1. Let $N_1 \overset{[111011]}{\text{———}} N_4 \overset{[000100]}{\text{———}} N_2 \overset{[111110]}{\text{———}} N_3$ be a Hamiltonian path from Figure 4.4. Vector summation of the *vector-weights* for this path = *vector-weight*(N_1, N_4) + *vector-weight*(N_4, N_2) + *vector-weight*(N_2, N_3)
= [1 1 1 0 1 1] + [0 0 0 1 0 0] + [1 1 1 1 1 0] = [2 2 2 2 2 1]
The cost(N_1, N_3) = Max[2 2 2 2 2 1] = 2, which is the peak power value for corresponding scan chain order. call this cost of tour as *path-weight* with respect to given path.

Hence from above illustration the graph theoretic problem can formally be stated in following way:
Problem Statement: Given an undirected graph G(V, E, W), find a Hamiltonian path which has minimum cost over all other possible paths. Where V is set of vetices V_is, E is set of edges, and W is set of *vector-weights*.
Note: The hardness of problem is NP-complete as this is reducible to TSP which is a known NP-complete problem.

4.4 Proposed Algorithm

As the problem is NP-complete we are propose a greedy based polynomial time (with respect to |V|) algorithm. The complete algorithm consist of two parts, Part-1, a greedy heuristic to find the Hamiltonian cycle and Part-2, a heavy-edge removal algorithm to find the path from the Hamiltonian cycle. Following two subsections explain algorithm Part-1 and Part-2.

4.4.1 Greedy Heuristic to Construct Hamiltonian Cycle

Problem Statement: Given a graph G(V, E, W), find a Hamiltonian cycle having minimum cost, $cost(N_s, N_e)$, here $s = e$ because for a cycle start and end node are same. *path-weight* in path.

Algorithm Part-1:

Step 1: Chose an initial node as N_1 and assign it to *current-node* and initialize corresponding *cumulative-node-cost* with [0 0 0 0 0 0] to start with.

Note: cumulative-node-cost is an array of integer values computed dynamically for a node visited currently.

Step 2: In this step the decision parameter *peak-power* and *average-power* are computed which are used in Step 3 to select a next node to visit. The *peak-power* can be called as primary parameter and the average-power as secondary parameter. The *peak-power* and *average-power* are computed with respect to the set of node which are not yet visited and are neighbor of current node, we call these node as *candidate-node* and any edge from current node to *candidate-node* is called as *candidate-edge*. A vector addition operation is performed between the *cumulative-node-cost* and *vector-weight* of each *candidate-edge* separately, we call these sum as *probe-path-weight*. The *peak-power* for each of the *candidate-edge* is computed by finding Max(*probe-path-weight* along the *candidate-edge*) respectively. And *average-power* is computed by finding Sum(*probe-path-weight* along the *candidate-edge*) respectively.

Step 3: A node N_i from the set of *candidate-node* is selected as the next node if the *peak-power* along the edge(*current-node*, N_i) is minimum among all other *peak-power*. If there are more than one node having equal *peak-power* then corresponding *average-power* is used as a tie breaker. The following example helps understanding the above explanation.

Example 4.2. Considering Figure 4.4, the initial node will be N_1 with *cumulative-node-cost* [0 0 0 0 0 0]. The neighboring nodes of N_1 are N_2, N_3, and N_4 which satisfy the property of being *candidate-node* because these are not yet visited. Based on the *peak-power* and *average-power*, a next node will be selected out of these *candidate-node*. The corresponding *candidate-edge* are edge(N_1, N_2), edge(N_1, N_3), and edge(N_1, N_4). The *vector-weight* of each of these *candidate-edge* are [1 1 1 1 1 1], [0 0 0 0 0 1], and [1 1 1 0 1 1] respectively. *Probe-path-weight* along each of the *candidate-edge* will be vector addition of *cumulative-node-cost* of current node with the *vector-weight* of each of these *candidate-edge* which are,

[0 0 0 0 0 0] + [1 1 1 1 1 1] = [1 1 1 1 1 1],

[0 0 0 0 0 0] + [0 0 0 0 0 1] = [0 0 0 0 0 1], and

[0 0 0 0 0 0] + [1 1 1 0 1 1] = [1 1 1 0 1 1] respectively.

The *peak-power* and *average-power* along each *candidate-edge* can now be computed as follows,

peak-power:

along edge, edge(N_1, N_2), will be $Max([1\ 1\ 1\ 1\ 1\ 1]) = 1$,

along edge, edge(N_1, N_3), will be $Max([0\ 0\ 0\ 0\ 0\ 1]) = 1$, and

along edge, edge(N_1, N_4), will be $Max([1\ 1\ 1\ 0\ 1\ 1]) = 1$.

average-power:

along edge, edge(N_1, N_2), will be $Sum([1\ 1\ 1\ 1\ 1\ 1]) = 6$,

along edge, edge(N_1, N_3), will be $Sum([0\ 0\ 0\ 0\ 0\ 1]) = 1$, and

along edge, edge(N_1, N_4), will be $Sum([1\ 1\ 1\ 0\ 1\ 1]) = 5$.

The next task is to select a next node based on the above computed *peak-power* and *average-power*. In this example *peak-power* along each *candidate edge* are equal *i.e.* 1,

hence *average-power* is used as a tie breaker. *Average-power* along edge, edge(N_1, N_3,), is 1 which is less than the *average-power* of other *candidate edge*. Hence node N_3 will be selected as a next node to be visited.

Step 4: In this step *cumulative-node-cost* and *current-node* are updated. The *cumulative-node-cost* for next node to visit = *cumulative-node-cost* of *current-node* + *vector-weight* of edge(*current-node*, *next-node-to-visit*). In Example 4.2 *current-node* will be updated with node N_3 and *cumulative-node-cost* for N_3 = [0 0 0 0 0 0] + [0 0 0 0 0 1] = [0 0 0 0 0 1]

Step 5: Repeat the algorithm from Step 2 till the time a Hamiltonian cycle is discovered.

The output of algorithm Part-1 will be a Hamiltonian cycle. For graph shown in Figure 4.4 algorithm Part-1 will give Hamiltonian cycle shown in Figure 4.5.

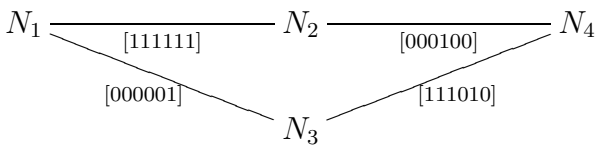


Figure 4.5: Hamiltonian cycle: output of algorithm Part-1

4.4.2 Heavy-edge Removal Algorithm to Construct Hamiltonian Path

Problem Statement: Given a Hamiltonian cycle H(V, E, W), find a Hamiltonian path which have minimum cost, *cost(N_s,N_e)*. In other way it can also be stated as, remove an edge from H such that the resultant path will have minimum possible peak power consumption.

Algorithm Part-2:

Step 1: Computation of *remainder-peak* and *remainder-average* power:

- Sum-up all *vector-weights* in vector addition manner
- Subtract each *vector-weight* from the summation result

- Compute *remainder-peak* and *remainder-average* for each possible path obtained after removal of each edge

Following example illustrates the above procedure.

Example 4.3. In this example Figure 4.5 will be used as a reference.

Summation of all the *vector-weight*s will be:

[1 1 1 1 1 1] + [0 0 0 1 0 0] + [1 1 1 0 1 0] + [0 0 0 0 0 1] = [2 2 2 2 2 2]

Computation of *remainder-peak* power and *remainder-average* power corresponding to each edge is done as follows.

For edge(N_1, N_2), the subtraction result = [2 2 2 2 2 2] - [1 1 1 1 1 1] = [1 1 1 1 1 1],

remainder-peak = Max([1 1 1 1 1 1]) = 1 and

remainder-average = Sum([1 1 1 1 1 1]) = 6.

For edge(N_2, N_4), the subtraction result = [2 2 2 2 2 2] - [0 0 0 1 0 0] = [2 2 2 1 2 2],

remainder-peak = Max([2 2 2 1 2 2]) = 2 and

remainder-average = Sum([2 2 2 1 2 2]) = 11.

For edge(N_4, N_3), the subtraction result = [2 2 2 2 2 2] - [1 1 1 0 1 0] = [1 1 1 2 1 2],

remainder-peak = Max([1 1 1 2 1 2]) = 2 and

remainder-average = Sum([1 1 1 2 1 2]) = 8.

For edge(N_3, N_1), the subtraction result = [2 2 2 2 2 2] - [0 0 0 0 0 1] = [2 2 2 2 2 1],

remainder-peak = Max([2 2 2 2 2 1]) = 2 and

remainder-average = Sum([2 2 2 2 2 1]) = 11.

Step 2: Selection of an optimal path based on the following criteria:

- The path having minimum *remainder-peak* will be the resultant path
- If more than one paths have equal *remainder-peak* then *remainder-average* will be used as tie breaker
- If all paths are having equal *remainder-peak* and *remainder-average* then any path based on random choice will be the resultant path

Based on the criteria given in Step 2, it can be observe in Example 4.3 that removal of edge(N_2, N_1) is resulting into *remainder-peak* = 1 (which is minimum among all other *remainder-peak* power) and *remainder-average* = 6. Hence removal of an edge, $edge(N_2, N_1)$ will leads to an optimal path $N_1 \rightarrow N_3 \rightarrow N_4 \rightarrow N_2$. The corresponding minimized peak power scan path will be $SF_1 \rightarrow SF_3 \rightarrow SF_4 \rightarrow SF_2$.

4.5 Analysis of Time and Space Complexity

4.5.1 Time Complexity

Heuristic to Construct Hamiltonian Cycle: This part of the algorithm will take $O(n * l^2)$ time to run, where l is length of scan chain and n is number of scan vectors. Total l number of nodes are visited in l iteration. In each iteration except first and last the peak power and average power are computed for each unvisited nodes. The peak and average power computation for each node will take $n + n = 2n$ time. In second iteration, the total number of unvisited nodes are $l - 1$, in third iteration it is $l - 2$, and in subsequent iteration the total number of nodes will decrease by one. Hence total time consumed in each iteration will be $2n * (l - i)$, where $i = 2$ to $l - 2$ is iteration number. Hence, the final expression will be $2n * \sum_{i=1}^{l-2}(l-i)$ which can be bound by $O(n * l^2)$.

Heavy-edge Removal Algorithm: This part of algorithm will run in $O(n * l)$ time. Total of $n * l$ time will be consumed in summing-up all the *vector-weights*. Each subtraction operation will require n time. Computation of *remainder-peak* and *remainder-average* will take $n + n = 2n$ time. Hence, the total time can be bound by $O(n * l)$.

4.5.2 Space Complexity

Overall space needed for running the algorithm is $\Theta(n + l)$. An array of size l is needed to keep the visited information for each node. Another array of size n is needed for *cumulative-node-cost*. A temporary buffer of size n is needed to keep *vector-weight*.

Hence, total space can be bounded by $\Theta(n + l)$.

Apart from the required arrays the algorithm also uses the external storage to keep the test patterns and responses. The size of these file depends on the pattern volume.

Note: Proposed algorithm does not use any explicit space to keep *vector-weights* for every edge rather these are computed dynamically from the scan pattern file.

Hence, the proposed heuristic does not suffer from any kind of strain related to time and space.

4.6 Experimental Result

Experiments are carried out on ITC99 [11] and ISCAS89 [6] circuits. The algorithm is implemented in C++. Scan insertion and circuit synthesis are performed using DFT Compiler [47] of Synopsys. Test patterns are generated using TetraMAX [48] ATPG tools of Synopsys. Scan patterns are low power adjacent filled and compacted at low merge level compaction. Specification of benchmark circuits and experimental results are provided in Table 4.3 and 4.4 respectively. For each benchmark circuit, Table 4.4 shows the number of transition for peak power due to industrial solution and due to proposed methodology. The fourth column of the Table 4.4 shows the %of peak power reduction achieved by [3] (the data shown are taken from [3]) and the last column shows the %of reduction achieved by proposed methodology.

From Table 4.4 the improvement by proposed work over [3] can be observed. At average the proposed work is able to reduce 22.43% of test-cycle peak power where as the referred work is able to rducing 17.83%. More over for the benchmark b09 the proposed methodlogy is providing the maximum reduction of 55%. For most of the benchmark circuits the proposed methodology is able to reduce more than 30% of peak power.

Table 4.3: Specification of benchmark circuits

Benchmark circuit	#Test pattern	Scan chain length	#PI	#PO	#Gates	Stuck-at fault coverage
b04	49	66	11	8	628	92.85
b07	53	45	1	8	427	99.95
b08	40	21	9	4	171	100
b09	27	28	1	1	160	100
b10	44	17	11	6	180	100
s420	53	16	18	1	218	99.32
s526	27	21	35	2	158	99.32
s1196	137	18	14	14	381	100
s5378	107	179	35	49	2779	99.87
s9234	123	211	36	39	5597	99.93
s13207	115	638	62	152	7951	99.97
s15850	96	534	77	150	9772	100
s35932	13	1728	35	320	16065	100
s38417	69	1636	28	106	22179	100
s35854	111	1426	38	304	19523	100

4.7 Concluding Remarks

This Chapter has presented a graph theoretic approach for reordering the scan cells to minimize the peak power during *test-cycle*. The importants of reducing the peak power during *test-cycle* is also discussed. The Chapter has presented the procedure of graph theoretic problem formulation and also presented a heuristic algorithm to solve the problem. The methodology proposed is a novel idea, as it opens up a new way of formulating a graph theoretic problem for scan reordering, which has broaden the solution space for scan reordering problem. The experimental results show that the proposed methodlogy is capable of reducing appreciable amount of peak power compared to the earlier methodlogy.

As it is discussed in Section 4.2.3 that the scan reordering methodology may suffer from two drawbacks, scan routing area overhead and alteration of transition fault coverage. In this proposed methodology we have not taken these drawbacks into consideration. However, the routing area overhead can be solved by integrating the proposed

Table 4.4: Experimental results

Benchmark circuit	#Peak trans. IndustrialSol	#Peak trans. ProposedSol	%of Reduct.[3]	%of Reduction
b04	32	16	18.2	50
b07	27	20	NA	25.92
b08	15	8	NA	46.66
b09	18	8	51.22	55.55
b10	12	9	12.8	25
s420	9	6	26.3	33.33
s526	13	10	22.4	23.07
s1196	7	6	19.0	14.28
s5378	77	58	21.2	24.67
s9234	106	72	18.8	32.07
s13207	266	166	16.9	37.59
s38417	381	368	24.9	3.41
Avg. Data	80.25	62.25	17.83	22.43

graph theoretic methodology with the methodology proposed in [20], where the proposed methodology can be applied within the cluster to take care of power and clustering ordering can be used to take care of routing congestion. Another draw back, alteration of transition fault coverage in skewed-load testing, needs further examination.

Chapter 5

Capture Power Reduction for Modular SoC Test

5.1 Introduction and Motivation

Modular-design approach becomes common now a days in order to design ICs in a timely manner under time to market pressure. As manufacturing is far from perfect, all ICs must be tested. An IC designed in a modular way can be tested in a modular fashion. The most common methodology to test a core is scan based test. Scan based tests may cause circuit switching activity in excess of the activity during normal operation of the circuit. Excess *peak power* consumption demands higher peak current, which may cause supply voltage droop, results into increase in gate delay during test. Increase in gate delay can cause good chips to fail at-speed tests, leading to *yield loss*.

Excessive switching activity during the application of scan tests occur in FFs and combinational logic during scan chain shifts to load tests and unload test responses as well as when the scan cell contents are updated using functional clocks. Therefore, the test power consumption can be divided into power consumed during the shift process and power consumed during capture mode. Power consumption in the combinational logic while test vectors are being shifted in the scan chains is unwanted and useless, hence it can be completely eliminated by gated FF output. Transitions in FFs while shifting are

unwanted but these are needed to load test vectors in scan chains, hence, these cannot be completely eliminated but such transitions can be reduced by test vector reordering [22, 49] and other approaches like scan chain reordering [20, 50, 51] and MT-fill[14]. The power dissipation during the test application (capture cycle) is needed, hence we can neither eliminate it nor reduce it. Therefore, cores which are scheduled together must dissipate the power during the capture cycle in order to generate test response. This power some times goes far beyond the normal power dissipation of the SoC because neither the test vector generation nor test scheduling care about the functionality of the SoC. *It becomes worse when capture cycles of multiple cores which are scheduled together coincide*. The location of the cores makes it more severe. This Chapter addresses the above stated issue - *reduction of peak power by reducing capture power when capture cycles of various cores coincide*, as it is likely to be more severe in the years to come when SoCs are being designed with large number of cores. The following example explains the problem of capture cycle coincidence for two simultaneously tested cores.

Example 5.1. Let's assume we have two cores which are being tested simultaneously. Let Core 1 has a scan chain of length 4 and Core 2 has a scan chain of length 5. Therefore, Core 1 needs 4 cycles to scan in a test vector and it captures test response in cycles 5; and capture cycle repeats after every 5 cycle [shown in Fig. 5.1(a)]. Similarly, Core 2 scans in a test vector in 5^{th} scan cycle and captures the response in cycle 6; and the capture cycle repeats after every 6 cycles [shown in Fig. 5.1(b)]. We assume that there is no power dissipation in combinational logic during the scan shift operation. Both cores, when scheduled together to test, will have simultaneous capture at cycles 30, 60, 90, ... (which are multiple of least common multiplier of 5 and 6) etc. It may results into power droop problem that causes chip to falsely fail the test, if the sum of capture power of Core 1 and Core 2 exceeds a threshold for the following vectors: 1. 6^{th} vector of Core 1 and 5^{th} vector of Core 2, 2. 12^{th} vector of Core 1 and 10^{th} vector of Core 2, and so on as illustrated in Fig 5.1(c). Therefore, it is important to care about capture power when capture cycle of various cores scheduled together coincides. It may result into pessimistic schedule as per existing scheduling algorithm if it is not treated separately.

Note that we use gated FF output which reduces shift power significantly; hence, we can assume shift power negligible in comparison to capture power. Moreover, scan shift clock is much slower in comparison to system clock.

To solve the problem described above, in this work two solutions has been proposed:

1. Test vector reordering for capture power reduction for a given schedule
2. Insertion of idle cycle to prevent capture cycle to coincide for a given schedule

The rest of the Chapter is organized as follows. The proposed techniques to handle capture power are described in Section 5.2. The experimental results are presented and analyzed in Section 5.3. The Chapter is concluded with concluding remarks in Section 5.4.

5.2 Proposed Technique

In this section the problem is formulated and two solutions to solve the problem are presented.

5.2.1 Problem Formulation

Capture power appears as a challenge for SoC testing, as mentioned in Section 5.1. Section 2.4 highlighted the shortcomings of the existing methodologies to deal with the capture power. The existing solutions are pessimistic if capture power is taken into account. Hence, a new methodology is proposed particularly to solve the capture power problem. The following definitions will be used in rest of the sections.

- P_{sf} is the power dissipated in FFs during shift operation
- P_{sl} is the power dissipated in logic during shift operation
- P_{cf} is the power dissipated in FFs during capture
- P_{cl} is the power dissipated in logic during capture

- $L^j_{in_i}$ is the length of input wrapper chain of j^{th} core for i^{th} TAM wire
- $L^j_{out_i}$ is the length of output wrapper chain of j^{th} core for i^{th} TAM wire
- $L^j_{sc_i}$ is the length of scan chain of j^{th} core for i^{th} TAM wire
- tc^i_j is the j^{th} capture cycle of i^{th} core
- $P_{i,j}$ is the power dissipated in logic by core i during application of test vector j (capture power)

The total power dissipation in the circuit is

$$P_t = \begin{cases} P_{sf} + P_{sl} & (during\ shift) \\ P_{cf} + P_{cl} & (during\ capture) \end{cases}$$

In case of gated FF out put $P_{sl} = 0$. It has been observed that the capture power (number of transitions in the combinational logic) is proportional to the number of transitions in the FFs [45]. In case of unavailability of circuit information, which is the case for hard cores, combinational circuit power can be computed as constant times the number of transitions in the FFs. Hence the total power would be

$$P_t = \begin{cases} (1+\alpha)P_{sf} & (during\ shift) \\ (1+\alpha)P_{cf} & (during\ capture) \end{cases}$$

If we assume output is gated and scan shift power is very small in comparison to capture power, the total power would be P_t= (1+ α) P_{cf}.

The test access mechanism (TAM) transports test patterns from ATE to the core and test response from the cores to ATE. The wrapper connects the core's functional inputs and outputs to the TAM and to the rest of the SoC. Assume that k TAM wires are used for core j.

The first capture cycle of core j with k TAM wires occurs at clock cycle max { $\forall_k$ $(L^j_{in_k}+L^j_{sc_k})$} and subsequent capture cycles occur at {first capture cycle + $(n-1)$ * max { $\forall_k$ $(max\ (L^j_{in_k}\ ,\ L^j_{out_k})\ +L^j_{sc_k})$}, for $0 < n \leq$ no. of test vectors}. The period of

capture cycle would be max { $\forall_k$ (max ($L^j_{in_k}$, $L^j_{out_k}$) $+L^j_{sc_k}$). Hence, the n^{th} capture cycle of j^{th} core, tc^j_n would occur at

tc^j_n = { max { $\forall_k$ ($L^j_{in_k}$+$L^j_{sc_k}$)}+ $(n-1)$ * max {$\forall_k$ (max ($L^j_{in_k}$, $L^j_{out_k}$) $+L^j_{sc_k}$)}}, for $0 < n \leq$ no. of test vectors, where k is number of TAM wires.

The test vector p for Core i and test vector q for Core j will have a simultaneous capture cycle iff

$$tc^i_p = tc^j_q \tag{5.1}$$

The solution of the Equation 5.1 gives the values of p and q where the capture cycle coincide and it will be periodic, whose period will be the least common multiple of periods of two cores.

Therefore, multiple cores can have simultaneous capture cycle which can be obtained by Equation 5.1. Scheduling of these cores with power limit using global power model [9] and cycle-accurate model [38] would be too pessimistic. However, once we get a schedule considering P_{sl} and P_{cf}, the problem of capture power can be handled in better way.

5.2.2 Two Solutions: Test Vector Reordering and Idle Cycle Insertion

Problem Statement: Minimize the capture power during the scan testing of multi module SoC for a given test schedule.

This Chapter proposes two solutions for reduction of capture power for scheduled cores. Thus it assume that the test schedule is given.

Solution 1: *Reordering test vectors for a given schedule.*

The test vectors whose capture cycles coincide can be obtained from the Equation 5.1 given above. Now, identify those vectors whose power exceed the threshold limit, P^{th}. The identified test vectors can be swapped with the low activity test vectors. Hence, power profiling and sorting of the test vectors must be carried out before the process starts. In order to swap the test vectors three different methodologies can be used. 1.

Exhaustively search the best combination of the test vectors whose sum is below the threshold P^{th}. 2. Use of Integer Linear Programming (ILP). 3. Use of heuristic. In this proposal a heuristic is proposed for reordering of test vectors.

Heuristic to swap test vectors:

1. Obtain conflicting test patterns and conflicting capture cycles (when capture cycle coincide) using Equation 5.1 for all cores
2. Find the ratio of conflicting test patterns $RCTP_i$ for all cores
3. For every conflicting capture cycle do the following
   (a) For all conflicting cores till total power in conflict cycle is greater than P_{th} do the following
      i. Choose a core in the order of minimum ratio of conflicting patterns $RCTP_i$
      ii. Swap the conflicting pattern with the minimum power pattern

The proposed heuristic chooses a core, to swap test vectors, with minimum $RCTP$ because the core has less number of test vectors to swap and it will leave room for cores with higher $RCTP$ to swap during later cycles. The proposed heuristic can give a better solution for test vector re-ordering. It may results into marginal increase in switching activity during the shift operation if test vectors are already ordered but we believe capture power is always higher than the shift power; hence, the increase in shift power could be neglected.

Solutions 2: *Insertion of idle cycle to avoid capture cycle coincidence*

The conflicting capture cycle can be avoided by insertion of some idle cycles before the capture cycle for the conflicting vectors. The cores should be sorted in order of *slack* (slack is the remaining idle time after completion of test) and insert an idle cycle for a core which has the largest *slack* so that it may not lead to extra test cycles.

Lemma 1. *Insertion of one idle cycle is sufficient to make it completely conflict free if the capture periods of two cores are even number.*

Proof. The capture cycle of two cores i and j will coincide iff $tc_p^i = tc_q^j$. The equation can be written as $\{ tc_1^i + (p-1) *$ capture period of core $i \} = \{tc_q^j + (q-1) *$ capture period of core $j\}$. Since capture periods are even numbers, then the above stated equation holds good in case both tc_1^i and tc_1^i are either odd or even. It implies,
$\{tc_1^i + (p-1) *$ capture period of core $i \} \neq \{1 + tc_q^j + (q-1) *$ capture period of core $j\}$.
Hence, insertion of one cycle is sufficient to eliminate capture cycle conflict (coincidence) between cores i and j. □

The above stated Lemma could be quite useful to minimize cycle penalty.

Procedure for Idle cycle insertion:

1. Identify conflicting vectors using equation $tc_p^i = tc_q^j$, i.e value of p for core i and value of q for core j
2. Compute slack for all cores
3. For every conflicting capture cycle if total power is greater than P_{th} do the following
   (a) For every conflicting core in order of slack do the following
      i. Insert an idle cycle before conflicting test vector of largest slack core till total power is greater than P_{th}
   (b) Recompute and update the conflicting capture cycles and slack values

The above two solutions, test vector reordering and insertion of idle cycle, works for a given schedule. We assume a good schedule is already generated using P_{sf} and P_{cf}.

5.3 Experimental Results

In order to demonstrate the effectiveness of the proposed technique, we have carried out experiments on an SoC d695 [24]. For d695, we have used real test data and filled the Xs in the test stimuli according to MT-fill (minimum transition fill) and the test responses in the most pessimistic way (maximum transition fill) as responses cannot be controlled as test stimuli. The test pattern used for the experiment are obtained from Miyase and Kajihara [30]. The test scheduler developed by Samii et al.[38] is used to generate the initial test schedule. After generating the test schedule the test vector reordering and idle cycle insertion techniques were applied. We conducted experiments for different TAM wires and different power. Power profile of different test vector has been obtained and sorted the test vectors in ascending order of power consumed. The experimental results are shown in Table 5.1.

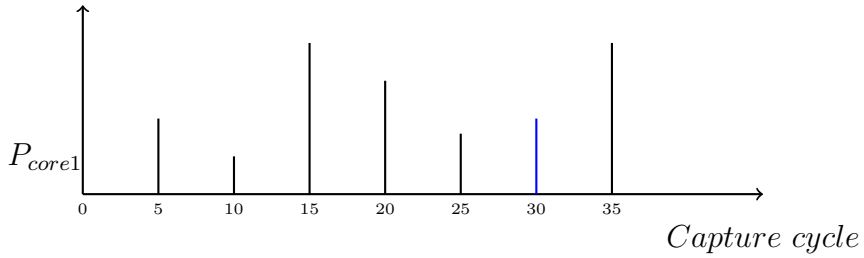
The column 1 of the Table 5.1 shows the number of TAM wires used for test scheduling, column 2 and 3 shows the minimum power (number of transitions) and test time obtained by Samii's [38] approach. Here we assume gated output structure and negligible shift power. Column 4 and 6 show power and test time by using test vector swapping (Solution 1) and column 5 shows the power saving. Similarly, column 7 and 9 show power and test time by using insertion of idle cycle (Solution 2) and column 8 shows the power saving. In this experiment test time did not increase by idle cycle insertion as this schedule had enough slack. The Table 5.1 clearly indicates that capture power can be further reduced by test vector swapping or idle cycle insertion. Here, we have compared our results with the minimum achievable test power of Samii et al. [38], i.e, the approach proposed by Samii et al. [38] cannot schedule for the power reported by our approach. The results show that we can reduce peak power by about 21% for SoC d695. It can be observed from the experimental results that the proposed methodology could not reduce power below 945 as these transitions come from a single vector of a core. Hence the power minimization can not go below this level. Therefore, it provides the lower bound.

Table 5.1: Results for SoC d695

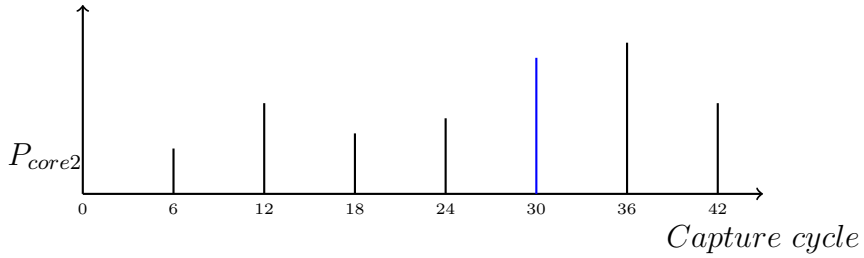
TAM	Soheil [2]		Solution 1			Solution 2		
	Power	TT	Power	% Save	TT	Power	% Save	TT
16	1004	43428	946	6	43428	946	6	43428
32	1040	24089	946	9	24089	946	9	24089
48	1130	18799	946	16	18799	946	16	18799
64	1210	13976	1018	16	13976	946	21	13976

5.4 Concluding Remarks

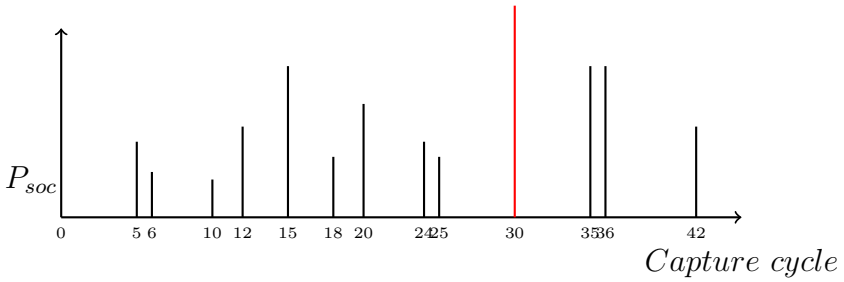
Power reduction during test has become an important issue in modern designs. Concurrent test of modular SoC helps in reduction of test application time whereas it may dissipate power greater than the power budget, specially during the capture cycle. The previously proposed approaches schedule the SoC test pessimistically due to the coincidence of capture cycles of more than cores. This Chapter has presented two approaches to reduce capture power: by vector reordering and idle cycle insertion techniques. Heuristics are used to select the vector pair for swapping and insertion of idle cycles. We presented results on an SoC d695, which are quite encouraging.



(a) Power profile of Core1



(b) Power profile of Core2



(c) Power profile of Core1 and Core2

Figure 5.1: Capture power profile

Chapter 6

Conclusions and Future Works

Excessive peak power consumption during scan testing is a majore concern for modern high density and high performance design [25]. The excessive peak power consumed during scan testing causes temperature and IR-drop related problem [43], [19]. This dissertation has proposed a set of solutions to reduce peak power during scan testing. Section 6.1 summarizes the contributions of this dissertation work and Section 6.2 provides some possible future work.

6.1 Contributions of Thesis

The dissertation has proposed a new set of methodologies to minimize peak power consumption during scan testing. Chapter 3 has proposed a new way of formulating a test vector ordering problem as graph theoretic problem for peak power minimization. Three graph problems are formulated. To solve the formulated problems three algorithms are proposed. Also the lower bound on minimum achievable peak power is defined based on the graph theoretic formulation. The experimental results show that all the proposed algorithms are effectively reducing the peak power. The first algorithm AlgoMinHam could able to reduce the peak power up to 27% without affecting the average power. Similarly the second algorithm AlgoMinWalk could able to reduc the peak power by 4.96% over the first algorithm. And the third algorithm AlgoMinEWalk could able to

reduce 0.63% over the second algorithm.

Chapter 4 has proposed a scan reordering methodology to reduce peak power during test-cycle. The test-cycle peak power is important to reduce because the excessive peak power in this cycle may affect the yield [43]. The chapter has contributed a new way of graph theoretic formulation for scan reordering to minimize the test-cycle peak power. The experimental results compared with the previous work show that the proposed methodology is performing better. The proposed work could able to achieve the reduction upto 55% compared to the industrial solution where as the referred methodology reduces 50% of peak power. The primary contributions of this approach are: it has broadened the solution space for test-cycle peak power problem and probably now using this approach the routing area and peak power tradeoff can be fixed easily.

Chapter 5 has proposed two methodologies to minimize capture power. The proposed technique minimizes the capture power below the limit the earlier proposed cycle-accurate model could achieve. The proposed technique enables the scheduling of multiple cores for parallel testing as now the peak power is minimized. The methodology proposed is software base, hence it doesnot incur any additional area. The test application time for an individual core also remain almost unchanged except for few cases.

6.2 Future Works

The dissertation work has also resulted in to a set of new problems which require further investigation. The proposed methodology for test vector reordering can be improved if a sophisticated algorithm is used. As the problem of vector reordering is formulated as TSP (Travelling Sales Person) problem, many known approximation algorithm can be examined to obtain better results in terms of power minimization. Moreover the problem can be extended to co-optimize the peak and average power based on the proposed graph theoretic formulation.

The proposed work for minimization of peak power during test-cycle based on scan

chain reordering [1] requires further investigation to improve the delay fault coverage along with the peak power minimization. Also, the routing area overhead has to be taken care of because the power aware scan order may not be optimal for routing area. The possible solution for routing area optimization could be the clustering algorithm [20].

The methodology proposed for capture power reduction for modular SoC test assumes the gated scan flip flip structure. However, this kind of architecture is not suitable for most of the industrial designs because this may add extra timing delay if fully gated architecture is used. Hence the proposed methodology can be extended for non gated scan architecture also if by some mean the scan shift power can be controlled along with the capture power.

[1] refer Chapter 4

Bibliography

[1] N. Badereddine, P. Girard, S. Pravassoudovitch, C. Landrault, AVirazel, and H.-J. Wunderlich. Minimizing peak power consumption during scan testing: Test pattern modification with x-filling heuristics. In *Proc. of Int'l Conf. on Design and Test of Integrated System in Nano Technology*, pages 359 – 364, 2006.

[2] N. Badereddine, P. Girard, A. Virazel, S. Pravossoudovitch, and C. Landrault. *Controlling Peak Power Consumption During Scan Testing: Power-Aware Dft and Test Set Perspective*. Springer Berlin / Heidelberg, August 2005.

[3] N. Badereddine, P.Girard, S. Pravossoudovitch, A.Virazel, and C. Landrault. Scan cell reordering for peak power reduction during scan test cycles. *IFIP Iinternational Federation for Information Processing*, 240:267 – 281, 2007.

[4] J. Bang-Jensen and G. Gutin. *Digraph Theory, Algorithm and Applications*. Springer-Verlag, London, first edition, 2006.

[5] Y. Bonhomme, P. Girard, C. Landrault, and S. Pravossooudovitch. Power driven chaining of flip-flops in scan architectures. In *Proc. of International Test Conference*, pages 796 – 802, 2002.

[6] F. Brglez, D. Bryan, and K. Koźmiński. Combinational profiles of sequential benchmark circuits. In *Proc. of International Symposium of Circuits and Systems*, pages 1929 – 1934, May 1989.

[7] K. Butler, J. Saxena, T. Frayer, G. Herthenigton, A. Jain, and J. Lewis. Minimizing

power consumption in scan testing: Pattern generation and dft techniques. In *Proc. IEEE International Test Conference*, pages 335 – 364, 2004.

[8] S. Cheng and R. Kapur. How power-aware test improve reliability and yield. *EE Times EDA news online, www.eetimes.com*, 15th, September 2004.

[9] R. Chou, K. Saluja, and V. Agrawal. Scheduling test for vlsi systems under power under power constraints. *IEEE Transactions on VLSI Systems*, 5(2):175 – 185, 1997.

[10] F. Corno, M. Rebaudengo, M. S. Reorda, and M. Violante. On reducing the peak power consumption of test sequences. In *Proc. European Conf. on Circuit Theory and Design*, pages 247 – 250, 1999.

[11] F. Corno, M. S. Reorda, and G. Squillero. Rt-level itc'99 benchmarks and first atpg results. *IEEE Design and Test of Computer*, 17(5):44 – 53, July 2000.

[12] V. Dabholkar, S. Chakravarty, I. Pomeranz, and S. Reddy. Techniques for minimizing power dissipation in scan and combinational circuits during test application. *IEEE Trans. on Comp.-Aided Design*, 17(12):1325 – 1333, December 1998.

[13] V. R. Devanathan, C. P. Ravikumar, and V. Kamakoti. A stochastic pattern generation and optimization framework for variation-tolerant, power-safe scan test. In *Proc. of the International Test Conference, Santa Clara, USA*, page 13.1, October 2007.

[14] R. S. et al. Controlling peak power during scan testing. In *Proc. IEEE VLSI Test Symposium*, pages 153 – 159, 2002.

[15] X. W. et al. On low capture power test generation for scan testing. In *Proc. IEEE VLSI Test Symposium*, pages 265 – 270, 2005.

[16] H. Gabow, Z. Galil, and T. Spencer. Efficient implementation of graph algorithms using contraction. In *Proc. 25th Annual Symp. Foundation of Computer Science*, pages 347 – 357, 1984.

[17] M. R. Garey and D. S. Johnshon. *Computers and Intractability: A Guide to Theory of NP-completeness*. W. H. Freeman and Company, 1979.

[18] S. Gerstendorfer and H.-J. Wunderlich. Minimized power consumption for scan-based bist. In *Proc. of International Test Conference*, pages 77 – 84, 1999.

[19] P. Girard. Survey of low-power testing of vlsi circuits. *IEEE Design & Test of Computers*, pages 82 – 92, May 2002.

[20] P. Girard and Y. Bonhomme. Low power scan chain design: A solution for an efficient tradeoff between test power and scan routing. *Journal of Low Power Electronics*, 1:85 – 95, 2005.

[21] P. Girard, L. Guiller, C. Landrault, and S. Pravassoudovitch. A test vector ordering technique for switching activity reduction during test operation. In *Proceedings of Ninth Great Lakes Symposium on VLSI*, pages 24–30, 1999.

[22] P. Girard, C. Landrault, S. Pravossoudovitch, and D. Severac. Reduction of power consumption during test application by test vector ordering. *IEEE Electronics Letters*, 33(21):1752 – 1754, October 1997.

[23] A. Hertwig and H.-J. Wunderlich. Low power serial built-in self-test. In *Proc. of European Test Workshop*, pages 49 – 53, 1998.

[24] V. Iyengar, K. Chakravarty, and E. J. Marinissen. Test wrapper and test access mechanism co-optimization for system-on-chip. In *Proc. of International Test Conference*, pages 1023 – 1033, 2001.

[25] R. Kapur and T. Williams. Tough challenges as design and test go nanometer. *Computer*, 32(11):42 – 45, November 1999.

[26] N. S. Kim, T. Austin, D. Baaruw, T. Mudge, K. Flautner, J. S. Hu, M. J. Irwin, M. Kandemir, and V. Narayan. Leakge current: Moore's law meets static power. *IEEE Computer*, 36(12):68 – 75, December 2003.

[27] H. F. Ko and N. Nicolici. Rtl scan design for skewed-load at-speed test under power constraints. In *Proceedings of IEEE International Conference on Computer Design*, pages 237 – 242, 2006.

[28] J. LI, Q. XU, Y. HU, and X. LI. On reducing both shift and capture power for scan-based testing. In *Proc. Design Automation Conference*, pages 653 – 658, 2008.

[29] X. Lin, R. Press, J. Rajski, P. Reuter, T. Rinderknecht, B. Swanson, and N. Tamarapalli. High-frequency, at-speed scan testing. *IEEE Design and Test of Computer*, 20(5):17 – 25, September-October 2003.

[30] K. Miyase and S. Kajihara. Xid: Don't care identification of test patterns for combinational circuits. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 23(2):321 – 326, 2006.

[31] N. Nicolic and X. Wen. Embedded tutorial on low power test. In *Proceedings of 12th European Test Symposium*, pages 202–210, 2007.

[32] N. Nicolici and B. M. Al-Hashimi. Scan latch partitioning into multiple scan chain for power minimization in full scan sequential circuits. In *Proc. IEEE/ ACM Design Automation and Test in Europe*, pages 715 – 722, 2000.

[33] M. Pedram. Power minimization in ic desing: Principles and applications. *ACM Transactions on Desing Automation of Electronics System*, 1(1):3 – 56, January 1996.

[34] J. M. Rabaey, A. Chandrakasan, and B. Nikolic. *Digital Integrated Circuits A Design Perspective*. Pearson Prentice Hall, 482, F.I.E, Pataparganj, Delhi 110092, India, second edition, 2007.

[35] P. Rosinger, B. M. Al-Hashimi, and N. Nicolici. Scan architecture with mutually exclusive scan segment activation for shift and capture power reduction. *IEEE trans. on Computer-aided Design of Integrated Circuits and Systems*, 23(7):1142 – 1153, July 2004.

[36] P. M. Rosinger, B. Al-Hashimi, and N. Nicolici. Scan architecture for shift and capture cycle power reduction. In *Proc. International Symposium on Defect and Fault Tolerance in VLSI System*, pages 129 – 137, 2002.

[37] S. Samii, M. Selkälä, E. Larsson, K. Chakrabarty, and Z. Peng. Cycle-accurate test power modeling and its application to soc test scheduling. In *Proc. Intl. Test Conf.*, pages 1 – 10, 2006.

[38] S. Samii, M. Selkälä, E. Larsson, K. Chakrabarty, and Z. Peng. Cycle-accurate test power modeling and its application to soc test architecture design and scheduling. *IEEE Transanction on Computer-Aided Design*, 27(5):973 – 984, May 2008.

[39] R. Sankaralingam, R. R. Oruganti, and N. A. Touba. Static compaction technique to control scan vector power dissipation. In *Proc. of VLSI Test Symposium*, pages 35 – 40, 2000.

[40] R. Sankaralingam and N. A. Touba. Inserting test points to control peak power during scan testing. In *Proc. IEEE Symp. Defect and Fault Tolerance*, pages 138 – 146, 2002.

[41] J. Savir. Skewed-load transition test: Part i, calculus. In *Proceedings of International Test Conference*, pages 705–713, 1992.

[42] J. Savir and S. Patil. Broad-side delay test. *IEEE Transanction on Computer-Aided Design of Integrated Circuit and System*, 13(8):1057 – 1065, August 1994.

[43] J. Saxena, K. M. Butler, V. B. Jayaram, S. Kundu, N. V. Arvind, P. Sreeprakash, and M. Hachinger. A case study of ir-drop in structured at-speed testing. In *Proc. IEEE Int. Test Conference*, pages 1098 – 1104, September 2003.

[44] J. Saxena, K. M. Butler, and L. Whetsel. An analysis of power reduction techniques in scan testing. In *Proc. IEEE Int. Test Conference*, pages 670 – 677, 2001.

[45] M. Selkälä. *Test Data Analysis for Accurate Power Estimation.* Linköping University, Linköping, Sweden, June 2006.

[46] O. Sinanoglu, I. Bayraktaroglu, and A. Orailoglu. Reducing average and peak test power through scan chain modification. *Journal of Electronics Testing: Theory and Application*, 19:457 – 467, 2003.

[47] *DFT Compiler*. Synopsys, Inc., 2006. Version Y-2006.06-SP4.

[48] *TetraMAX*. Synopsys, Inc., 2006. Version Y-2006.06-SP4.

[49] J. Tudu, E. Larsson, V. Singh, and V. D. Agrawal. On minimization of peak power during soc test. In *Proc. European Test Symposium*, pages 25 – 30, 2009.

[50] J. Tudu, E. Larsson, V. Singh, and H. Fujiwara. Scan cells reordering to minimize peak power during scan testing of soc. In *Proc. IEEE Workshop on RTL and High Level Testing*, pages 34 – 48, November 2009.

[51] J. Tudu, E. Larsson, V. Singh, and H. Fujiwara. Scan cell reordering to minimize peak power during test cycle: A graph theoretic approach. In *Proc. IEEE European Test Symposium*, pages 24 – 28, May 2010.

[52] L.-T. Wang, C.-W. Wu, and X. Wen. *VLSI Test Principles and Architectures Design for Testability*. Morgan Kaufmann Publishers Elsevier, 500 Sansome street, Sanfrancisco, CA 94111, 2006.

[53] L. Whetsel. Adapting scan architecture for low power scan operation. In *Proceedings International Test Conference*, pages 863–873, 2000.

[54] Y. Wu and M. C.-T. Chao. Scan-chain reordering for minimizing scan shift power based on non-specified test cubes. In *Proc. IEEE VLSI Test Symposium*, pages 147 – 174, April-May 2008.

[55] Y. Zorian. A distributed bist control scheme for complex vlsi devices. In *Proceedings of the 11th IEEE VLSI Test Symposium*, pages 4 – 9, 1993.