UNIVERSITY OF SOUTHAMPTON
Faculty of Engineering, Science and Mathematics
School of Electronics and Computer Science

Test and Diagnosis of Resistive Bridges in Multi-V_{dd} Designs

by

Syed Saqib Khursheed

A thesis submitted for the degree of
Doctor of Philosophy

February 2010

UNIVERSITY OF SOUTHAMPTON

ABSTRACT

FACULTY OF ENGINEERING, SCIENCE AND MATHEMATICS
SCHOOL OF ELECTRONICS AND COMPUTER SCIENCE

Doctor of Philosophy

Test and Diagnosis of Resistive Bridges in Multi-V_{dd} Designs

by Syed Saqib Khursheed

A key design constraint of circuits used in hand-held devices is the power consumption, mainly due to battery life limitations. Adaptive power management (APM) techniques aim to increase the battery life by adjusting the supply voltage (V_{dd}) and operating frequency, according to the workload. APM-enabled devices raise a number of challenges for existing manufacturing test and diagnosis techniques, as certain defects exhibit V_{dd} dependent detectability. This means that to achieve 100% fault coverage, APM-enabled devices should be tested at all operating voltages using repetitive tests. Repetitive tests at several V_{dd} settings are undesirable as it increases the cost of manufacturing test. This thesis provides two new and cost-effective Design for Test (DFT) techniques to avoid repetitive tests thereby reducing test cost. The first technique uses test point insertion (TPI) to reduce the number of test V_{dd} settings. TPI capitalizes on the observation that each resistive bridge defect consists of a large number of logic faults, including detectable and non-detectable logic faults. It targets resistive bridges requiring test at higher V_{dd} settings, and converts un-detectable logic faults at the lowest V_{dd} setting, into detectable logic faults by using test points. Test points provide additional controllability and observability at the fault site. TPI has shown encouraging results in terms of reducing the number of test V_{dd} settings, however it does not achieve single V_{dd} test for all designs. Taking this issue into account, another gate sizing (GS) based DFT technique is proposed. It targets bridges that require multi-V_{dd} test and increases the drive strength of gates driving such bridges. The number of test V_{dd} settings are reduced minimizing test cost. Experimental results show that for all designs, the proposed GS technique achieves 100% fault coverage at a single V_{dd} setting; in addition it has a lower overhead than the TPI in terms of timing, area and power.

The V_{dd} dependent detectability of resistive bridges demands re-evaluation of existing diagnosis techniques, as all existing techniques use a single voltage setting for fault diagnosis, which may have a negative impact on diagnosis accuracy, affecting subsequent design cycle and yield. This thesis proposes a novel and cost-effective technique to improve diagnosis accuracy of resistive bridges in APM-enabled designs. It evaluates the impact of varying supply voltage on the accuracy of diagnosis and demonstrates how additional voltage settings can be leveraged to improve the diagnosis accuracy through a novel multi-voltage diagnosis algorithm. The diagnosis cost is reduced by identifying the most useful voltage settings and by eliminating tests at other voltages thereby achieving high diagnosis accuracy at reduced cost. All developed test and diagnosis techniques have been validated using simulations with ISCAS and ITC benchmarks, realistic fault models and actual bridges extracted from physical layouts.

Contents

List of Figures

List of Tables

List of Acronyms

ATE	Automatic Test Equipment
ATPG	Automatic Test Pattern Generator
AVS	Adaptive Voltage Scaling
BI	Bridge Intersection
BIST	Built-in Self Test
BL	Bridge Location
CAD	Computer Aided Design
C-ADI	Covered Analogue Detectability Interval
CMOS	Complementary Metal Oxide Semiconductor
CN	Common Nets
CNT	Contains
CP	Control Point
CSA	Carry Save Adder
CUT	Circuit under Test
D/ND	Detected/Not-Detected
DA	Deterministic Algorithm
DFT	Design for Test
DL	Defect Level
DSM	Deep Submicron
DUT	Device Under Test
DV	Detection Value
DVS	Dynamic Voltage Scaling
ECL	Exclusive Control point Candidate List
EDA	Electronic Design Automation
EMT	Empty
EXT	Exact
FC	Fault Coverage

FIC	Fan-in Cone
FP	Failing Pattern
FVC	First Valid Candidates
G-ADI	Global Analogue Detectability Interval
GDR	Gate Delay Ratio
GS	Gate Sizing
HPM	Hardware Performance Monitor
IC	Integrated Circuit
IDDQ	Supply current (Idd) in the quiescent state
LF	Logic Fault
LFSR	Linear Feedback Shift Register
LSC	Logic State Configuration
MOSFET	Metal Oxide Semiconductor Field Effect Transistor
MVTG	Multi-Voltage Test Generator
nm	nanometer
NMOS	N-channel MOSFET
NRINEV	Non-Redundant Interval at Non-Essential V_{dd}
OP	Observation Point
ORA	Output Response Analyzer
PA	Probabilistic Algorithm
PDA	Personal Digital Assistant
PDR	Path Delay Ratio
PECO	Probabilistic Estimate of Controllability and Observability
PI	Power Interface
	Primary Input
PLL	Phase Lock Loop
PMOS	P-channel MOSFET
PMScan	Power Managed Scan
POM	Primary Output Matching
PP	Partially Passing
ppm	parts per million
PRI	Passing Resistance Intersection
PV	Process and Voltage (variations)
RBF	Resistive Bridge Fault
RORS	Resistive Opens and Resistive Shorts
RRI	Resistance Range Intersection
SEU	Single Event Upset

SF	Stuck-at Fault
SI	Scan In
SLAT	Single location at a time
SO	Scan Out
SPICE	Simulation Program with Integrated Circuit Emphasis
SRAM	Static Random Access Memory
SVS	Static Voltage Scaling
TAT	Test Application Time
TDR	Total Detectable Resistance
TMAX	Synopsys TetraMAX
TPI	Test Point Insertion
USB	Universal Serial Bus
VC	Valid Candidates
V_{dd}	Supply voltage for CMOS transistor
VLV	Very Low Voltage
Y	Yield

Declaration of Authorship

I, Syed Saqib Khursheed, declare that this thesis and the work presented in it are my own and has been generated by me as the result of my own original research.

Test and Diagnosis of Resistive Bridges in Multi-V_{dd} Designs

I confirm that:

1. This work was done wholly or mainly while in candidature for a research degree at this University;
2. Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated;
3. Where I have consulted the published work of others, this is always clearly attributed;
4. Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work;
5. I have acknowledged all main sources of help;
6. Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself;
7. Either none of this work has been published before submission, or parts of this work have been published as listed on page 24

Signed:

Date:

Acknowledgements

I extend my most sincere gratitude to my supervisor Prof. Bashir M. Al-Hashimi, without whose support and guidance, the successful completion of this Ph.D. project would not have been possible. I am grateful for the research facilities provided by the School of Electronics and Computer Science at the University of Southampton and the Engineering and Physical Sciences Research Council (EPSRC) for funding this project. I also thank Dr. Peter Wilson for his supportive feedback during my MPhil transfer examination.

I am thankful to all my co-authors and all those people who guided and supported me technically during my Ph.D. These include Prof. Sudhakar Reddy, Dr. Paul Rosinger, Dr. Peter Harrod, Prof. Krishnendu Chakrabarty, Dr. Urban Ingelsson, Dr. Patrick Girard, Dr. Ilia Polian, and Dr. Bisjawit Mishra.

The constructive discussions and feedback of my colleagues and friends have also contributed greatly in this Ph.D. project. I therefore thank Rishad A. Shafik, Ke Li, Dr. Saket Srivastava, Dr. Arash Ahmadi, Mustafa Imran Ali, Alex Weddell, Amit Acharyya, Ghaithaa Manla, Sheng Yang, Karim El Shabrawy, Aissa Melouki, Evangelos Mazomenos, Hamed Shahidipour, Eduardo Mangieri, and Anton Kulakov for their support in both technical and non-technical discussions.

I deeply acknowledge the inspiration of Dr. Aimen El-Maleh, who introduced me to test research and encouraged me to pursue this Ph.D.

Finally, I would like to thank my parents (Dr. Khursheed Iqbal and Dr. Rukhsana Khursheed), brothers (Dr. Moazzum Khursheed and Naveed Khursheed) and last but not least my lovely wife (Yusra Saqib) for their continuous love, support and understanding throughout my Ph.D.

Chapter 1

Introduction

Energy-efficiency is a key requirement for portable, battery-powered appliances. Several adaptive power management methods have been employed in a wide range of consumer electronics to optimize their power consumption. A popular adaptive power management technique is scaling the supply voltage and operating frequency according to the processing load [Schmitz et al., 2004], as implemented in several state-of-the-art processors [Martin et al., 2002, Intel, 2007]. Typically, a design with adaptive power management has a set of discrete supply voltage/frequency settings it can switch between depending on the current workload and power saving mode. Multi-V_{dd} design [Keating et al., 2007] is another effective power saving technique, which operates gates on non-critical paths at a lower V_{dd} setting than those on critical paths thereby reducing overall power consumption. This work proposes cost-effective *test and diagnosis* solutions for a dominant deep submicron defect (bridge defect) in the context of multi-V_{dd} designs.

This chapter gives an overview of recent manufacturing test techniques commonly employed in devices using low power design techniques. The aim of this chapter is to provide preliminary information for the subsequent chapters in the thesis. Recent low power design techniques are discussed in Section 1.1. Section 1.2-1.6 summarize recent manufacturing test techniques including a discussion on fault models (Sec. 1.3), test generation (Sec. 1.4), diagnosis (Sec. 1.5), and design for test (DFT) techniques (Sec. 1.6). The contribution of each chapter is summarized in Section 1.7, and finally Section 1.8 presents the list of publications generated from the work presented in this thesis.

1.1 Low Power Design

The last decade has witnessed a tremendous increase in the usage of low-power battery driven devices, for example, smart phones, laptops and PDAs. The mobile phone industry has recorded a yearly growth rate of 24% in the number of mobile service subscriptions. At the start of the century, just 12% of the world's population had a mobile phone, which had risen in 2008 to about 61% (or approximately 4 billion subscriptions) [Wray, 2008, News, 2008]. The UK, with a population of around 60 million people, has in total 70 million mobile phone subscriptions [Dennis, 2008]. The demand for rich feature sets in these devices has increased as well, which support applications for: web-browsing, multimedia, email, GPS navigation etc., putting a severe stress on the battery life. Over the years, low power design techniques have evolved to support the demand for rich feature-set in these devices and to increase battery life.

Low power design techniques aim to reduce power consumption per unit time. Total power consumption can be divided into two main categories, i.e. dynamic and static power, as shown in Eq. (1.1). Dynamic power is consumed when the device is active and signals are propagated from one part to another. On the other hand, static power is consumed when the device is powered up but there is no activity in the device and no signal propagation. Dynamic and static power are both likely to increase in upcoming years and it is predicted that dynamic power will double from 90 nm to 45 nm devices, while static power will increase by 6.5 times [Keating et al., 2007]. This clearly poses a challenge for researchers in the field of hand-held electronic devices.

$$P_{Total} = P_{Dynamic} + P_{Static} \tag{1.1}$$

$$P_{Dynamic} = \alpha.f.C_L.V^2 \tag{1.2}$$

$$I_{Static} = I_{Subthreshold} + I_{Oxide} + I_{BTBT} \tag{1.3}$$

Dynamic power can be expressed using Eq. (1.2), where α is related to effective percentage of gates switching, V is the operating voltage, f is operating frequency and C_L is load capacitance which is proportional to the number of gates. From this equation, it can be seen that dynamic power is directly proportional to operational frequency, driven load capacitance, and most importantly, the square of the operating voltage. This means reducing operating voltage can produce more significant dynamic power savings than other parameters, i.e., f and C_L. Static power

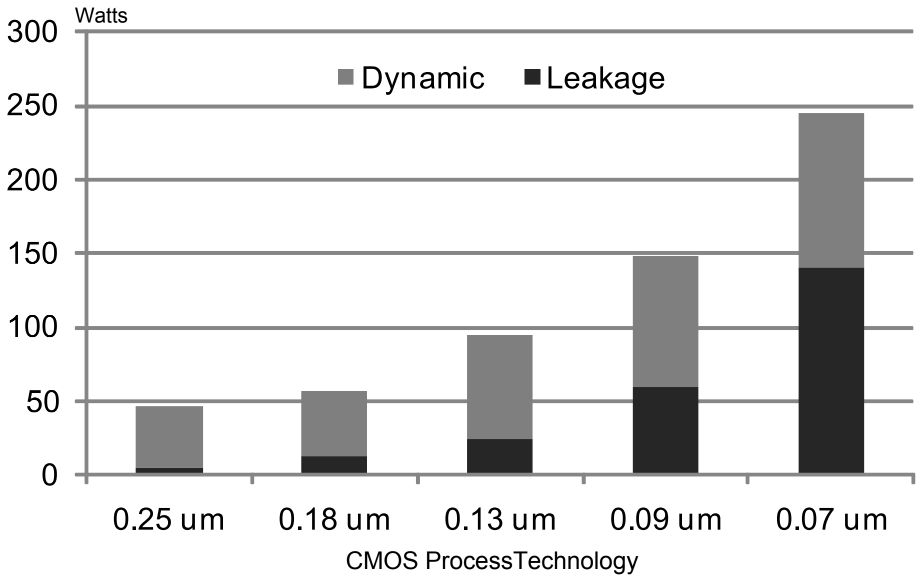


FIGURE 1.1: Static v Dynamic Power Consumption [Smith, 2009]

is mainly due to three types of leakage currents: subthreshold, gate-oxide and Band-To-Band Tunneling (BTBT) leakage currents as shown by Equation (1.3). Sub-threshold leakage is a dominant type of leakage current and is due to movement of minority carriers (holes for n-type and electrons for p-type material) from drain to source of a transistor, when it is operating in cut-off region ($V_{gs} < V_t$). One popular technique to reduce subthreshold leakage current is to use high V_t gates. As V_t has a negative exponential relationship with I_{Sub}, even a small increase in V_t results in a reduction in subthreshold leakage current [Kim et al., 2003b]. Gate leakage current is due to current through the gate oxide insulation and has significantly increased due to reduction in thickness of gate oxide T_{ox} with technology scaling, which is only a few atoms thick in 90 nm CMOS process technology. It can be reduced by using a high-k dielectric material. The BTBT current (I_{BTBT}) is due to a high electric field across a reverse-biased p-n junction between the source/drain and bulk of the CMOS device, which causes significant current to flow through the junction [Roy et al., 2003]. The increase of dynamic and static power consumption with technology scaling is shown in Figure 1.1. It can be seen that static power is a major contributor to total power consumption in 70 nm processes. Subthreshold and gate leakage currents are dominant causes of static power in nanometer CMOS, but it is also affected by Gate Induced Drain Leakage and Reverse Bias Junction Leakage [Fallah and Pedram, 2005].

Some of the most widely used low power design techniques include: Clock Gating, Multi-Vt design, and Multi-V_{dd} design techniques [Keating et al., 2007, Tiwari et al., 1998]. Clock gating is motivated by the fact that clock tree contributes towards switching activity and it can be as high as 50% of dynamic power consumption of a design. This is because of two reasons: firstly, the clock nets are long and drive high load capacitances. Secondly, they are subject to a high switching activity. Clock gating targets parts of the design that are not required and turn off the clock supply to them. This clock activity is disabled by adding logic elements in parts of the circuit such that logic elements fed by the flip-flops do not change their state un-necessarily. In an experiment conducted using two designs with and without clock gating, it was found that the power savings measured on a real chip varied from 34% to 43% on designs fabricated using 180 nm CMOS process technology [Pokhrel, 2007].

Multi-Vt design is another effective low power design technique commonly used to trade-off static power with speed. The idea is fairly straightforward, a design is synthesized using standard V_t cells and once the timing requirement is met, cells on non-critical paths are replaced by high V_t cells that are slower but produce smaller static power. On the other hand, low V_t and faster cells are placed on the critical path to meet the timing constraint; in this case design objective is to place minimum number of low V_t cells to meet timing [Luo et al., 2008]. It was shown in [Luo et al., 2008] that leakage current of a low V_t cell can be as high as 17.3 times that of high V_t cell. High V_t cells are 30% slower than their low V_t counterparts on 65 nm CMOS process technology.

The multi-$\mathrm{V_{dd}}$ design technique provides the highest amount of power savings [Keating et al., 2007] as it reduces the operating voltage of functional blocks according to their workload requirement. Unfortunately, voltage reduction comes at a cost of reduced operating frequency and therefore the voltage level and frequency setting of each block is determined after analysing the performance requirement of each individual block. The multi-$\mathrm{V_{dd}}$ design technique divides the design and supplies each block with a specific set of voltage setting in order to meet performance requirements of respective block(s). For example, a USB device has much lower performance requirements than the cache, therefore USB can operate at a lower voltage than that supplied to the cache without degrading the overall system performance. This is further shown in Figure 1.2, where the voltage level of path “B” is reduced thereby reducing power consumption, without affecting the overall performance of the design. From Eq. (1.2), it can be seen that supply voltage has a quadratic relationship with dynamic power and therefore reducing voltage has more pronounced effect on power savings than other parameters, i.e., f, and C_L.

It is important to note that all these low power techniques aim to reduce energy expenditure and not power alone. Battery life is determined by total energy used by a device and is given by an integral of power over time. Power is an instantaneous parameter, while energy shows total power spent over a period of time. Since low power design techniques increase delay, it is possible that a device consumes lower power (using low power design techniques) but the same amount of energy as a faster device (due to slow completion of tasks). Therefore, it is crucial to

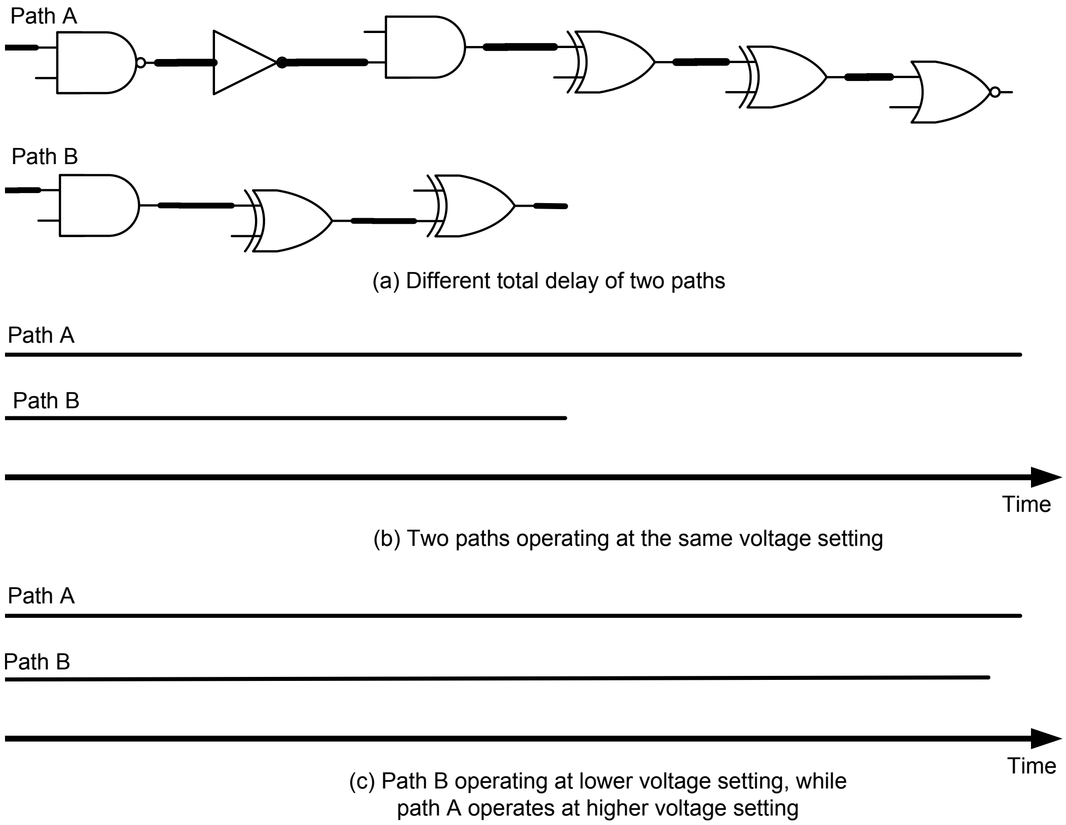

Path A
Path B
(a) Different total delay of two paths
Path A
Path B
Time
(b) Two paths operating at the same voltage setting
Path A
Path B
Time
(c) Path B operating at lower voltage setting, while path A operates at higher voltage setting

FIGURE 1.2: Multi Voltage design principle

analyze the impact of low power design technique on overall energy expenditure of the system to make it energy-efficient rather than power-efficient.

In general, multi-V_{dd} designs can be broadly categorized into two different types: First, a simple multi-V_{dd} scheme divides different blocks of the design according to their peak workload and performance requirement. Each block is then supplied with a static (fixed) voltage setting at which it operates and this type of voltage scaling is referred as Static Voltage Scaling (SVS). Second, a more sophisticated technique changes the voltage setting of each block dynamically according to its performance requirement. This second type of voltage scaling is referred as Dynamic Voltage Scaling (DVS) or Adaptive Voltage Scaling (AVS). AVS is the more advanced of the two as it takes into account the operating conditions for example, temperature, process and power supply variations, while DVS doesn't have this capability and is designed to operate under worst operating conditions at fixed voltage and frequency settings, for a given work requirement. A typical AVS-based system is shown in Figure 1.3. As can be seen, the energy management unit actively monitors the process and temperature variations in addition to performance requirements, as monitored by the hardware performance monitor, and varies the supply

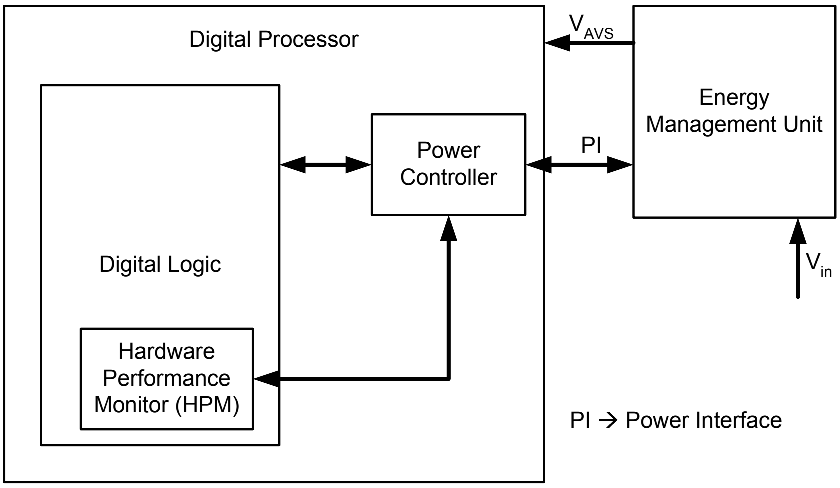


FIGURE 1.3: Adaptive Voltage Scaling [Zarr and Taylor, 2009]

voltage according to the workload. Significant power savings are achieved with Multi-V_{dd} design techniques in comparison to single V_{dd} design techniques, as shown in Figure 1.4. In a study conducted by [Chang and Pedram, 1997] it was shown that Multi-V_{dd} designs employing three V_{dd} settings achieve 40% more power savings than single V_{dd} design. This thesis also deals with devices employing Multi-V_{dd} settings in a DVS or AVS setup.

Multi-V_{dd} design achieves high savings in terms of energy expenditure, but it brings new challenges for design and test of integrated circuits. Recent research and a large number of publications in the literature have addressed challenges including task scheduling, level shifter design, floor planning, board level and test complexities of designs employing multi-V_{dd} settings [Pedram, 1996, Benini and Micheli, 2000, Lackey et al., 2002, Srivastava and Sylvester, 2003, Seo et al., 2005, Lee et al., 2006, Chen et al., 2007, Ingelsson, 2009]. In the next section, manufacturing test techniques commonly used in these devices are reviewed.

1.2 Manufacturing Test

The complex digital logic that constitutes an electronic design is tested to ensure that the design operates correctly and meets the desired specification before shipping it to the customers. During test, the design is configured in test mode and utilizes test logic (see Sec. 1.6 for more details on test logic) to support high quality test of the design. During testing the design is exercised with a large number of test patterns, which are sequence of ‘0s’ and ‘1s’ that utilize all gates/paths in

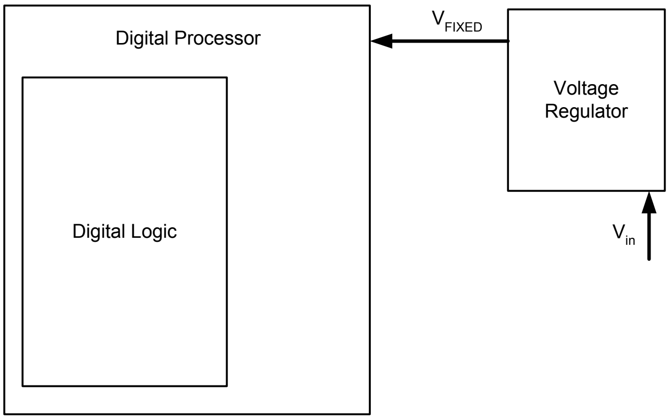


FIGURE 1.4: Fixed Supply Voltage [Zarr and Taylor, 2009]

the design[1] and the circuit response is compared with the expected response. The circuit is said to be *faulty* if the test output does not match the expected response and error is propagated to the output of the design, otherwise it is considered fault-free. Two types of faults lead to error propagation to the primary output of the design, which include manufacturing defects and soft errors. Manufacturing defects are introduced during manufacturing process and are permanent after their occurrence. Soft errors on the other hand are due to α-particles or neutrons and are attributed to operating conditions and environmental factors. Soft errors can alter the signal value, but since the effect is "transient" in nature, these are referred as "Single Event Upset" and affect the in-field reliability of the device. The impact of soft errors is increasing with technology scaling and reduction in operating voltage. The lower operating voltage reduces the amount of charge required to change the logic value of a gate, also called "critical charge". Since soft errors are transient in nature, manufacturing test can't detect such errors, therefore different design techniques have been proposed to improve in-field reliability of a device in the presence of soft errors by using Gate Sizing [Zhou and Mohanram, 2006], Redundancy [Gomaa and Vijaykumar, 2006], and Error Correction [Karnik and Hazucha, 2004, Meaney et al., 2005, Ejlali et al., 2006]. These methods are generally referred as "Design for Reliability".

The test community has a general consensus that delay in finding out and repairing a defective device has very high cost, which increases by a factor of 10 times, between different stages of manufacturing (from device to board level, to system level and lastly to in-field operational device). This is also referred as "rule of 10" [Wang et al., 2006]. In a manufacturing process, it is desirable to achieve high yield, which is the ratio of total acceptable parts to the total number of parts fabricated and shown in Eq. (1.4). The yield is usually considered acceptable at 500 defective parts per million (ppm) of fabricated parts, and very high quality at 100 ppm, however

[1] the number of test patterns depends on the size and complexity of the design

the test community aims to achieve *zero defect* target that is to have less than or equal to 3.4 defective ppm [Girard et al., 2009].

$$Yield = \frac{\textit{Total acceptable parts}}{\textit{Total number of fabricated parts}} \tag{1.4}$$

The purpose of manufacturing test is to reduce the *reject rate*, which is the ratio of field-rejected parts to the total number of parts passing manufacturing test, and is given by Eq. (1.5) and Eq. (1.6) [McCluskey and Buelow, 1988].

$$\textit{Reject Rate} = \frac{\textit{Total faulty parts passing manufacturing test}}{\textit{Total number of parts passing manufacturing test}} \tag{1.5}$$

$$\textit{Defect Level} = 1 - Y^{(1-FC)} \tag{1.6}$$

where, “Y” represents yield and “FC” represents fault coverage, which is used to measure test quality. Fault coverage is an important parameter to quantize the percentage of total faults detected by a test while considering the complete fault domain; fault coverage is expressed by Eq. (1.7),

$$FC = \frac{\textit{Detected Faults}}{\textit{Total Faults}} * 100 \tag{1.7}$$

Eq. (1.6) shows the relationship between yield, defect level and fault coverage. For example, assume yield of a certain process is 50% and fault coverage of manufacturing test is 95%, the defect level can be calculated using Eq. (1.6), i.e., $1 - (0.5)^{(1-0.95)}$ = 0.034 or 3.4% of shipped devices are defective or the defect level is 34,000 ppm. Therefore, to reduce the defect level to the acceptable limit of 500 ppm for the same process yield, a much higher quality test should be used. The fault coverage of such a test should be FC = 1-(log(1-DL)/log(Y)) = 99.93%.

Yield is also associated with the failure rate, λ, of a device, which represents the frequency of failing products per unit time. It can be understood using the popular Bathtub curve in reliability theory, shown in Figure 1.5. As can be seen, the failure rate can be broadly categorized into three main sections: infant mortality, working life and wear out period. The infant mortality period (with decreasing failure rate) occurs when a product is in its early production stage. Failures that occur in this period are mostly attributable to poor process or design quality, which leads to poor product quality. The product should not be shipped during this period to avoid massive field

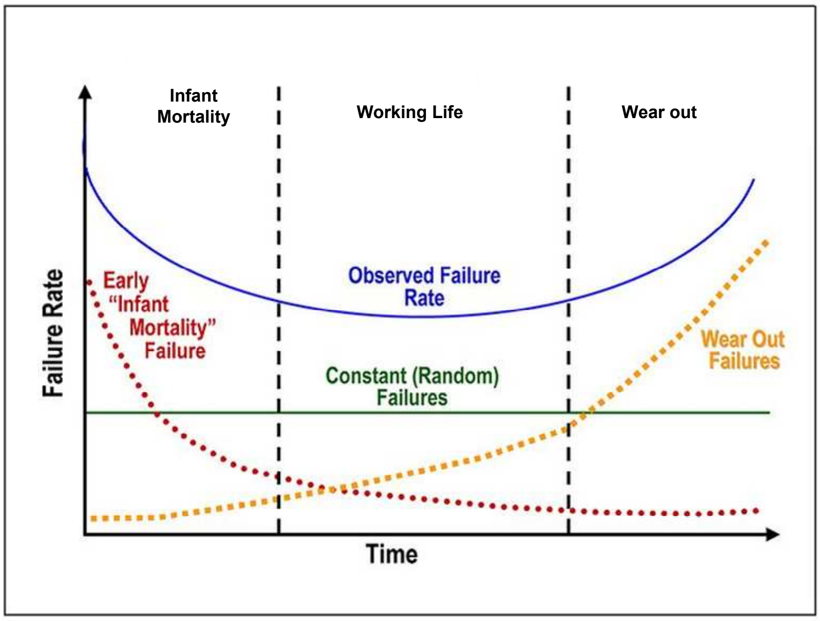


FIGURE 1.5: Bathtub curve [Yan and English, 1997]

returns. The working life period (with constant failure rate) represents the product's "working life". Failures during this period tend to occur randomly. The wearout period (with increasing failure rate) indicates the "end-of-life" of the product. Failures during this period are caused by age defects, such as metal fatigue, dielectric breakdown etc. For electronic products, this period is of less concern because end users often replace electronic products before the devices reach their respective wearout period.

1.3 Fault Models

Testing of digital circuits rely on fault models which are meant to mimic the physical behaviour of defects while taking into account all physical details linked with the behaviour of a defect at the device level. Fault models are important for test generation, fault simulation, fault diagnosis and quality prediction in the following way:

1. Fault simulation programs are built around fault models and are meant to measure the fault coverage matrix shown by Eq. (1.7) of a given test set.
2. Test generation programs or Automatic Test Pattern Generator (ATPG) benefit from fault models in two different ways. Firstly, they are guided by fault models that point towards

a fault requiring test, thereby providing a measure of completeness to the test process. Secondly, they provide a measure of effectiveness by computing the fault coverage matrix, shown by Eq. (1.7), which can be used to compare the efficiency of different ATPG programs [Girard et al., 2009].

3. Diagnosis and silicon debug programs are meant to identify the location and type of defect causing a malfunction in the circuit. Different types of diagnosis techniques exist in literature but all of them are built around a fault model. It was shown in a study [Aitken and Maxwell, 1995] that higher diagnosis accuracy can be achieved by using a simple diagnosis technique with an advanced fault model in comparison to advanced diagnosis technique with a simple fault model.

4. The defect level shown by Eq. (1.6) is used to determine the quality of a shipped product, which capitalizes on fault models to calculate fault coverage [Williams and Brown, 1981, McCluskey and Buelow, 1989].

Fault models are used to study and simulate the defect behaviour and are also used to generate test patterns that are meant to excite and propagate the fault to primary output(s). There are many different physical defects, for example, resistive shorts [Hao and McCluskey, 1993], resistive opens [Zain Ali and Zwolinski, 2006, Kruseman and Heiligers, 2006], transmission gate open [Chang and McCluskey, 1996a, Zain Ali and Zwolinski, 2006], gate oxide shorts [Soden and Hawkins, 1986, Chang and McCluskey, 1996b], threshold voltage shift [Hao and McCluskey, 1993], diminished drive strength [Chang and McCluskey, 1996a] etc. Therefore there is no single fault model that may capture the impact of each one of these at higher level of abstraction and this is why test is generated considering a number of defects and their respective fault models. Some well known and commonly used fault models include the following:

1.3.1 Stuck-at fault

Some defects can un-intentionally cause a logic signal to get connected to one of the power rails, i.e., V_{dd} or Gnd, forcing the logic node to be clamped at the voltage of the rail causing the stuck-at fault. The fault is referred to as “stuck-at 0” in a case where the node is connected to the ground rail. On the other hand, if a node is clamped to V_{dd}, it is referred as “stuck-at 1”. The stuck-at fault model is one of the most widely-used fault models for test generation. The nodes affected by this type of fault are either an input or output of a gate and it is also known as a gate-level stuck-at fault model [Wadsack, 1978, Park et al., 1994, Patel, 1998, Bushnell and Agrawal, 2000].

1.3.2 Stuck-open, Stuck-short faults

The stuck-open fault models a physical scenario, where the drain or source of a transistor is disconnected inside a gate leading to faulty behaviour of the gate [Soden et al., 1989]. Stuck-open faults can't be detected using the stuck-at fault test and it needs two test vectors for detection. The first test vector drives the output of a gate to logic high or low, while the second test vector compliments the output logic value using each transistor in the pull-up or pull-down network of a CMOS gate. Stuck-short faults produce a conducting path between V_{dd} and ground and may be detected by a test technique called I_{DDQ} testing, that monitors the current flow during steady-state condition [Bushnell and Agrawal, 2000].

1.3.3 Bridging fault

The bridging fault models a physical scenario where interconnect lines are accidentally connected with one another, thereby deviating the circuit behaviour from ideal. As shown in Figure 1.6-A, two interconnects are connected forming a bridge between outputs of driving gates (shown as D_1 and D_2) and successor gates (shown as S_1 and S_2). The bridge fault is excited only by driving D_1, D_2 at opposite logic values, which is also called "fault activation". The bridging fault model has evolved over the years and four different models have been proposed, which include:

1. The Wired-AND, Wired-OR bridge fault model was meant for bipolar devices and is shown by Figure 1.6-B, which feeds the same value to the successor gates (S_1 and S_2). The value fed is determined by either logic AND or logic OR functions.

2. This model was replaced by the dominant bridge fault model, shown in Figure 1.6-C, in which the logic value interpreted by successor gate depends on the strength of the gates driving the bridge and stronger gate drives the successor gates.

3. The dominant model was replaced by the four-way bridge fault model shown in Figure 1.6-D, also known as the dominant-AND/dominant-OR bridge fault model. This model assumes that only one driving gate dominates and only one successor gate is affected by the logic value calculated using the logic-AND or logic-OR function.

4. Finally, the most sophisticated fault model is called *Parametric Bridge* fault model, shown in Figure 1.7, which takes into account: drive strength of driving gates (D_1 and D_2), logic threshold of successor gates (V_{LT1} and V_{LT2}) and treats the resistance range as a continuous parameter with value from [0 - ∞), shown as variable resistor R_{sh} in Figure 1.7. The

> voltages on inputs of successor gates V_0 and V_1 vary with changes in bridge resistance and the logic value interpreted depends on the logic threshold voltage of each input of the respective gate (V_{LT1} and V_{LT2}). For example, for the range of resistances for which V_1 is less than V_{LT2}, gate S_2 interprets faulty logic 0 at its input I_2 and for the voltage range where V_1 is greater than V_{LT2} the faulty behaviour disappears and the correct logic value is interpreted by S_2 at input I_2. The research presented in this thesis employs the parametric bridge fault model, which is discussed in detail in Section 2.1.

Resistive bridges represent a major class of defects in deep submicron CMOS and have received increased attention with regard to modeling, test generation and diagnosis. Several publications have shown high occurrences of resistive bridges in CMOS designs [Ferguson and Shen, 1988, Galiay et al., 1980, Hawkins et al., 1994, Sengupta et al., 1999, Polian et al., 2005]. They are formed during the manufacturing process by a redundant metal connecting two nodes of a design, which deviates the behaviour of design from ideal (or desired) behaviour. This type of defect can be classified into intra-gate and inter-gate defects. Intra-gate bridge defects are due to redundant metal in a gate for example, between gate and drain of two transistors of a NOR gate. On the other hand, inter-gate bridge defects are due to a redundant metal between interconnects as illustrated by Figure 1.7. An experimental study presented in [Sousa et al., 1991, Engelke, 2009] shows that inter-gate defects have a much higher occurrence than intra-gate defects. Another study shows that inter-gate bridge defects can constitute 50% or more of total defect count [Ferguson and Shen, 1988]. This thesis focuses only on inter-gate bridge defects and assumes that at a given time, a fault-site (all gates in Figure 1.7) is operating at a specific voltage setting, i.e., V_1, V_2, or V_3. These inter-gate bridge defects will be referred to as bridge defects from now onwards.

1.3.4 Delay fault

Delay faults model the behaviour caused by process variation and physical defects, for example, resistive opens and resistive bridges[2], that may cause excessive circuit delay and violate circuit timing [Franco and McCluskey, 1991, Majhi and Agrawal, 1998, Nassif, 2000, Kim et al., 2003a]. On one hand, larger than expected delays cause data set-up time violations at the inputs of flip-flops (or latches) causing a manufactured circuit to fail to operate at the desired frequency of operation. Larger delays are normally referred to as delay faults. On the other hand, delays smaller than expected cause hold time violations leading to circuit failure and have been studied

[2] Resistive bridges and resistive opens are discussed in detail in Section 2.1 and Section 2.2 respectively.

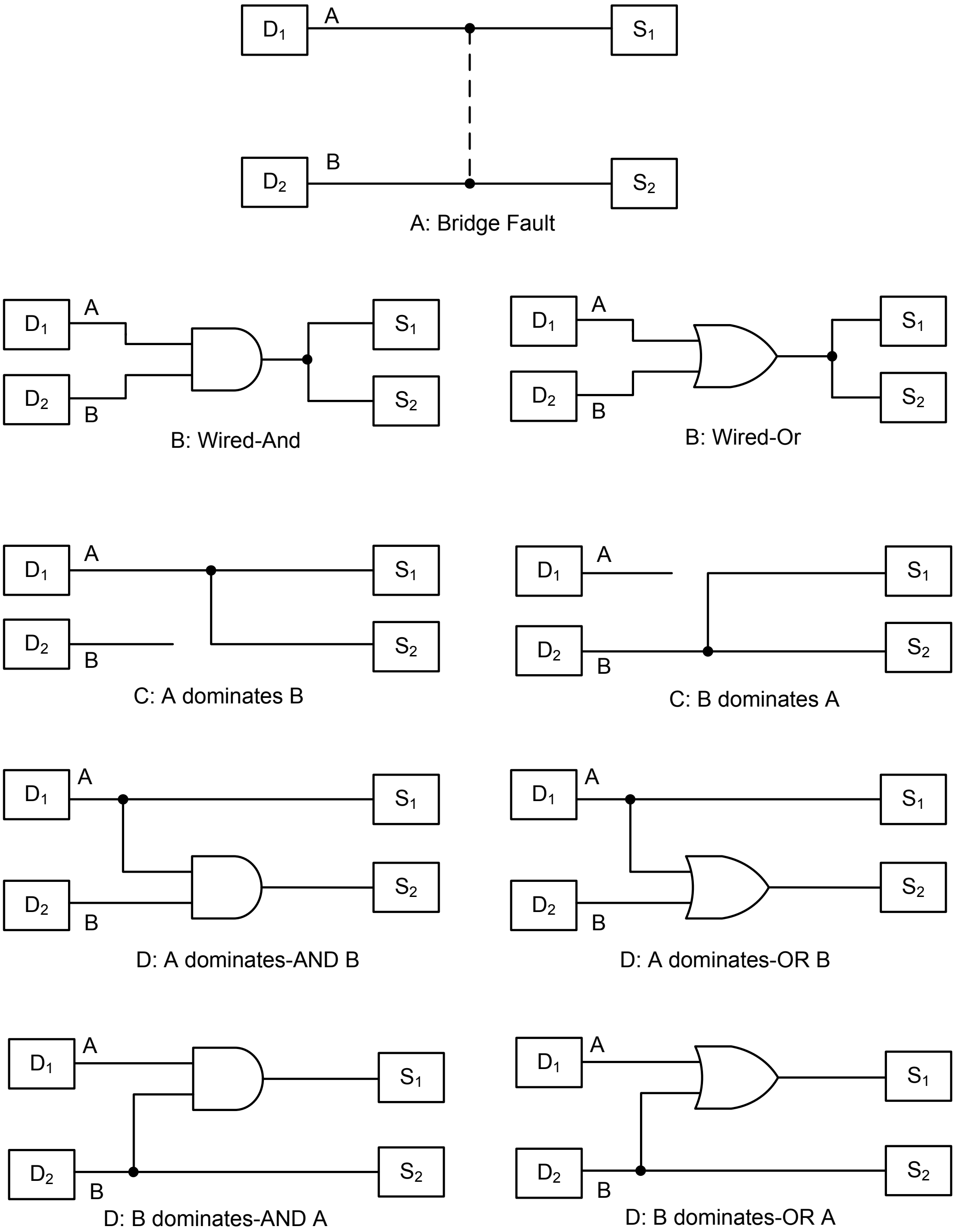


FIGURE 1.6: Evolution of Bridge Fault

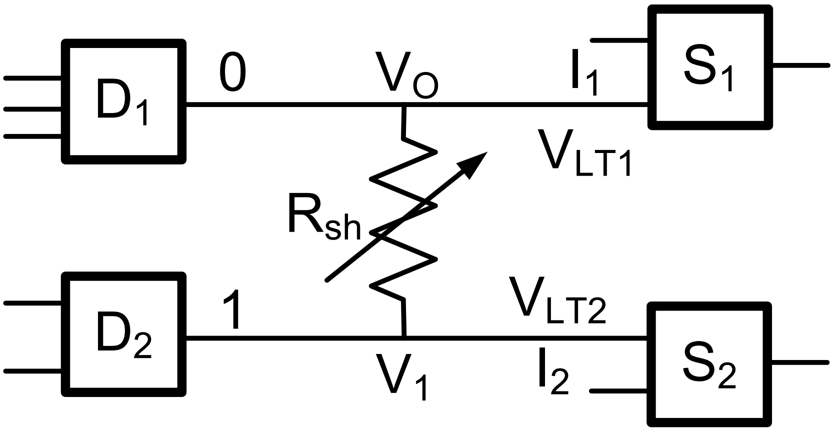


FIGURE 1.7: Parametric bridge fault model

in the context of scan chain failures. There are different types of delay fault models, which include the following:

1. The gate-delay fault models the behaviour of a signal while propagating through a gate, if a signal violates its timing due to excessive delay through the gate [Pramanick and Reddy, 1997].
2. The path-delay fault models the cumulative delay of a path to include gates and interconnects that exist in that path of a circuit [Lin and Reddy, 1987]. Small delay fault is a class of path-delay fault and it models faults that introduce less than one clock cycle delay and has received increased attention in recent years [Yilmaz et al., 2008, Goel et al., 2009].

An overlap exists between fault detection using different fault models, therefore a test to detect the small delay fault may also detect stuck-at faults. This is why manufacturing test commonly applies tests targeting delay faults first, followed by stuck-at faults, bridge faults and finally transistor level stuck-open faults to achieve a high fault coverage in the minimum possible test application time [Girard et al., 2009].

1.4 Test Generation

Test generation targets logic faults and produces an error at one of the primary output(s) of the design, where the logic fault belongs to the fault domain of a specific fault model. Test generation consists of two phases, fault activation and fault propagation. Fault activation produces logic values on the node that is opposite to that produced by the fault. For example, in the case of the circuit shown in Figure 1.8 line S with fault stuck-at 0 can be activated by producing a logic-1 at that node, which is possible by applying a logic-0 on input B that produces the opposite value on line S. Fault propagation directs this fault effect from the fault site to one of the primary output(s), such that a distinction can be made in the behavior of faulty design from a fault-free design by a simple comparison. The fault shown in Figure 1.8 can be propagated to the primary output by producing a non-controlling value at the other input of gate G7, which is possible by applying logic-1 at input A, of gate G1. The test generator returns an input test vector consisting of boolean values that produce this distinction in the presence of targeted faults. For the case where line S is stuck-at 0, test vector 00XXX detects the fault (where X represents the don't care condition on respective inputs) by activating and propagating the fault to primary output Y. Similarly, line T stuck-at 1 can be detected by a test pattern X0X01. The don't care bits of the test vector are randomly filled by test generators before producing a final test set. The don't care bits in a test pattern are exploited for various purposes, for example, test compaction, test compression and low power ATPG.

The purpose of test compaction is to reduce test data volume without affecting the fault coverage of the original test set [El-Maleh et al., 2006]. In the above example, using a test merging algorithm [El-Maleh and Khursheed, 2007], the two test vectors 00XXX, X0X01 can be combined to make a single test vector, i.e., 00X01 to detect both logic faults with just one test vector. Test compaction algorithms can be broadly categorized into two types: static compaction and dynamic compaction [Rudnick and Patel, 1999]. Static compaction algorithms attempt to reduce the test size after test generation and are applied as a post processing step to test generation algorithms. Dynamic compaction algorithms are a part of test generation procedure and attempt to reduce the test size at the same time as test generation.

Test compression attempts to reduce test data volume by utilizing don't care bits and encoding of test vectors [Gonciari et al., 2003, Touba, 2006, Kapur et al., 2008]. Automatic Test Equipment (ATE) has limited memory and number of channels (ATE bandwidth) that severely limits test speed resulting in long test application time. Test compression aims to alleviate these problems by encoding test data in a compressed form so less data needs to be transferred, thereby reducing the test time and the need for tester memory. The encoded data is decompressed on-chip through dedicated circuitry before its application to CUT. In a similar way, test data response is compressed to efficiently utilize ATE bandwidth. Compression techniques utilize a large number of don't care bits in test patterns that are normally filled by test generators, these test pattern bits are left un-filled by ATPG and are subsequently used by compression algorithms for test data volume reduction. This is achieved without compromising the fault coverage of test data.

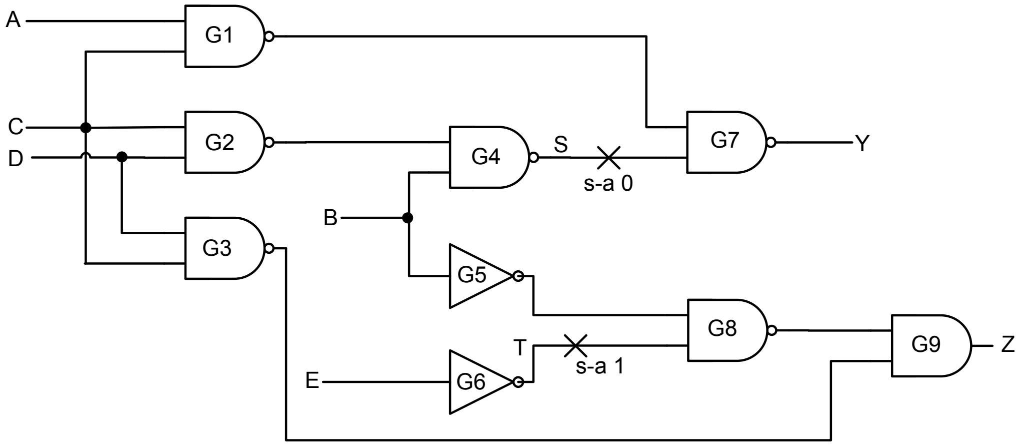


FIGURE 1.8: C17- ISCAS Design

Low power ATPG algorithms use don't care bits to reduce switching activity at the primary inputs of the design, thereby reducing the switching activity of the design [Wang and Gupta, 1998, Zhang and Roy, 2000]. From Equation (1.2) it is clear that switching activity has direct proportionality with dynamic power and therefore results in a reduction of dynamic power. For instance, in the above example, the two test vectors 00XXX, X0X01 can be transformed into 00001, 00001 to achieve lower switching activity at the inputs of the design.

Test generation for sequential circuits also requires fault activation and propagation to primary outputs, but its complexity increases due to the presence of flip-flops. A test vector for detecting a fault in sequential circuits typically requires more than one test vector, as fault activation and propagation traverses through flip-flops of the design. Due to the high complexity of designs with large number of flip-flops and multi-million gate count, test generation doesn't result in satisfactory fault coverage and the EDA industry has moved towards scan design that converts sequential circuits into combinational by replacing flip-flops with scan cells. Scan design transforms the test generation complexity of sequential circuits to that of combinational thereby significantly reducing test generation effort [Fujiwara and Toida, 1982, Jha and Gupta, 2003]. Scan design, its types and benefits are discussed in detail in Section 1.6.2.

1.5 Diagnosis

This section presents a brief overview of various diagnosis techniques, and serves as a background for Chapter 5 of this thesis.

Manufacturing test separates faulty circuits from fault-free to ensure the high quality of shipped products. All faulty circuits are analyzed to determine the root cause of their failure and this process is called diagnosis [Waicukauski and Lindbloom, 1989, Henderson and Soden, 1997]. The purpose of diagnosis is to determine the location and type of defect that deviates the circuit from ideal behaviour. Volume diagnosis takes into account a large set of failing ICs and statistical analysis is performed to figure out yield-limiting defects and design issues. This information is used for improving the subsequent design cycle and yield [Hora et al., 2002].

Diagnosis algorithms can be broadly categorized into three different types:

1. The effect-cause algorithm uses fault observing output(s) of a circuit and isolates the logic structures feeding those outputs for further analysis [Abramovici and Breuer, 1980, Wu and Rudnick, 1999, Abramovici et al., 1998]. In the effect-cause diagnosis algorithm, the circuit is traced backwards i.e., from primary outputs to primary inputs, using the fault observing output(s) found by failing patterns. As a result, a list of all possible fault candidates is generated, let FL denote this list. The algorithm further analyzes the failing patterns by fault-simulating each pattern in the presence of a fault $f \in FL$. It compares the output response of the tester with that of fault simulation and only in the case of a match, a fault is added to a list FL'. This step further reduces the size of potential candidates. After analysing all failing patterns the list of potential candidates is further reduced by solving a minimum set cover to determine common faults across all the failing patterns in the list FL' and the outcome is stored in a separate list FL''. Next, all the faults in FL'' are fault-simulated using each of the passing patterns, and the algorithm updates a counter whenever a fault is detected by a passing pattern. Finally, all faults are sorted using the mismatch count to represent the likelihood of each fault as a root cause of IC failure [Zou et al., 2007].

2. The cause-effect algorithm uses a database containing the output response of a circuit in the presence of a specific fault, when certain test pattern(s) are applied to the faulty design. This database is referred to as a dictionary and is generated using a fault model and a test set. The dictionary holds the test response of a circuit in the presence of a fault, which is compared with the observed output and this comparison is used to reduce the size of potential fault locations. For ease of comparison and higher accuracy of fault diagnosis, test patterns that detect the minimum number of faults per test are desirable to reduce the number of potential fault locations; such test patterns are referred to as high-resolution test patterns. A test pattern is said to have SLAT (Single location at a time) property if it propagates a single fault to the primary outputs of the circuit [Bartenstein et al., 2001, Huisman, 2004]. High resolution tests for improving diagnosis accuracy

is well researched [Camurati et al., 1990, Gruning et al., 1991, Agrawal et al., 2003, Veneris et al., 2004, Bhatti and Blanton, 2006]. These techniques can alter the usual test generation that produces a distinction between good and faulty circuits by producing a distinction between two faults. This is achieved by activating and propagating each fault through a different path [Camurati et al., 1990, Gruning et al., 1991].

3. Adaptive diagnosis techniques simultaneously perform diagnosis and test generation. The test response of a circuit under diagnosis is analyzed to guide test generation and is used to improve resolution of the subsequent test. This process continues until acceptable diagnosis resolution is achieved. Such diagnostic ATPG does not rely on pre-computed fault dictionaries [Gong and Chakravarty, 1995, Holst and Wunderlich, 2007].

1.6 Design for Testability

Testability analysis is different from fault modeling as it evaluates the relative degree of difficulty in testing each node in a design. Design for Testability (DFT) techniques evaluate a circuit and modify it to achieve higher test quality [Williams and Parker, 1983]. The relative difficulty in testing a node has two main components, i.e., Controllability and Observability [Abramovici et al., 1998]. Controllability of a node represents the relative difficulty in setting logic-1 or logic-0 at a node, while observability measures the relative difficulty in observing the logic value of a node at the primary output(s) of the design. A node is said to be easily testable if it can be easily controlled and observed. A number of different testability algorithms have been reported in the literature which assign controllability and observability to a node [Chandra and Patel, 1989]. These measures are used by designers to modify the design to achieve higher testability of a node and overall design, thus achieving high quality test. Four commonly-used DFT techniques are discussed next:

1.6.1 Test Point Insertion

Test Point Insertion (TPI) is a DFT technique that adds logic elements (referred as *test points*) to increase controllability or observability of a node [Hayes and Friedman, 1974, Hayes, 1974]. Controllability of a node can be improved by adding an AND/OR gate and connecting the other input of the gate to the test input, which is connected with the scan chain. In this way, an AND gate is used to improve 0-controllability and an OR gate is used to improve 1-controllability of a node. Traditionally, AND/OR gates have been used for improving controllability, but more

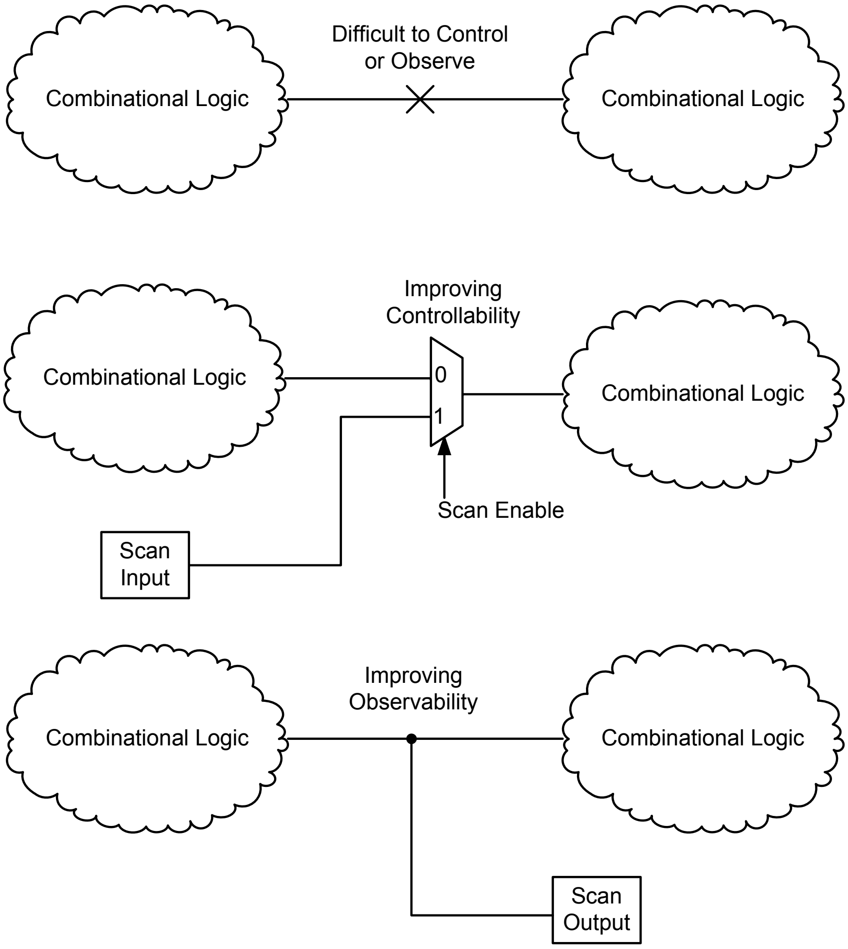


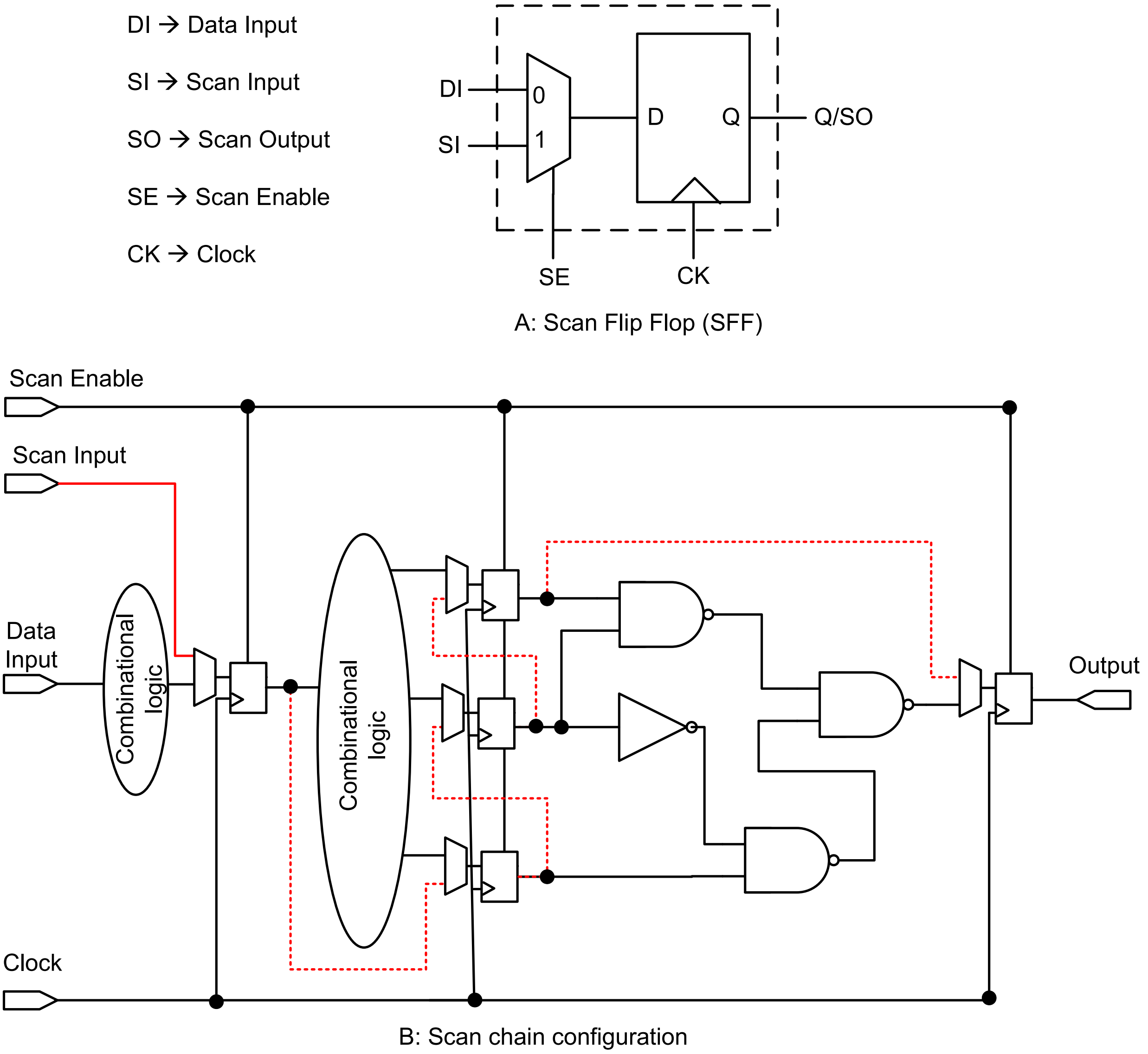
FIGURE 1.9: Test Point Insertion

recently a scan cell feeding multiplexer has been used to improve logic-1 and logic-0 controllability at the same time. This is further shown in Figure 1.9, where a multiplexer is added to improve the controllability of a desired logic value, similarly an observation point can be added to improve observability of a node. Test points have been used for improving fault coverage and compaction [Geuzebroek et al., 2000] by accessing parts of logic that are otherwise difficult to access, but they can violate timing if used in critical path of the design [Vranken et al., 2004]. TPI incurs an area and power overhead due to the additional logic inserted in the design [Abramovici et al., 1998].

1.6.2 Scan Chains

It is known that ATPG complexity for sequential circuits is much higher than for combinational circuits [Marchok et al., 1996, Cheng, 1996]. The presence of memory elements (flip-flops and latches) in sequential circuits poses a challenge for test generation. It is because of these memory elements that the controllability and observability of a node reduces in sequential circuits, negatively affecting testability and test generation effort. A test for a single stuck-at fault in a sequential circuit may require a long sequence of test vectors for detection. This has negative implications on test generation for highly complex sequential designs with large numbers of memory and logic elements, and therefore a very large number of possible logic faults. On one hand, logic faults may become untestable and on the other hand detectable logic faults may need a large number of test vectors, which increases test application time thereby increasing test cost. For these reasons, DFT techniques such as Scan Chains are introduced that convert flip-flops into scan cells allowing easier access to all nodes by treating sequential designs as combinational during test mode. A typical scan cell and its conversion from a flip-flop is shown in Figure 1.10-A. It works on the principle of converting flip-flops into scan flip-flops (also called scan cells) and connecting all scan cells together to construct a shift register, called a scan chain, that is used to improve the testability of the sequential design. Scan chain layouts, where all scan cells are connected with one another are called “Full scan designs” and scan chain layouts where some of the flip-flops are converted to scan cells are called “Partial scan designs”. Scan chains operate in three different modes: normal mode, shift mode and capture mode. The circuit operates in its original configuration during normal mode and uses shift mode and capture mode during test mode. During test mode the test pattern is scanned-in and scanned-out of the scan chain using shift mode. The test pattern is then applied to the combinational logic using capture mode, i.e. capturing the test response. The test pattern is clocked in the scan chain using a scan-in input port and then shifted from one scan cell to another using scan-out and scan-in ports, until the complete test pattern is loaded. The test response is captured by scan cells and is scanned-out using the shift mode of operation. The scan-in port of the first scan cell and scan-out port of the last scan cell are connected to the primary input and primary output of the design, which are connected to the Automatic Test Equipment (ATE) or Built-in Self Test (BIST) engine during test mode. Figure 1.10-B shows scan configuration during test mode. Test application time can be reduced by shifting out the test response while shifting in a new test pattern at the same time.

The scan chain has an area overhead but it simplifies the test generation complexity of sequential designs to that of combinational logic blocks. This allows using combinational ATPG for sequential designs.



A: Scan Flip Flop (SFF)

B: Scan chain configuration

FIGURE 1.10: (A) Scan cell architecture showing a typical scan cell and (B) Scan chain configuration during test mode

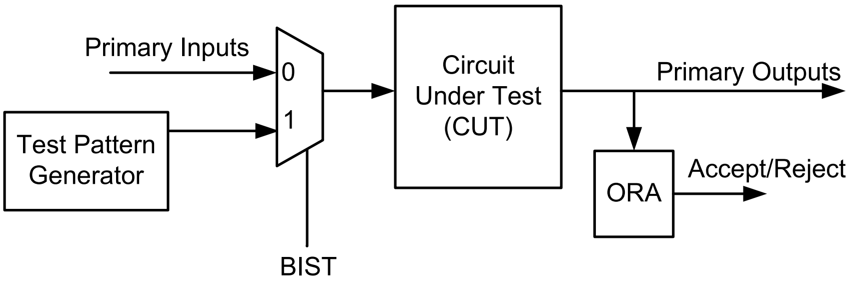


FIGURE 1.11: BIST Architecture [Girard et al., 2009]

1.6.3 Built-in Self-Test

Conventional test uses Automatic Test Equipment (ATE) for test applications that requires scanning-in test patterns through scan chains, capturing test response and finally scanning-out test response, where it is further analyzed to ensure that Circuit Under Test (CUT) is working properly. For complex designs with millions of gates requiring very large number of test patterns, the test application time becomes excessively long making the overall test process expensive. Built-in Self-Test (BIST) is a DFT technique that is widely employed to overcome these issues [Agrawal et al., 1993a,b]. It is achieved by deploying test modules for test pattern generation and response analysis along with the real chip, making the test engine a part of design. During test mode when the BIST engine is activated, a test pattern generator starts test generation and application, test response is then analyzed and the BIST engine generates a Pass/Fail signal depending on the outcome. A typical BIST architecture is shown in Figure 1.11. As can be seen, a test pattern generator and an output response analyzer (ORA) are part of design. Pseudo-random test patterns are generated using a linear feedback shift register (LFSR) that generates test patterns for testing the CUT [Abramovici et al., 1998]. Test generation using BIST has been an active area of research to achieve high fault coverage, which is not possible using random test patterns alone. This includes techniques using exhaustive, pseudo-exhaustive test patterns, additional hardware to keep test patterns for hard-to-detect faults as exhaustive test are impractical for large designs with multi-million gates [Chatterjee and Pradhan, 2003]. BIST can be incorporated in a design using two different architectures: test-per-scan BIST and test-per-clock BIST. Test-per-scan BIST follows the usual procedure of shifting the test pattern in the scan chain before test application. Test-per-clock BIST loads the test pattern in the scan chain and captures the response in the same clock cycle (it is also known as at-speed test as it executes test at the speed of system clock frequency and results in much smaller test application time than test-per-scan BIST). Test-per-clock BIST has higher area cost than test-per-scan BIST due to the additional logic for the test application and test response analysis in the same clock cycle [Girard et al., 2009].

1.7 Thesis Organization

Chapter 2 - Literature Survey

This chapter presents a coherent overview of recently reported research in testing strategies for multi-voltage designs including defect modeling, test generation and DFT solutions. The chapter also outlines a number of important research problems that are addressed in this thesis to develop high-quality and cost-effective test solutions for multi-V_{dd} designs.

Chapter 3 - Test cost reduction using Test Points

This chapter presents a technique to achieve a cost-effective test method based on Test Point Insertion (TPI) to test bridge defects in multi-V_{dd} designs. It is motivated by experimental results that the majority of circuits (8 out of 12) require testing at more than one voltage setting to achieve 100% bridge fault coverage [Ingelsson, 2009], which means that the ATE (Automatic Test Equipment) will have to switch between different voltage settings to apply the test. Switching between different V_{dd} settings during test is not a trivial task, and therefore a large number of V_{dd} settings required during test can have a detrimental impact on the overall cost of test. Consequently it would be desirable to keep the number of V_{dd} settings required during test to a minimum. Chapter 3 presents a technique to reduce the number of test V_{dd} settings (and therefore test cost) without compromising the fault coverage of the original test.

Chapter 4 - Test cost reduction using Gate Sizing

This chapter presents an improved technique over that presented in Chapter 3 for reducing the number of test V_{dd} settings in multi-V_{dd} designs with bridge defects. It targets resistive bridges that cause faulty logic behaviour to appear at a non-desired test V_{dd} settings and uses Gate Sizing (GS) to expose the same physical resistance at the lowest (preferred) test V_{dd}. The number of test voltages is then reduced, minimizing test cost. This chapter shows that it is possible to achieve 100% fault coverage using a single test V_{dd} setting unlike the case with TPI (Chapter 3). This chapter also evaluates the timing, area and power cost of the proposed GS technique and comparison with the TPI technique shows that the proposed gate sizing technique achieves the same objective at lower cost in terms of timing, area and power.

Chapter 5 - Bridge Defect Diagnosis

This chapter presents a study on diagnosing resistive bridge defects in the context of multi-V_{dd} designs. There is no reported work in the literature on diagnosing multiple-voltage enabled ICs and the aim of this chapter to propose a technique for diagnosing bridge defects in such ICs. Using synthesized ISCAS benchmarks, with realistic extracted bridges and a parametric fault model, the chapter investigates the impact of varying supply voltage on the accuracy of diagnosis and demonstrates how the additional voltage settings can be leveraged to improve the diagnosis resolution through a novel multi-voltage diagnosis algorithm. It also shows the most useful voltage settings to reduce the diagnosis cost by eliminating tests at certain voltage setting using the proposed multi-voltage diagnosis approach, thereby achieving high diagnosis accuracy at reduced cost.

Chapter 6 - Conclusions and Future Work

This chapter summarizes the contributions presented in this thesis and outlines a number of research problems that merit further investigation to achieve efficient and cost-effective manufacturing test of future ICs.

1.8 Contributions

The contributions of the research work presented in this thesis have been published as follows:

Book Chapter

1. **Khursheed, S.**, Al-Hashimi, B. M., *Test Strategies for Multiple-Voltage Designs*, Springer book "Power-Aware Testing and Test Strategies for Low Power Devices", Patrick Girard, Nicola Nicolici, and Xiaoqing Wen (Editors), Nov. 2009. **Invited Monograph**

Journal Publications

2. **Khursheed, S.**, Ingelsson, U., Rosinger, P., Al-Hashimi, B. M., and Harrod, P., *Bridging Fault Test Method with Adaptive Power Management Awareness*, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Vol. 27, No. 6, June, 2008.
3. **Khursheed, S.**, Al-Hashimi, B. M., Reddy, S. M., Harrod, P., *Diagnosis of Multiple-Voltage design with bridge defect*, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Vol. 28, No. 3, March 2009.
4. Ingelsson, U., Al-Hashimi, B. M., **Khursheed, S.**, Reddy, S. M., Harrod, P., *Process Variation-Aware Test for Resistive Bridges*, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Vol. 28, No. 8, Aug, 2009.

5. **Khursheed, S.**, Al-Hashimi, B. M., Chakrabarty, K., Harrod, P., *Gate-Sizing-Based Single V_{dd} Test for Bridge Defects in Multi-Voltage Designs*, Submitted to IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, on 28th Oct. 2009 and revised on 4th Feb. 2010.

Conference Publications

6. **Khursheed, S.**, Al-Hashimi, B. M., Harrod, P., *Test Cost Reduction for Multiple-Voltage Designs with Bridge Defects through Gate-Sizing*, Design Automation and Test in Europe (DATE), 20th to 24th April, 2009, Nice, France.

7. **Khursheed, S.**, Rosinger, P., Al-Hashimi, B., Reddy, S. and Harrod, P., *Bridge Defect Diagnosis for Multiple-Voltage Design*, Proceedings IEEE European Test Symposium, 25th to 29th May' 08, Lago Maggiore, Italy. **Nominated for Best Paper Award**

8. Ingelsson, U., Rosinger, P., **Khursheed, S.**, Al-Hashimi, B. M., and Harrod, P., *Resistive bridging faults DFT with adaptive power management awareness*, Proceedings IEEE Asisn Test Symposium (ATS), pages 101-106, 8th to 11th Oct, 07.

Chapter 2

Literature Survey

Minimizing power consumption through the use of low power design techniques has been an active research area for nearly two decades, motivated by the portable and hand-held devices application market. The operating voltages needed for such designs are generated either through dedicated multiple power supplies on chip [Hamada et al., 1998] or through adaptive voltage scaling circuitry consisting of DC-DC converters and voltage controlled oscillators [Lee and Sakurai, 2000]. These techniques operate gates or circuits not on the critical path of a design at lower operating voltage than those on the critical path thereby achieving low power without compromising performance. Commercial CAD tools support multi-$\mathrm{V_{dd}}$ design approach (Synopsys $galaxy^{TM}$) and for that reason it is normally employed in designs where power consumption is a key requirement. This chapter addresses the following general question, "Can existing DFT techniques be used to test multi-$\mathrm{V_{dd}}$ designs?" The simple answer is yes and to ensure high defect coverage it is necessary to repeat the test at all operating voltages of the design since some defects may show $\mathrm{V_{dd}}$ dependency as demonstrated in Section 2.1.2. This may not be viable in designs where cost is of great importance as the case with hand-held devices market. Recently researchers have started to develop specific test solutions to multi-$\mathrm{V_{dd}}$ designs where the aim is to improve defect coverage without the need to repeat the test at all operating voltages of the design. Testing multi-$\mathrm{V_{dd}}$ designs is an orthogonal problem to Very Low Voltage (VLV) testing [Hao and McCluskey, 1993], which was proposed over a decade ago to improve reliability. It was shown that testing between $2V_t$ and $2.5V_t$, where V_t is the transistor threshold voltage, achieves high fault coverage for resistive bridges. The differentiation is that in multi-$\mathrm{V_{dd}}$ designs there are a number of operating Vdds, in practice up to four, and the aim of multi-$\mathrm{V_{dd}}$ test is to determine the minimum number of voltage settings necessary to ensure the highest level of fault coverage.

This chapter outlines recent work related to the presented research work undertaken in this thesis for two major types of defects: resistive bridge and resistive open in the context of multi-V_{dd} designs. A non-resistive defect (e.g., a short) between an interconnect line and power supply (V_{dd}) or ground rail (Gnd) can be modeled using a stuck at fault model, which represents permanent failure of the line in terms of stuck-at 1 (short with V_{dd}) or stuck-at 0 (short with Gnd) respectively. Such type of failures do not show V_{dd} dependent detectability[1] and therefore are not discussed in this chapter. Sections 2.1 and 2.2 discuss test techniques for resistive bridge and resistive open defects in the context of multi-V_{dd} designs. The DFT technique for devices employing multi-V_{dd} is discussed in Section 2.3, with the aim to achieve cost-effective test as well as reducing power dissipation during test. Section 2.4 provides the motivation for the research carried out in this thesis and outlines the objective of each research problem addressed in this thesis. Section 2.5 provides a brief summary of emerging and new test research problems at the time of compilation of this thesis, and finally, Section 2.6 concludes the chapter.

2.1 Test for Multi-Voltage Design: Bridge Defect

Resistive bridge represent a major class of defects for deep submicron (DSM) CMOS. It is due to an un-wanted metal connection between two lines of the circuit, which deviates the circuit from its ideal behavior. A typical resistive bridge is shown in Figure 2.1. A study on resistive bridge distribution is reported in [Rodriguez-Montanes et al., 1992] based on 14 wafers from different batches and production lines. The study shows that around 96% of bridges have a resistance value which is less than 1 kΩ. On the other hand, a physical defect between an interconnect line and power supply (V_{dd}) or ground rail (Gnd) is referred to as hard-short (bridge with 0 Ω resistance).

This section discusses modeling and test generation of resistive bridge for multi-V_{dd} designs. Section 2.1.1 describes the analog and digital behavior of resistive bridge at single voltage setting. This is further extended by showing V_{dd} dependency of resistive bridge in Section 2.1.2.

2.1.1 Resistive Bridge Behavior at Single V_{dd} Setting

The resistance of a bridge is a continuous parameter which is not known in advance. A recent approach based on interval algebra [Engelke et al., 2004], [Engelke et al., 2006b] allowed

[1] Stuck-at fault model does not capture physical complexities at the fault site and therefore more complex fault models have evolved to improve testability of the design. For a comprehensive discussion on evolution of fault models see [Delgado, 2008].

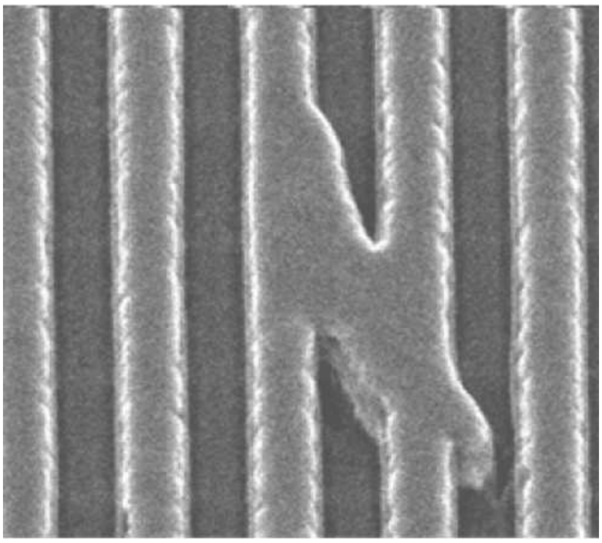


FIGURE 2.1: Resistive Bridge [Kundu et al., 2001].

treating the whole continuum of bridge resistance values $R_{sh} \in [0\ \Omega, \infty)$ by handling a finite number of discrete intervals. The key observation which enables this method is that a resistive bridge changes the voltages on the bridged lines from 0 V (logic-0) or V$_{\text{dd}}$ (logic-1) to some intermediate values, which will be different for different R_{sh} values. The logic behavior of the physical defect can be expressed in terms of the logic values perceived by the gate inputs driven by the bridged nets based on their specific input threshold voltage.

A typical bridge fault scenario is illustrated in Figure 2.2. D1 and D2 are the gates driving the bridged nets, while S1, S2, S3 and S4 are successor gates, i.e. gates having inputs driven by one of the bridged nets. The resistive bridge affects the logic behavior only when the two bridged nets are driven at opposite logic values. For example, consider the case when the output of D1 is driven high and the output of D2 is driven low. For illustration, we assume that the shown bridge R_{sh} affects only the output of D1, i.e., S1, S2 and S3 are affected by the resistive bridge. The dependence of the voltage level on the output of D1 (V_O) on the equivalent resistance of the physical bridge is shown in Figure 2.3. The deviation of V_O from the ideal voltage level (V$_{\text{dd}}$) is highest for small values of R_{sh} and decreases for larger values of R_{sh}. To translate this analog behavior into the digital domain, the input threshold voltage levels V_{th1}, V_{th2} and V_{th3} of the successor gates S1, S2 and S3 have been added to the V_O plot. For each value of the bridge resistance R_{sh}, the logic values at inputs I_1, I_2 and I_3 can be determined by comparing V_O with the input threshold voltage of the corresponding input. These values are shown in the second part of Figure 2.3. Crosses are used to mark the faulty logic values and ticks to mark the correct

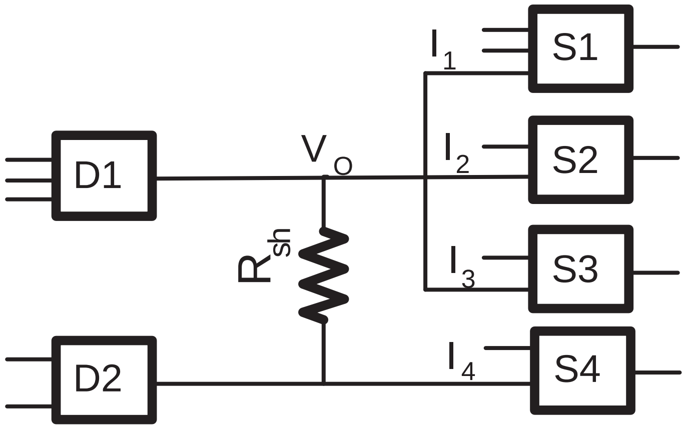


FIGURE 2.2: Example of a Resistive Bridge fault.

ones. It can be seen that, for bridge with $R_{sh} > R_3$, the logic behavior at the fault site is fault-free (all inputs interpret the correct value), while for bridge with R_{sh} between 0 and R_3, one or more of the successor inputs are interpreting a faulty logic value. The R_{sh} value corresponding to R_3 is normally referred to as "critical resistance" as it represents the crossing point between faulty and correct logic behavior. Methods for determining the critical resistance have been presented in several publications [Sar-Dessai and Walker, 1999], [Engelke et al., 2006b].

A number of bridge resistance intervals can be identified based on the corresponding logic behavior. For example, all bridges with $R_{sh} \in [0, R_1]$ exhibit the same faulty behavior in the digital domain (all successor inputs interpret faulty logic value). Similarly, for bridges with $R_{sh} \in [R_1, R_2]$, successor gates S2 and S3 interpret the faulty value, while S1 interprets the correct value. Finally, for bridges with $R_{sh} \in [R_2, R_3]$ only S3 interprets a faulty value while the other two successor gates interpret the correct logic value. Consequently, each interval $[R_i, R_{i+1}]$ corresponds to a distinct logic behavior occurring at the bridge fault site. The logic behavior at the fault site can be captured using a data structure further referred to as logic state configuration (LSC), which can be looked at as logic fault model [Khursheed et al., 2008]. This data structure used to capture resistive bridge fault is more complex than the one used for stuck-at fault model, as it holds the details of four important parameters of a bridge fault site. These four parameters include: boolean inputs to the driving gates, logic threshold of the driven gate inputs, voltage setting, and resistance interval covered. Boolean inputs to the driving gates (D1, D2 as in Figure 2.2) influence the voltage V_O on the bridged nets. This is because the boolean inputs to the driving gates switches the PMOS transistor(s) of the pull-up network (for gate driving high), and NMOS transistors of the pull-down network (for gate driving low) and the

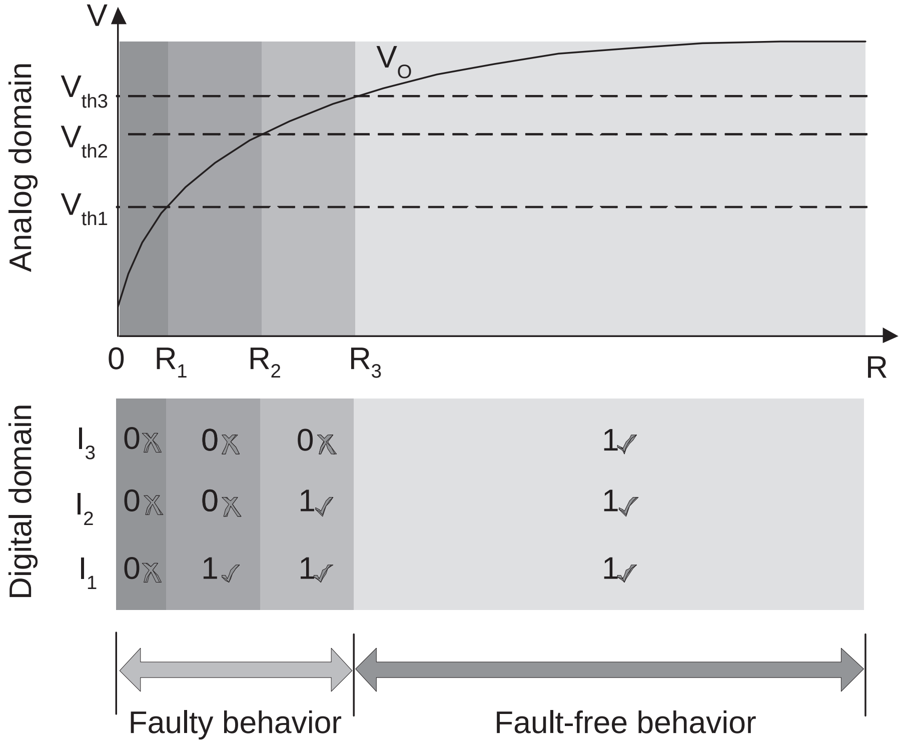


FIGURE 2.3: Behavior of a bridge fault at a single V_{dd} setting in analog and digital domains.

overall configuration of PMOS and NMOS transistors (including drive strength of each) influence the voltage V_O on the bridged nets. The next two parameters (logic threshold values and V_{dd} settings) are added in the LSC because logic threshold values of gates' inputs driven by the bridge varies across different voltage settings and affects the logic fault behavior[2]. Section 2.1.2 presents more details with illustrative example on the change of logic fault behavior with change in supply voltage. Finally, the resistance interval for example, $[0, R_1]$ that exhibit the same faulty behavior at the given inputs to the driving gates and V_{dd} settings is also stored in the LSC. For all experiments reported in this thesis, the resistance interval is calculated by using nominal process parameter values (of transistors) without considering corner cases, this assumption is also used in other recently published dissertations on resistive bridge defects [Ingelsson, 2009, Engelke, 2009]. It reduces computation complexity and acts as a simplifying assumption, however the resistance interval may slightly vary at other process corners, as discussed in Section 6.2.

The union of the resistance intervals corresponding to detectable faults forms the Global Analogue Detectability Interval (G-ADI) [Engelke et al., 2006b]. Basically, G-ADI represents the

[2] The tool flow showing the mechanism to generate logic threshold values is presented in Appendix A.

TABLE 2.1: Test set targeting resistance intervals

Test Set	Detectable Intervals
T_1	$[0, R_1]$
T_2	$[R_1, R_2]$
T_3	$\emptyset$
T_4	$[R_2, R_3]$

entire range of detectable physical defects. Given a test set TS, the Covered Analogue Detectability Interval (C-ADI) represents the range of physical defects detected by TS. The C-ADI for a bridge defect is the union of one or more disjoint resistance intervals, the union of intervals corresponding to detectable faults [Renovell et al., 1996], [Engelke et al., 2004], [Engelke et al., 2006b], and [Engelke et al., 2006a]. For example, considering the bridge fault shown in Figure 2.2 and corresponding resistance intervals in Figure 2.3, assume the test vectors for each detectable interval are tabulated in Table 2.1. The table shows four test vectors along with the resistance interval covered by each of the test vectors. The C-ADI of test vectors T_1, T_2, and T_3 can be given by the union of corresponding resistance intervals, i.e., $[0, R_1] \cup [R_1, R_2] \cup \emptyset = [0, R_2]$. The G-ADI is the union of all detectable resistance intervals, i.e., $[0, R_1] \cup [R_1, R_2] \cup [R_2, R_3] = [0, R_3]$. The quality of a test set is estimated by measuring how much of the G-ADI has been covered by the C-ADI. When the C-ADI of test set TS is identical to the G-ADI of fault f, TS is said to achieve full fault coverage for f.

A number of studies have shown that the detectable resistance range of bridge defect increases with lowering the supply voltage [Hao and McCluskey, 1993, Zain Ali, 2009]. A study reported in [Mandava et al., 1999] was conducted on a bridge fault to determine the impact of resistance range detection at three different V_{dd} settings. For that purpose, various resistances were inserted at the bridge fault site and static[3] and path delay test were used for fault detection. The results are tabulated in Table 2.2, which can be used to highlight the following four findings: firstly, as the supply voltage is reduced, the detectable resistance range increases for both logic and delay test techniques; secondly, at a given voltage setting delay test is able to cover higher resistance range than covered by logic test; thirdly, by reducing test V_{dd} setting, logic test is able to cover some of the resistance range covered by delay test at higher V_{dd} setting; and finally, after a certain bridge resistance range, the bridge fault can not be detected by either logic or delay test techniques.

Several test generation methods for resistive bridge faults RBF have been proposed for a fixed supply voltage setting [Sar-Dessai and Walker, 1999], [Maeda and Kinoshita, 2000], [Shinogi

[3] Static test implies test pattern applied without timing consideration

TABLE 2.2: Effect of voltage scaling on detectability of resistive bridges [Mandava et al., 1999]

V_{dd} Settings	Resistance Range						
	500 Ω	600 Ω	800 Ω	850 Ω	900 Ω	950 Ω	1000 Ω
2.5V	LE *	LE	TE *	TE	TE	TE	TE
2.2V	LE	LE	LE	TE	TE	TE	TE
1.9V	LE	LE	LE	LE	LE	TE	TE

V_{dd} Settings	Resistance Range						
	1050 Ω	1100 Ω	1150 Ω	1200 Ω	1400 Ω	1660 Ω	1800 Ω
2.5V	TE	TE	TE	TE	FF *	FF	FF
2.2V	TE	TE	TE	TE	TE	FF	FF
1.9V	TE	TE	TE	TE	TE	TE	TE

* LE → Logic Error, TE → Timing Error, FF → Fault Free

et al., 2001], [Chen et al., 2005], and [Engelke et al., 2006a]. The method presented in [Maeda and Kinoshita, 2000] is to guarantee the application of all possible values at the bridge site without detailed electrical analysis. In [Chen et al., 2005], the effect of a bridge on a node with fanout is modeled as a multiple line stuck-at fault. The study in [Sar-Dessai and Walker, 1999], identifies only the largest resistance interval and determines the corresponding test pattern. In contrast to [Sar-Dessai and Walker, 1999], the sectioning approach from [Shinogi et al., 2001] considers all the sections (resistance intervals) $[R_i, R_{i+1}]$. For each section, the corresponding LSC (and associated faulty logical behavior) is identified. This avoids the need for dealing with the resistance intervals and improves the test quality compared with [Sar-Dessai and Walker, 1999], but the number of considered faults grows. In [Engelke et al., 2006a], the authors combined the advantages of the interval based [Sar-Dessai and Walker, 1999] and the sectioning approach [Shinogi et al., 2001] into a more efficient test generation procedure by targeting the section with the highest boundaries first. Interval based fault simulation is then used to identify all other sections covered by the test pattern.

Prior research has analyzed the effect of varying the supply voltage on the fault coverage using pseudo random tests [Engelke et al., 2004]. The reported experimental results show that the fault coverage of a given test can vary both ways when the supply voltage is lowered, because not all faults can be covered using a single V_{dd} setting during test. However [Engelke et al., 2004] suggests that applying the tests at a lower supply voltage in addition to the nominal can improve the fault coverage. This finding is further elaborated by Figure 2.4. It shows the number of defects and respective resistance values, which cannot be detected (test escapes) at V_{dd} = 0.8 V (which would be a preferred V_{dd} for a 1.2 V process according to [Renovell et al., 1996], [Engelke et al.,

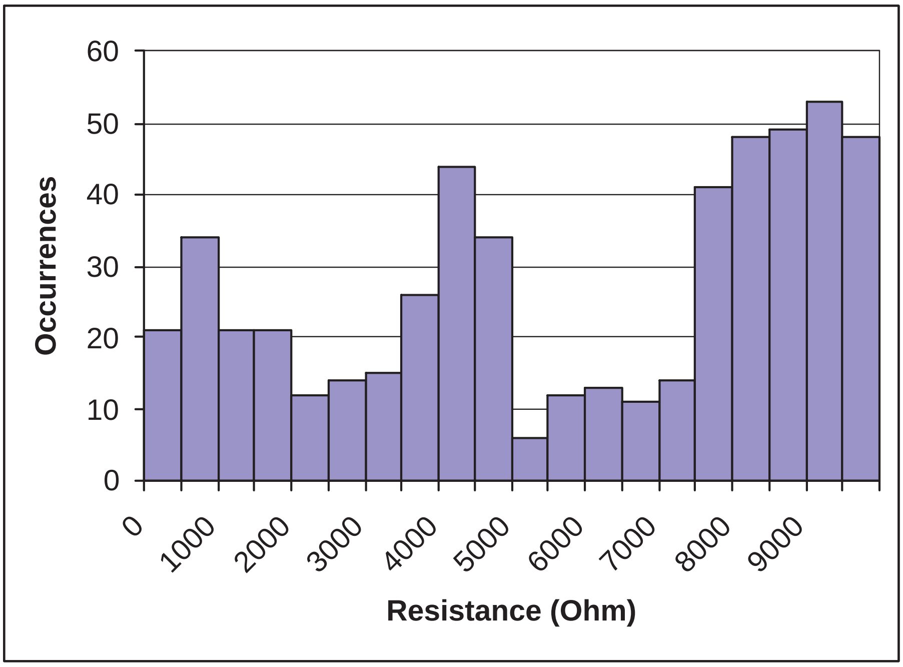


FIGURE 2.4: Resistance values that cannot be detected at the lowest V_{dd} setting [Ingelsson, 2009].

2004]). The test escapes at 0.8 V, as shown in Figure 2.4 is based on seven of the medium and large size ISCAS 85 and 89 benchmarks. The random spread of these defects across the resistance range suggests that to ensure high fault coverage it will be necessary to test at more than one V_{dd} setting for 100% fault coverage, as motivated by [Ingelsson, 2009]. In the next section we explain why it may be necessary to use more than one V_{dd} setting during test to ensure full bridge fault coverage for multi-V_{dd} designs.

2.1.2 Resistive Bridge Behavior at Multi-V_{dd} Settings

This section provides an analysis of the effect of varying supply voltage on bridge fault behavior. Figure 2.5 show the relation between the voltage on the output of gate D1 (Figure 2.2) and the bridge resistance for two different supply voltages Vdd_A and Vdd_B. The diagrams in Figure 2.6 show how the analog behavior at the fault site translates into the digital domain. In this example, three distinct logic faults LF1, LF2 and LF3 could be identified for each V_{dd} setting. However, because the voltage level on the output of D1 does not scale linearly with the input threshold voltages of S1, S2 and S3 when changing the supply voltage (this has been validated through SPICE simulations), the resistance intervals corresponding to LF1, LF2 and

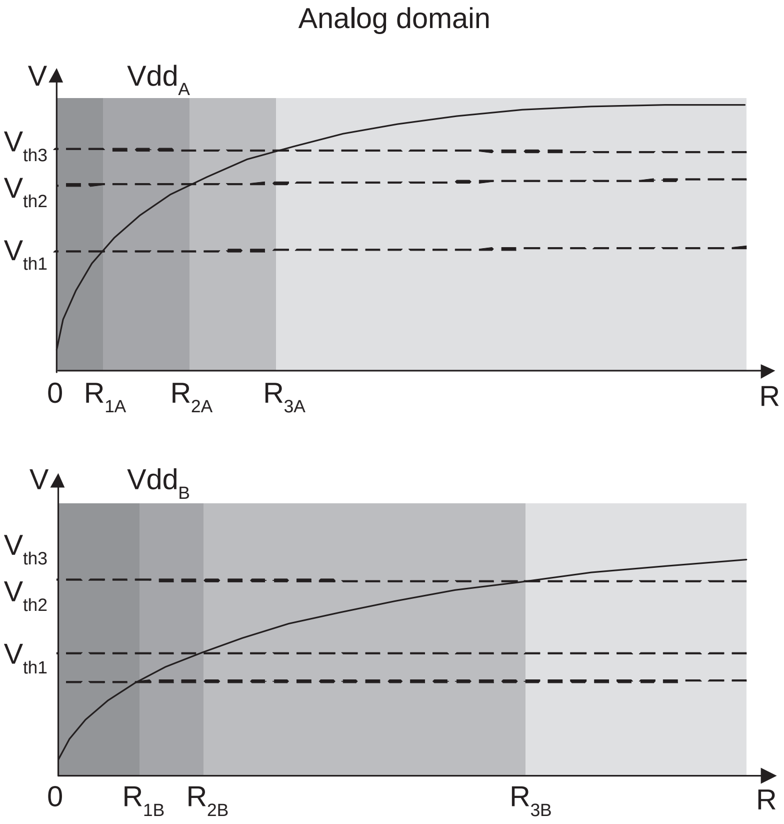


FIGURE 2.5: Effect of supply voltage on bridge fault behavior: Analog domain.

LF3 differ from one supply voltage setting to another. This means that a test pattern targeting a particular logic fault will detect different ranges of physical defects when applied at different supply voltage settings. For example, at Vdd$_A$, a test pattern targeting LF3 will detect bridge with $R_{sh} \in [R_{2A}, R_{3A}]$, while at Vdd$_B$ it will detect a much wider range of physical bridge ($R_{sh} \in [R_{2B}, R_{3B}]$). Analysing this from a different perspective, a bridge with $R_{sh} = R_{3B}$ will cause a logic fault at Vdd$_B$ but not at Vdd$_A$. To demonstrate the need for using multiple V$_{\rm dd}$ settings during test we use the following two scenarios. In Case 1 (Figure 2.7) all three logic faults LF1, LF2 and LF3 are non-redundant. Figure 2.7 shows the ranges of bridge resistance corresponding to faulty logic behavior for the two V$_{\rm dd}$ settings (basically the G-ADI sets corresponding to the two V$_{\rm dd}$ settings). Previous work on test generation for bridge faults [Engelke et al., 2006a] has used the concept of G-ADI assuming a fixed V$_{\rm dd}$ scenario. [Ingelsson et al., 2007] has extended the concept of G-ADI to capture the dependence of the bridge fault behavior on the supply voltage by defining the multi-V$_{\rm dd}$ G-ADI as the union of V$_{\rm dd}$ specific G-ADIs for a given design.

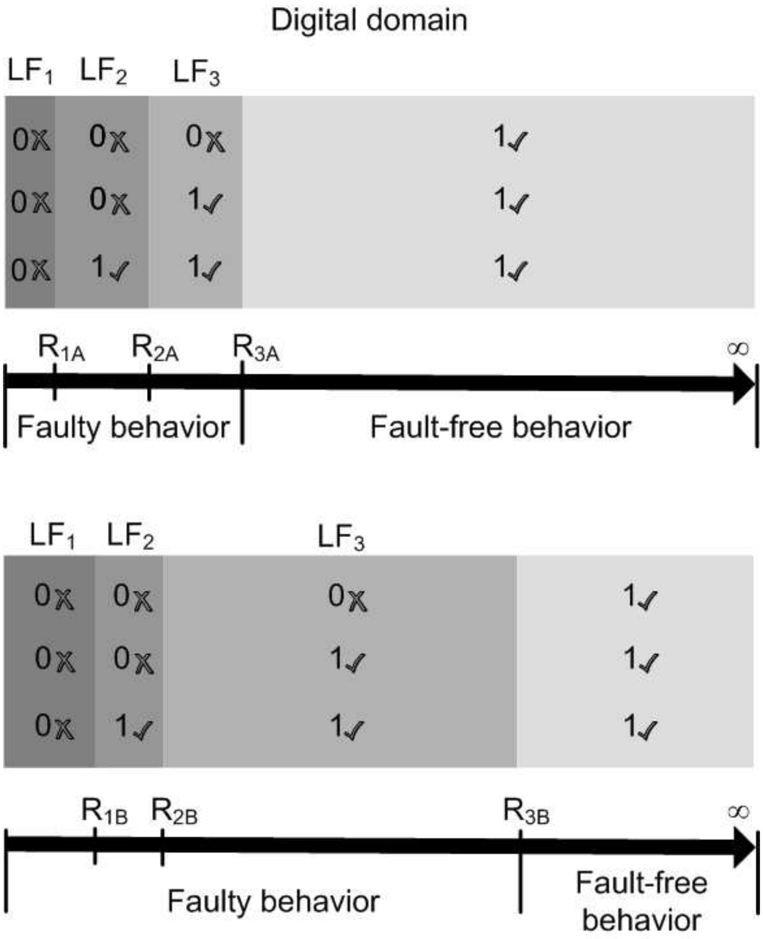


FIGURE 2.6: Effect of supply voltage on bridge fault behavior: Digital domain.

$G\text{-}ADI = \bigcup G\text{-}ADI(Vdd_i)$

The overall G-ADI consists of the union of the two V_{dd} specific G-ADI sets. It can be seen that $G\text{-}ADI(Vdd_A)$ represents about 45% of the overall G-ADI while $G\text{-}ADI(Vdd_B)$ fully covers the overall G-ADI. This means that a test set detecting LF1, LF2 and LF3 will achieve full bridge fault coverage when applied at Vdd_B. In Case 2 from Figure 2.7, only LF2 and LF3 are non-redundant, which means that there is no test pattern which can detect LF1. In this case, $G\text{-}ADI(Vdd_A)$ represents about 30% of the overall G-ADI while $G\text{-}ADI(Vdd_B)$ represents about 90% of the overall G-ADI. This means that full bridge fault coverage cannot be achieved using a single V_{dd} setting.

From this analysis it can be concluded that to achieve full G-ADI coverage in a variable V_{dd} system, it may be necessary to apply tests at several V_{dd} settings. Instead of repeating the same

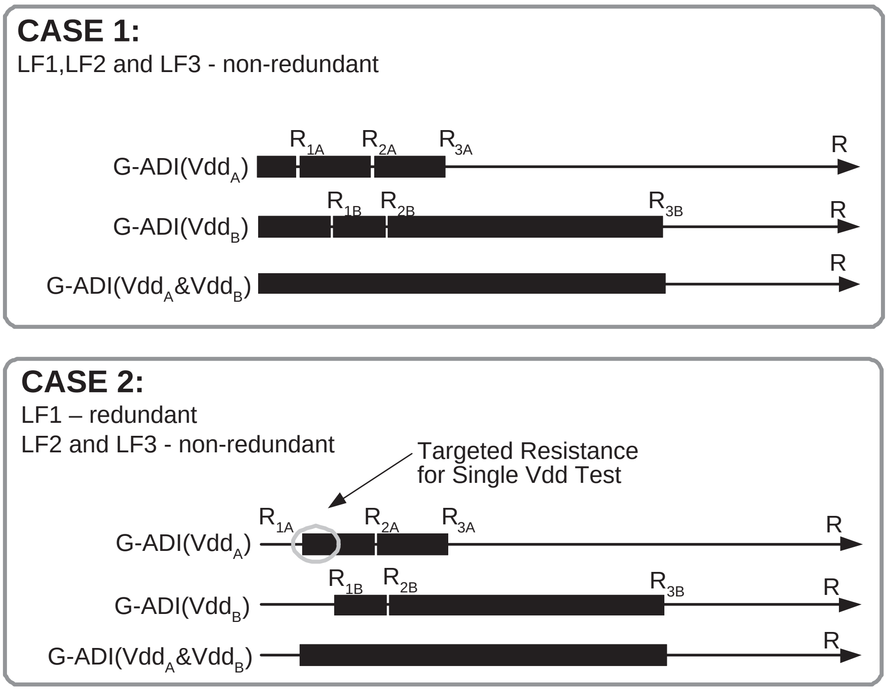


FIGURE 2.7: Effect of supply voltage on bridge fault behavior: Observable bridge resistance ranges.

test at all V_{dd} settings, which would lead to long testing times and consequently would increase the manufacturing cost, it would be desirable to be able to determine for each V_{dd} settings only the test patterns which effectively contribute to the overall defect coverage.

It has been shown in [Engelke et al., 2004] that the fault coverage of a test set targeting resistive bridge faults RBF can vary with the supply voltage used during test. This means that, depending on the operating V_{dd} setting, a given RBF may or may not affect the correct operation of the design. Consequently, to ensure high fault coverage for a design that needs to operate at a number of different Vdds, it may be necessary to perform testing at more than one V_{dd} to detect faults which manifest themselves only at particular Vdds. A Multi-V_{dd} Test Generation (MVTG) methodology is presented in [Ingelsson, 2009], which computes a number of V_{dd} specific test sets to achieve 100% fault coverage. In [Ingelsson, 2009] experiments are conducted using ISCAS-85' and 89' benchmark designs and fault list is compiled using coupling capacitance between neighboring nodes, these are most likely to form a bridge. Three V_{dd} settings are used for the experiment, i.e., 0.8 V, 1.0 V and 1.2 V and the outcome is tabulated in Table 2.3. The first two columns show the benchmark designs along with the number of faults extracted for each design. In this experiment, Synopsys $TetraMAX^{TM}$ is used to generate a test set for each design, which is then fault simulated at 0.8 V (since higher resistive bridge fault coverage

TABLE 2.3: Results of using Synopsys TetraMAX and Multi-V_{dd} Test Generation (MVTG) as a combined test generation flow for RBF [Ingelsson, 2009].

		TMAX		MVTG top-up			
		0.8 V		0.8 V	1.0 V	1.2 V	Tot.
Design	# of RBF	DC	#tp	#tp	#tp	#tp	#tp
c1355	80	83	33	32			65
c1908	98	98	42	27			69
c2670	104	90	27	50			77
c3540	363	96	72	126	6	1	205
c7552	577	95	44	198		1	243
s838	34	88	17	17	2		36
s1488	435	96	82	82	2		166
s5378	305	95	60	123			183
s9234	223	89	48	92	2		142
s13207	358	95	60	89	5	1	155
s15850	943	98	56	144	4	5	209
s35932	1170	96	33	89	36	66	224

is achieved at a lower V_{dd}). The fault coverage achieved and the number of test patterns in the TetraMAX test-set are shown in the third main column of Table 2.3. Subsequently, MVTG [Ingelsson, 2009] is used to generate top-up tests, targeting bridges that are not fully covered by the TetraMAX test-set. It is therefore used to provide the remaining defect coverage up to 100%. The sizes of the test sets generated by the MVTG top-up run are given in the fourth column for each V_{dd} setting. Finally, the total test pattern count is shown in the last column of Table 2.3, marked as “Tot.”. From test flow point of view, it is therefore suggested to use MVTG [Ingelsson, 2009] as a post-processing step to cover resistance intervals that remains uncovered by commercial ATPG tools.

2.2 Test for Multi-Voltage Design: Open Defect

Section 2.1 considered test techniques for bridge defect, this section discusses test techniques for open defects, which is another dominant defect type commonly found in deep-submicron CMOS. It is due to unconnected nodes in a manufactured circuit that were connected in the original design and therefore deviates the circuit from ideal behavior. Open defects can be

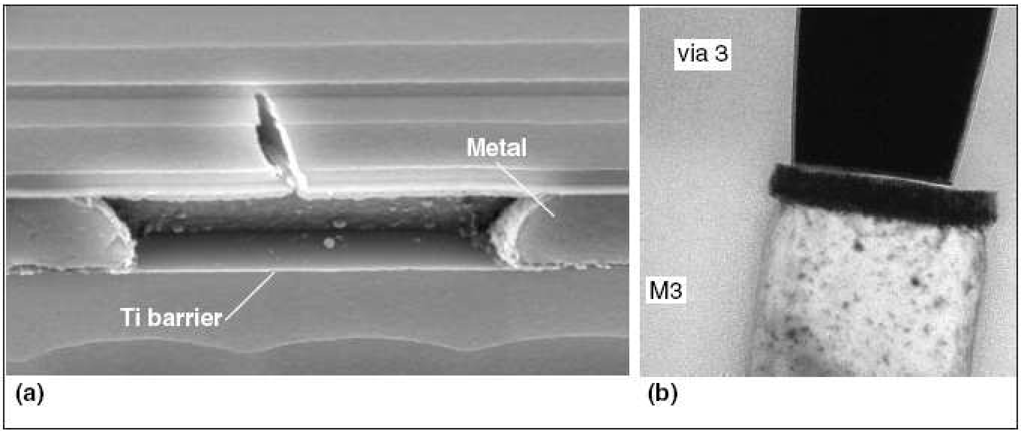


FIGURE 2.8: Resistive or Weak Open Defects: (a) Cross section of metal open line; and (b) a resistive via [Montanes et al., 2002].

classified as full or strong opens with resistance greater than 10 MΩ and resistive or weak open with resistance less than 10 MΩ [Montanes et al., 2002]. Full open cause logic failures that can be tested using static tests (test patterns applied without timing consideration). On the other hand, resistive open show timing dependent effects and therefore should be tested using delay tests. Figure 2.8 shows a cross-section of resistive open defect. In this section electrical characteristics of full open is discussed first, followed by resistive open.

2.2.1 Testing Full Open Defect

Figure 2.9 shows open defect distribution in six different metal layers corresponding to 7440 dies from 12 lots, manufactured in 180 nm CMOS process. As can be seen, the majority of open defects can be categorized as strong or full open defects. Similar trend is reported for contact or via open [Montanes et al., 2002]. The occurrence frequency of full-open defects is expected to increase in future technologies [Sreedhar et al., 2008], [Arumi et al., 2008a]. Two fault models are available in literature for modeling full-open defects, which can be categorized as capacitance based full-open fault model [Henderson et al., 1991], [Johnson, 1994], [Choudhury and Sangiovanni-Vincentelli, 1995], [Rafiq et al., 1998] and leakage-aware full-open fault model [Lo et al., 1997], [Guindi and Najm, 2003], [Sreedhar et al., 2008], [Arumi et al., 2008a]. Several recent studies have used capacitance based models [Gomez et al., 2005], [Zou et al., 2006], [Rodriguez-Montanes et al., 2007], [Spinner et al., 2008], [Arumi et al., 2008b] for testing full-open defects, which uses the following electrical characteristics: 1) the capacitance between floating line (disconnected from the driver node) and its neighboring line(s), 2) the parasitic capacitance due to transistors (PMOS and NMOS connected to floating line) driven by

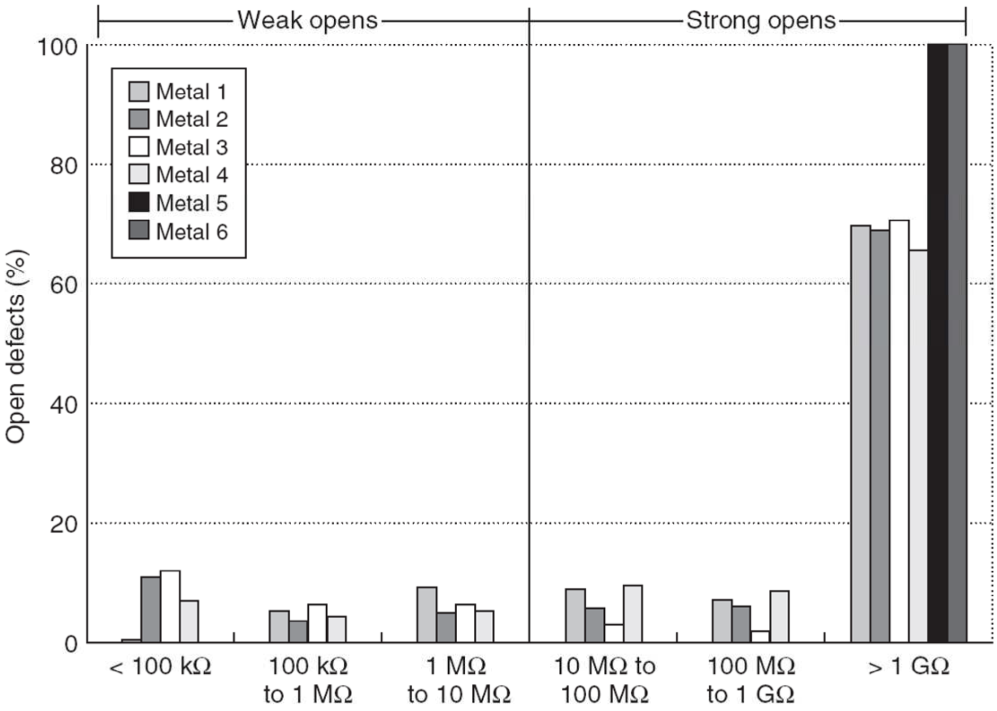


FIGURE 2.9: Distribution of metal open resistances [Montanes et al., 2002].

the floating net, and 3) the trapped charge on the floating net. If F represents a floating net that is disconnected from its driver, then voltage V_F is given by [Zou et al., 2006], and [Ingelsson, 2009]:

$$V_F = \frac{C_{High}}{C_{High} + C_{Low}} V_{dd} + \frac{Q_{trap}}{C_{Gnd}} \tag{2.1}$$

where, V_F is voltage on the floating net, C_{High} and C_{Low} is capacitance due to neighboring lines driving high and low respectively (including capacitance due to V_{dd} and Gnd), V_{dd} is the supply voltage, $\frac{Q_{trap}}{C_{Gnd}}$ represents the trapped charge on the floating net. From (2.1), it can be noticed that for detecting full-open defects, V_F can be induced such that voltage on the floating net is higher than the logic threshold L_{th} voltage of the gate input, i.e., $V_F > L_{th}$, thereby exciting a stuck-at 1 fault. Voltage on the floating net can be induced by using test patterns that result in setting the neighboring nets to desired logic value, thereby increasing the fraction $\frac{C_{High}}{C_{High}+C_{Low}}$, as shown in (2.1). Similarly a stuck-at 0 fault can be induced on the floating net. The fault effect can then be propagated to any of the primary outputs for detection [Zou et al., 2006].

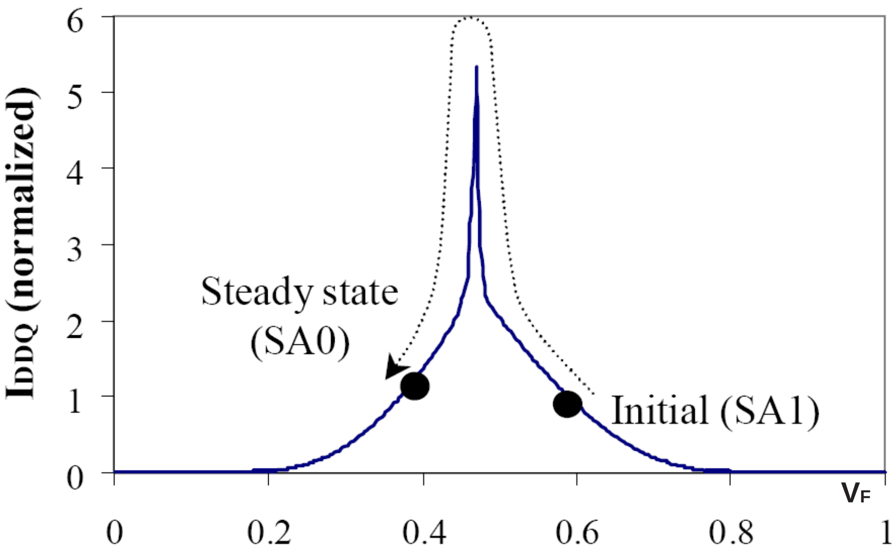


FIGURE 2.10: Change in logic value due to gate tunneling leakage [Arumi et al., 2008a].

In nanometer CMOS ($\leq$ 90 nm), since the thickness of gate oxide is few tens of $\mathring{A}$, it does not act as a strong insulator. This results in higher gate-tunneling leakage current in comparison to previous technologies [Sreedhar et al., 2008], [Arumi et al., 2008a], [Ingelsson, 2009], and therefore affects the voltage on the floating net causing full-open defect. A floating net connected to a gate has a bi-stable input state [Sreedhar et al., 2008], [Arumi et al., 2008a]. In [Sreedhar et al., 2008] an inverter synthesized using 45 nm technology was simulated with a floating input and the change in input voltage was observed. It was found that the voltage on the floating net increased from 0V to 0.17 V (due to gate leakage through the PMOS, as inverter output goes to logic high) and the input voltage reduced from 0.8 V to 0.58 V (due to gate leakage through the NMOS, as inverter output goes to logic low). Furthermore, in [Arumi et al., 2008a] an experiment is conducted using 0.18 μm technology with an open defect. It is shown that an interconnect open initially set to behave as stuck-at 1 (using (2.1) and procedure described above to set a particular logic value on an interconnect) changes to stuck-at 0 in approx. 2 seconds, due to gate tunneling leakage currents. Voltage behavior of the floating net is shown in Figure 2.10. It is therefore concluded that for nanometer CMOS, gate tunneling leakage is a dominant player in setting the voltage on the floating net and the final steady state value is independent of the initial state. Furthermore, it is predicted that the time period to reach the steady state will reduce in future technologies and will be in the order of hundreds of μs.

The V_{dd} dependent detectability of full-open defects is investigated in a study presented in [Ingelsson, 2009] using static test and leakage-aware fault model. The experiments utilize ISCAS

TABLE 2.4: Fault coverage at three V_{dd} settings using 1000 pseudo random test patterns [Ingelsson, 2009].

Design	# of Full-opens	Fault Coverage at three Voltage Settings		
		1.2V	**1.0V**	**0.8V**
c432	123	98.4	98.4	98.4
c499	197	99.5	99.5	99.5
c880	222	99.6	99.6	99.6
c1355	236	97.4	97	97.9
c1908	214	99.1	99.5	99.1
c2670	472	86.7	86.7	86.7
c3540	468	99.3	98.9	99.1
c5315	623	100	100	100
c7552	887	97.1	97.3	97.3
s641	107	98.1	98.1	98.1
s1488	290	99.3	99.3	99.7
s5378	634	96.5	97.3	97.2
s9234	483	92.5	90.5	90.3

benchmarks that are tested at three V_{dd} settings. The results are presented in Table 2.4 for leakage-aware fault model. As can be seen, the fault coverage does not vary across V_{dd} settings, and it has very small impact for some designs, for example, in case of c1355 the fault coverage varies by less than 1% across three V_{dd} settings. This is because the fault detection is independent of the logic value at the input of the gate driving the floating net and irrespective of the logic value at the inputs of gate driving the net, a test may still detect the fault. This provides extra flexibility in terms of fault detection for this class of defect. Therefore, it is concluded that V_{dd} setting does not affect detectability of full-open defects and any test V_{dd} setting can be used for testing full-open defects. It also shows results for capacitive fault model that does not take leakage current into account, and draws similar conclusions as for leakage-aware fault model.

2.2.2 Testing Resistive Open Defect

This section summarizes recent research on test techniques for resistive interconnect open defect and the impact of voltage setting on their testability. Resistive open can be modeled as a resistor between two unconnected nodes, since it shows small inductive/capacitive component, which can be neglected for simplicity as used in [Kruseman and Heiligers, 2006], and [Zain Ali and

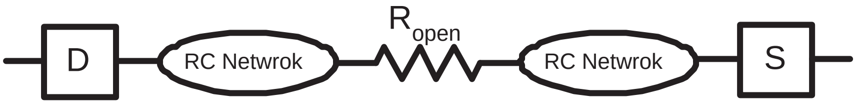


FIGURE 2.11: Circuit Model of Resistive Open Defect.

Zwolinski, 2006]. Figure 2.11 shows a typical resistive open fault model, where "D" and "S" represent the driver and successor gate respectively.

Resistive open shows timing dependent effects and therefore should be tested using delay tests. Delay fault testing is used to catch defects that create additional than expected delay and thereby cause a malfunction of the IC [Kruseman and Heiligers, 2006]. Using delay fault testing, a defect is detectable only when it causes longer delay than that of the longest path in a fault free design. It was shown in [Kruseman et al., 2004] that majority of tested paths show less than one-third delay in comparison to that of the longest path. Therefore a defect in any of these shorter paths can only be detected if it causes higher delay than that of the longest path in the design.

In [Kruseman and Heiligers, 2006] the optimal test conditions for testing resistive open is analyzed for non-speed-binned ICs, which are designed to meet timing under worst process and working conditions and typically have a logic depth of 30-70 gates. It is argued that for designs operating at few hundred MHz, one can expect to detect defects with resistance of 100 kΩ or more, while delay caused by smaller resistance defects are of the order of gate delays and does not cause additional delay even if they occur at the longest path. The paper analyses two major sources of open defects, i.e., incompletely filled vias and partial breaks in the poly of the transistor (due to salicidation). Furthermore, it is argued that resistive open shows better detectability on silicon at elevated V_{dd} settings. This phenomenon is elaborated using two examples, shown in Figure 2.12 and Figure 2.13 and discussed next. Figure 2.12 shows the delay caused by two different resistive opens (due to 1 MΩ and 3 MΩ) while considering these defects in the longest path and using different supply voltage settings (1.8 V being nominal supply voltage). The figure also shows the delay of the longest path in fault free design (using solid gray line) and at various voltage settings. As can be seen, the defect induced extra delay added to the expected delay is highest at elevated supply voltage (V_{dd} = 2.0 V) for both resistive open defects. Also, as expected, higher delay is observed at 3 MΩ than 1 MΩ. Figure 2.13 shows the effect of resistive open in a shorter path, with half the delay as the longest path in a fault-free design. Defects with same resistance values as Figure 2.12 are inserted in the shorter path, and the delay is compared with that of the longest path (shown by solid gray line). As can be seen, delay due to 1 MΩ resistance show marginal detectability only at elevated V_{dd} setting (2.0 V), by causing higher delay than that of the longest path. It becomes undetectable at lower V_{dd} settings, as it shows

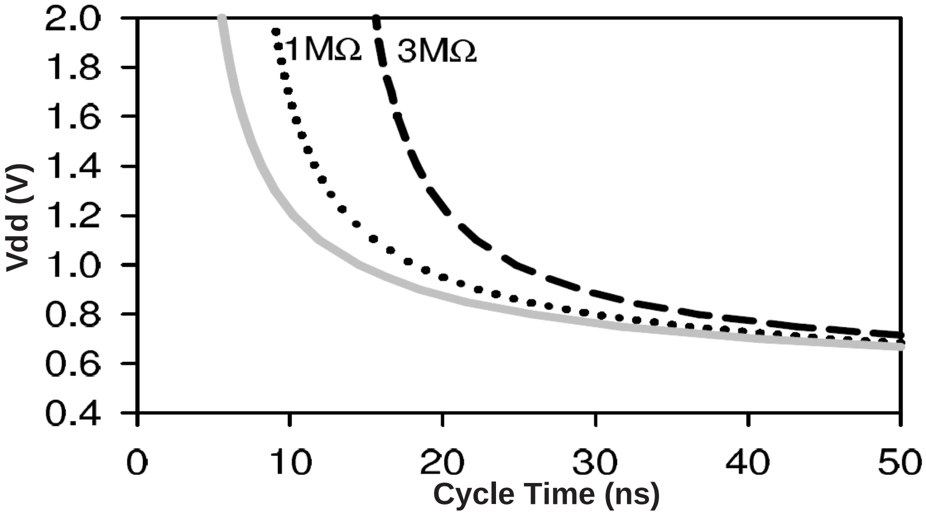


FIGURE 2.12: Comparison of path delays due to resistive open defect in the longest path at different supply voltage settings. Solid gray line shows the fault free design, while dotted and dashed lines show path delays using 1 MΩ and 3 MΩ in the longest path [Kruseman and Heiligers, 2006].

lesser delay than that of the longest path. On the other hand, 3 MΩ defect resistance is best detectable at elevated V_{dd} (2.0 V) and becomes undetectable as V_{dd} setting is reduced further from 0.9 V. The behavior shown by these two examples (illustrated by Figure 2.12 and Figure 2.13) is commonly observed on silicon and is generalized using Figure 2.14. As can be seen from Figure 2.14, resistive open in general show better detectability at elevated V_{dd} setting and becomes undetectable at reduced V_{dd}. Finally [Kruseman and Heiligers, 2006] shows some cases where resistive open defects are better detectable at reduced V_{dd} setting.

[Zain Ali and Zwolinski, 2006] has also studied delay behavior for devices operating at multi-V_{dd} settings. Two types of defects are examined, i.e., transmission gate open and resistive open. Experiments are conducted using 0.35 μm using five (3.3, 3.0, 2.7, 2.5 and 2.0 V) discrete voltage settings on a 4 level carry save adder (shown in Figure 2.15). Each unit of carry save adder (for e.g., CSA-01) is made up of 5 transmission gates. The impact of transmission gate open is studied first, by inserting two NMOS open defects (one at a time) as shown in Figure 2.15 (marked as "Fault A" and "Fault B"). The fault site and signal propagation path of inserted defects is shown in Table 2.5. Gate Delay Ratio (GDR) and Path Delay Ratio (PDR)[4] is calculated and results indicate that higher gate/path delay ratio is observed as V_{dd} setting is reduced and the two faults (transmission gate open) behaves as stuck-at fault (SF) at lower V_{dd} settings. As

[4] In [Zain Ali and Zwolinski, 2006] GDR (PDR) is calculated as a delay ratio between faulty and fault-free signal propagating gate (path) of a design.

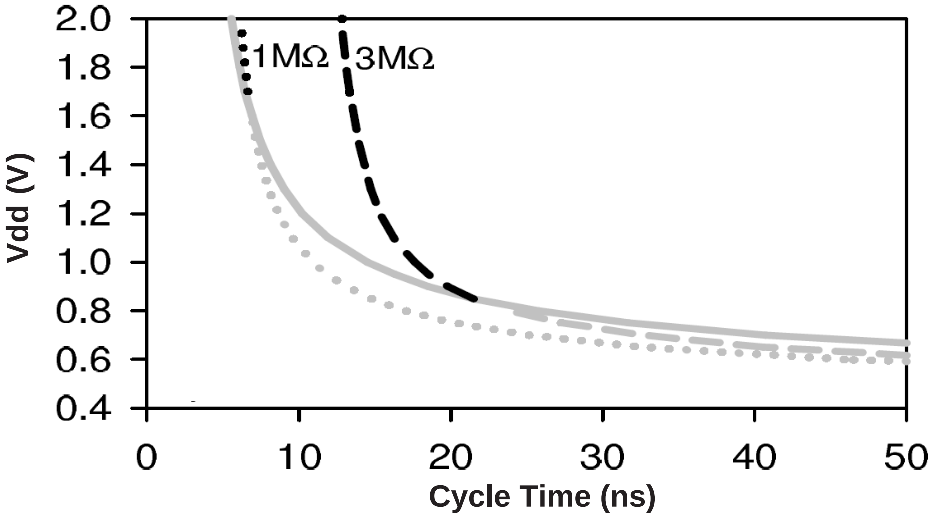


FIGURE 2.13: Comparison of path delays due to resistive open defect in a short path at different supply voltage settings. The longest path is shown by a solid gray line (for the fault free design), while dotted and dashed lines show path delays using 1 MΩ and 3 MΩ resistances in a shorter path [Kruseman and Heiligers, 2006].

expected, increased GDRs for both the faults result in higher PDRs at respective paths as well. Similar observations were reported in [Chang and McCluskey, 1996a] using 0.6 μm and 0.8 μm technology and similar experimental setup. Study reported in [Chang and McCluskey, 1996a] has suggested using $2V_t$ to $2.5V_t$ (Very Low Voltage (VLV) testing) for detecting defects due to transmission gate open, threshold voltage shift and diminished-drive strength. This explains the SF behavior of transmission gate open at reduced V_{dd} settings.

TABLE 2.5: Signal Propagating Path for Faults A and B [Zain Ali and Zwolinski, 2006].

Fault Site		Signal Propagating Path
A	CSA-11 NMOS Open	CSA-01(A) → CSA-11(B) → CSA-21(B) → CSA-32(Cin) → CSA-32(Cout)
B	CSA-22 NMOS Open	CSA-01(A) → CSA-11(B) → CSA-22(Cin) → CSA-32(B) → CSA-32(Cout)

The impact of interconnect resistive open is also studied in [Zain Ali and Zwolinski, 2006] by inserting two defects separately in the circuit, marked as “Fault C” and “Fault D” as shown in Figure 2.15. For this experiment, three different resistance values (25 KΩ, 250 KΩ and 1 MΩ) are used on both locations and results show that Path Delay Ratio (PDR) due to these two faults increases with higher V_{dd} setting. As expected, PDR is more prominent for 1 MΩ resistance at elevated V_{dd} setting than the other two resistance values. These findings show that interconnect

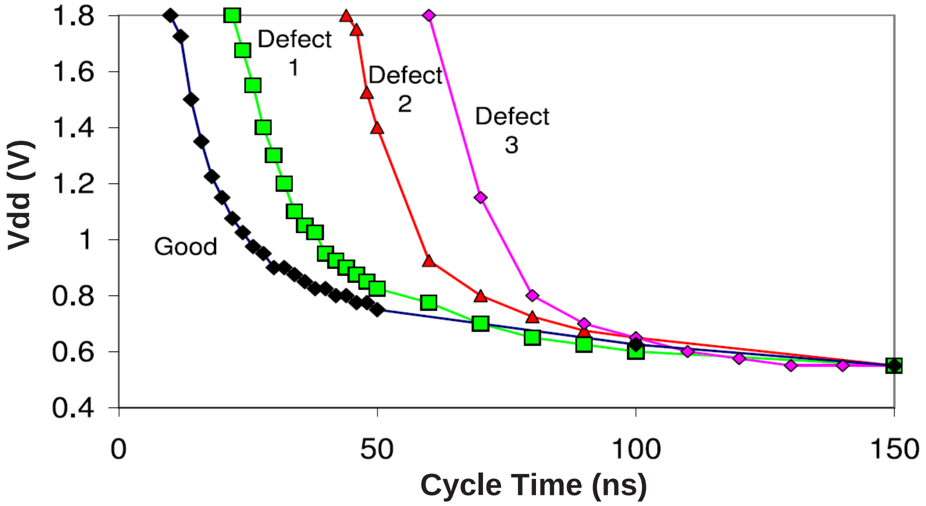


FIGURE 2.14: Delay behavior of fault-free design (marked as "Good") in comparison to delay defect behavior due to three different defects [Kruseman and Heiligers, 2006].

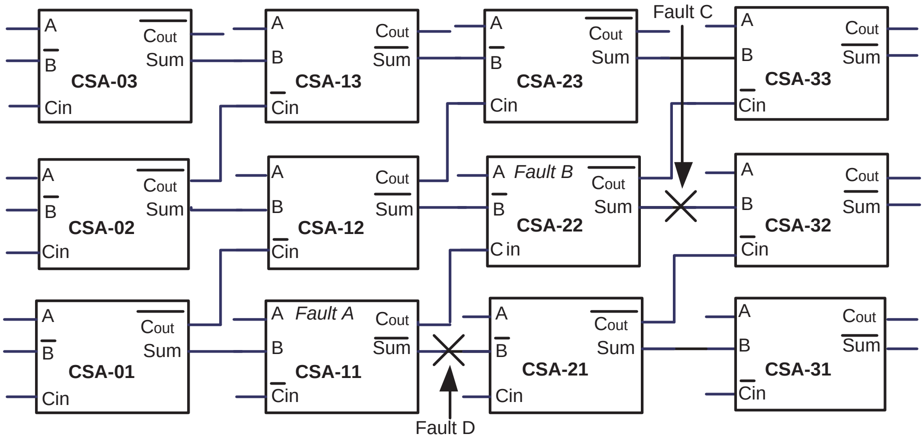


FIGURE 2.15: 4-Level Carry-Save Adder, each adder cell is made of five transmission gates [Zain Ali and Zwolinski, 2006].

resistive opens are better detectable at elevated V_{dd} setting by delay test techniques. On the other hand, transmission gate opens are better detectable at lower V_{dd} settings. The application of delay test at single V_{dd} setting reduces test cost by avoiding repetitive tests at other V_{dd} settings.

In brief, interconnect-open defects have attracted a significant research effort world-wide to reduce test cost without affecting the fault coverage – in the context of multi-V_{dd} designs. Recent studies have shown that full-open defects can be tested using static test techniques at any

V_{dd} setting, as they do not exhibit V_{dd} dependent detectability [Ingelsson, 2009]. On the other hand, resistive-open defects are better detectable at elevated V_{dd} setting using delay test techniques [Kruseman and Heiligers, 2006, Zain Ali, 2009]. For these reasons, interconnect-open defects are not further investigated in this thesis, as a candidate defect for reducing test cost.

2.3 DFT for Low Power Design

Sections one and two outlined test techniques for resistive bridge and resistive open for multiple-voltage designs. In this section, a summary of recent low cost scan techniques for reducing power dissipation during test mode is given [Nicolici and Al-Hashimi, 2003]. These techniques are developed for devices employing multiple-voltage settings.

2.3.1 Multi-Voltage Aware Scan

Designs that employ multiple voltage settings are divided into various voltage domains during physical placement of the design. Each voltage domain feeds various logic blocks and level shifters are used to communicate logic values across logic blocks operating under different voltage settings [Shi and Kapur, 2004]. The insertion of scan chains across logic block poses a challenge for scan chain ordering in multiple voltage designs due to two main reasons. Firstly, it is desirable to reduce the number of level shifters required to transmit voltage levels from one scan chain to another, placed across different voltage domains. Secondly, power consumption during test can be reduced by fewer voltage domain crossing by the scan cells.

These challenges are met by multi-voltage aware scan cell ordering [Colle et al., 2005]. The proposed methodology arranges scan cells based on respective voltage domains. This is achieved by scan cells ordering in such a way that scan cells operating under the same voltage levels are connected together. This in turn minimizes the number of level shifters that are otherwise required if scan cells are ordered without consideration of multi-voltage designs. Furthermore, it reduces power dissipation by minimizing signal transmission in fewer voltage domain crossing. Experiments are conducted using industrial design with 4 voltage domains and it is shown that multi-voltage aware scan chain ordering shows 93% reduction in the number of level shifters, in comparison to scan chain ordering technique, which connects physically closer scan cells without considering its operating voltage. The proposed scheme has been implemented in Synopsys EDA tools and the DFT flow is shown in Figure 2.16. As can be seen, DFT Compiler recognizes the voltage/power domains and clusters the scan chains within the respective domains.

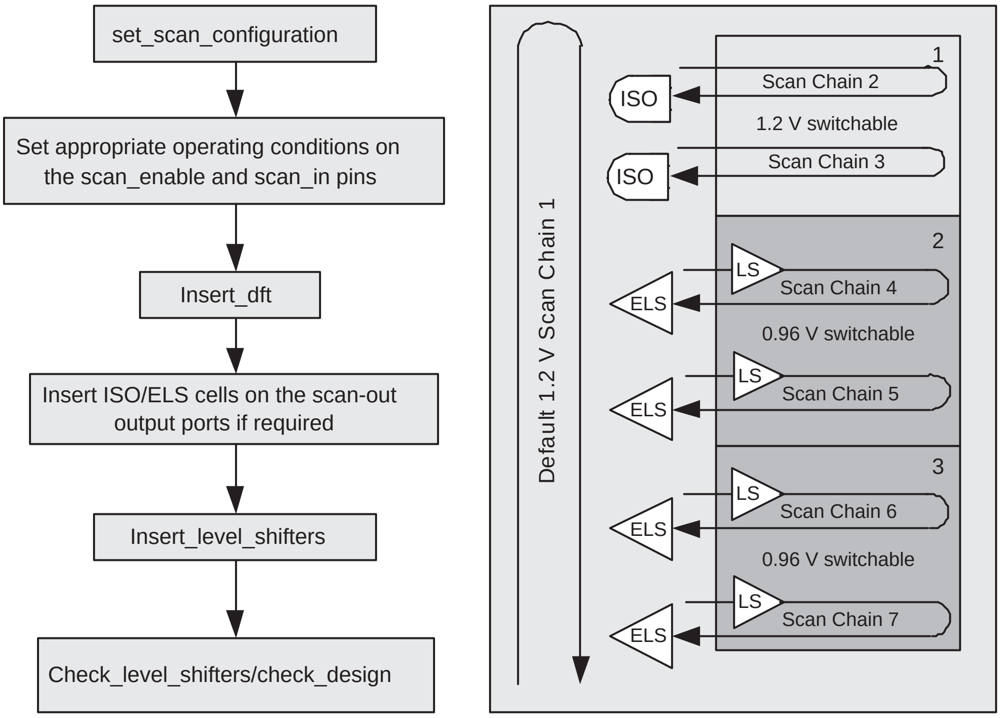


FIGURE 2.16: DFT Synthesis flow for Multi-V_{dd} design using Synopsys Design Compiler [Baby and Sarathi, 2008].

The number of level shifters in the design are minimized by disabling voltage/power domain mixing, which is managed by “set scan configuration”.

Recently a power-aware scan chain method is presented in [Chickermane et al., 2008] for multi-V_{dd} designs. The method is implemented using daisy-chaining scan approach to efficiently utilize expensive tester resources (bandwidth) and reduce test cost. The method avoids signal integrity issues during test by employing bypass multiplexers, which allows bypassing signals from power domains that are switched off during test. Daisy-chain implementation along with bypass multiplexers (1, 2, 3 and 4) and four different power domains (A, B, C and D) is shown in Figure 2.17. As can be seen, bypass multiplexers allow testing of specific power domains in multi-V_{dd} environment. As an example, in a particular power mode, where power domains C and D are ON, while A and B are OFF, muxes 1 and 2 goes in bypass mode, while 3 and 4 are in pass-thru mode. This forms a scan chain between SI, 3, 4 and SO. The bypass multiplexers are placed on always-on power domain. This approach is implemented in Cadence $Encounter^{TM}$ test tools.

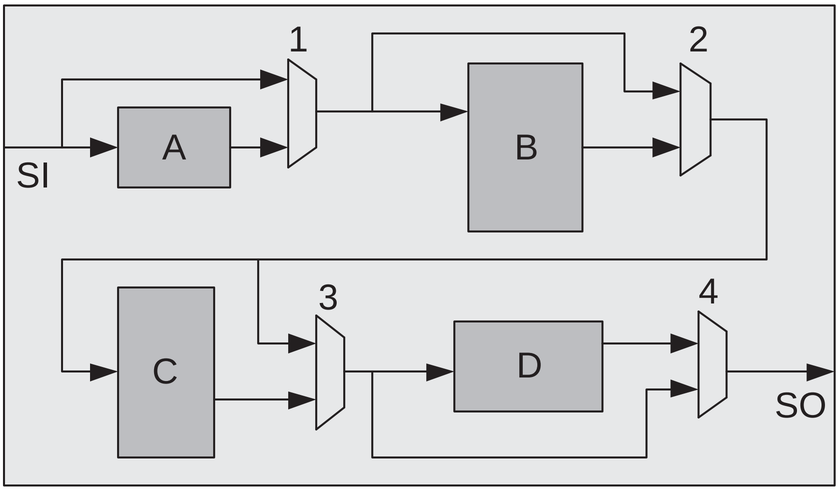


FIGURE 2.17: Power-Aware Daisy-chaining scan path [Chickermane et al., 2008].

2.3.2 Power-Managed Scan Using Adaptive Voltage Scaling

Reducing power dissipation during test has been an active area of research for nearly a decade and numerous techniques have been reported [Girard, 2002], [Bhunia et al., 2005]. Recently an interesting technique that reduces both dynamic and leakage power during test through the use of adaptive voltage scaling PMScan (Power Managed Scan) has been reported [Devanathan et al., 2007b]. The presented methodology is motivated by three factors. Firstly, it is known that dynamic power is proportional to V^2 [Weste and Eshraghian, 1994] and gate leakage power is proportional to V^4 [Krishnarnurthy et al., 2002], where V is the operating voltage of the device. Therefore, reduction in supply voltage can significantly reduce total power (dynamic plus leakage) during test. Secondly, infrastructure for adaptive voltage scaling is widely deployed in modern microprocessors to reduce power consumption during functional mode. Therefore, it is suggested in [Devanathan et al., 2007b] to reuse voltage scaling infrastructure to reduce implementation (due to physical design and area) overheads. Thirdly, scan shift frequency is usually much slower than the operational frequency of the device, therefore scan shift operation is ideal for voltage scaling during test[5]. Therefore PMScan proposes voltage scaling during test to provide a trade-off between test application time and test power. This is achieved by modifying voltage regulation circuitry (used for adaptive voltage scaling) such that scan shift operation

[5] Voltage scaling is widely used to reduce power consumption, while ensuring that timing requirements are met. It is therefore more effective for tasks that are less computationally intensive, i.e., tasks that can be completed at a slower speed.

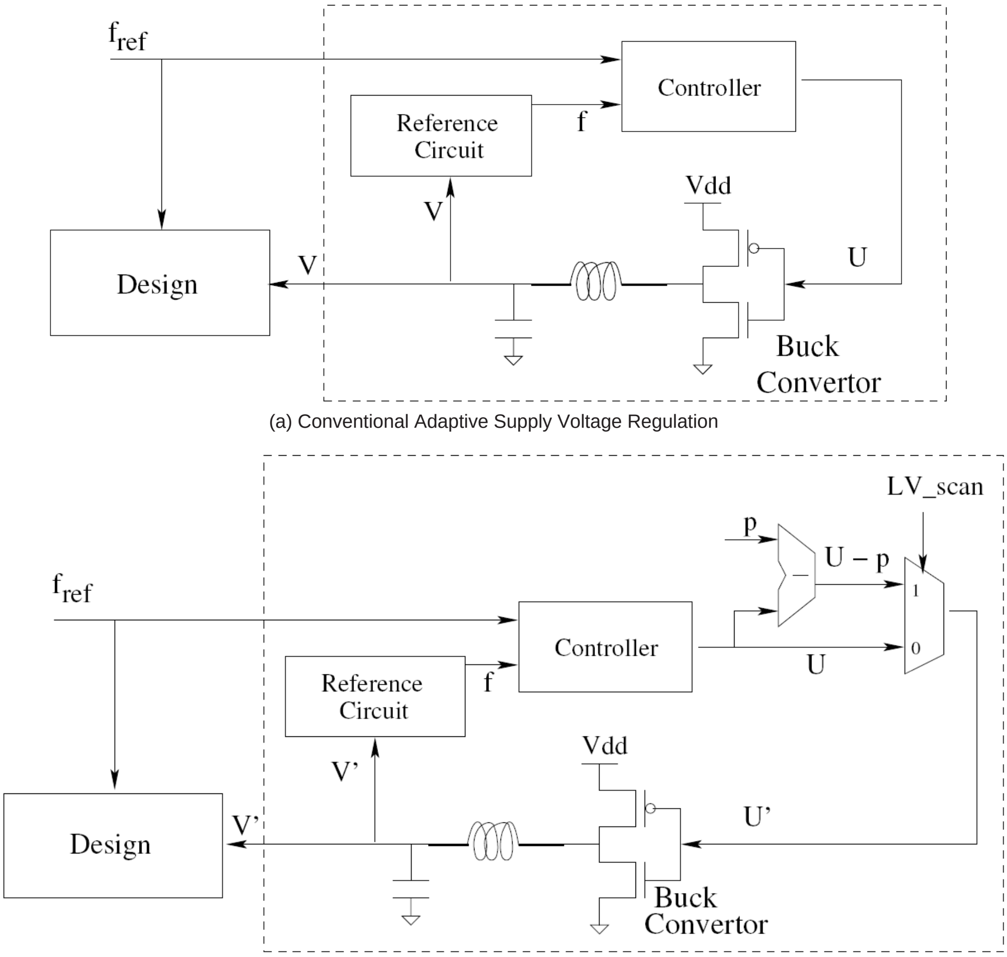
meets acceptable timing, while supply voltage during scan shift is reduced. The voltage regulation circuitry changes the supply voltage to nominal during scan capture mode to ensure at-speed testing.

The conventional voltage scaling circuitry and the one proposed in [Devanathan et al., 2007b] are shown in Figure 2.18. Figure 2.18(a) shows the conventional adaptive supply voltage circuitry showing the voltage regulation component in the dashed box. It uses feedback control and adjusts the supply voltage 'V' using a dc-dc converter such that the delay of the circuit fits in one clock cycle of the desired clock frequency $\mathrm{f_{ref}}$, which is usually generated using on-chip PLL. The reference circuit is made of a ring oscillator and determines the maximum delay of the design over process, voltage and temperature variations. It determines the maximum frequency 'f' corresponding to the voltage 'V' provided to it. In [Devanathan et al., 2007b] the conventional voltage regulation design is modified for voltage scaling during scan shift operation, as shown in Figure 2.18(b). It is designed such that when the signal LV_scan = 1, the supply voltage 'V' is lowered by 'p'. On the other hand when LV_scan = 0, the output 'U' is applied to the multiplexer as in conventional design. Refer to [Devanathan et al., 2007b] for more details on design of such regulator.

Experiments are conducted using 90 nm library with nominal 1.1 V supply voltage using Synopsys $PrimePower^{TM}$ for power analysis. The first experiment is conducted using seven different ISCAS 89 benchmarks using reduced $\mathrm{V_{dd}}$ (0.77 V) and at 25 MHz scan shift frequency. Average dynamic, peak dynamic and leakage power is compared between proposed PMScan technique with that of conventional scan (unaware of voltage scaling). It is shown that on average PMScan reduces average dynamic power by about 44%, peak dynamic by 42%, leakage power by 91% contributing to overall total power by 64% in comparison to conventional scan. Moreover, it is shown that these results can be further improved by 5%, by using NOR-Gating scheme [Girard, 2002][6] along with PMScan. The second experiment analyses test time and test power trade-off. It is conducted using an industrial design (with 9 million gates and 7 unwrapped cores), at three different voltage (1.1 V, 1.0 V and 0.77 V) and scan shift frequency (25 MHz, 75 MHz, and 125 MHz) settings. It is shown that for test application at 0.77 V and 125 MHz scan shift frequency, test time reduces by 80%, while total power increases by 16%, in comparison to test application at 0.77 V with 25 MHz scan shift frequency.

Another effective technique for reducing leakage power is by employing state retention logic [Keating et al., 2007]. Recently a method to test state retention logic is proposed in [Chakravadhanula et al., 2008]. State retention logic is tested by scanning in test patterns, followed by powering

[6] NOR gate is used to halt unnecessary toggling of combinational logic (fed by scan flip-flop) during scan shift operation.



(a) Conventional Adaptive Supply Voltage Regulation

(b) Scan Shift Supply Voltage Regulation (PMScan)

FIGURE 2.18: Block diagram of Adaptive Supply Voltage Regulation in: (a) Conventional design, (b) PMScan [Devanathan et al., 2007b].

down the logic block containing state retention logic and then powering up again. This is followed by scanning out the test patterns, and is matched against the scanned in data for coherency.

2.4 Motivation and Objectives

From the detailed literature review, it is clear that to address various challenges brought forward by the deep submicron defects, a significant effort has been made by researchers around the world on modeling and test generation of such defects. Testing for battery powered, low-power devices add a new dimension to DFT techniques used for testing these defects. Low power

(DVS-enabled) systems can run at different voltage and frequency (V/F) settings during normal operation, it is therefore necessary to ensure that the system will function correctly at any possible V/F setting. Research on very low voltage (VLV) testing [Hao and McCluskey, 1993] and more recently [Engelke et al., 2004, Ingelsson et al., 2007, Engelke, 2009, Ingelsson, 2009] has shown that while some faults cannot be observed at the nominal power supply voltage, they become apparent in different operating conditions, such as lower supply voltage. This means that traditional test methodologies assuming a fixed/nominal power supply voltage and clock frequency, cannot guarantee fault-free operation for such systems. This is because traditional DFT techniques have been developed assuming a fixed V/F setting, which means that whilst DVS-enabled ICs may pass production tests, they can fail in the field at different operating V/F conditions, causing problems with reliability. A possible solution to this problem is to perform production tests at various V/F settings, improving reliability at the expense of increased manufacturing test cost. The semiconductor industry is highly competitive, particularly for consumer products pricing and hence this is not a viable option. This means that the application of DVS to reduce energy consumption may have a detrimental impact on the quality of the manufacturing test employed to detect permanent faults.

The objectives of the research reported in this thesis are as follows:

1. Through simulations, investigate the behavior of defects showing V_{dd} dependent detectability and in the context of multi-V_{dd} designs analyse the detrimental impact of repetitive tests on test cost.

2. Develop effective and low-cost DFT techniques to address the above mentioned challenge brought forward by defects exhibiting V_{dd} dependent detectability.

3. Investigate the impact of multi-V_{dd} designs on diagnosis accuracy, and develop low-cost diagnosis technique to achieve high resolution diagnosis, while targeting such defects.

4. Validate the developed techniques through extensive simulations using advanced parametric fault model, state of the art DFT tools and benchmarks designs widely used in both academic and industrial research.

The main focus of this thesis is to develop DVS-aware DFT and diagnosis techniques. This requires a careful evaluation of the impact of the available DVS-aware DFT techniques on the system manufacturing cost, which can increase due to test application at more than one V_{dd} setting. Similarly, traditional diagnosis techniques using a fixed V/F setting needs to be re-evaluated to ensure high diagnosis accuracy at low cost. The two main objectives of this thesis include:

reducing manufacturing test cost for DVS-enabled devices, and proposing a cost-effective diagnosis technique to improve diagnosis accuracy for such devices.

It has been shown in [Engelke et al., 2004] and more recently in [Ingelsson et al., 2007] that the fault coverage of a test set targeting resistive bridging faults (RBF) can vary with the supply voltage used during test. This means that, depending on the operating V_{dd} setting, a given RBF may or may not affect correct operation of the design. Consequently, to ensure high fault coverage for a design that needs to operate at a number of different V_{dd} settings, it is necessary to perform testing at more than one V_{dd} to detect faults that manifest themselves only at a particular V_{dd}. It was shown in [Ingelsson, 2009] that the majority of circuits (8 out of 12) require testing at more than one voltage setting to achieve 100% fault coverage, which means that the ATE (Automatic Test Equipment) will have to switch between different voltage settings to apply the test. Switching between different V_{dd} settings during test is not desirable and can impact the cost of test. The switching overhead is also linked with test compaction [El-Maleh and Khursheed, 2007], which aims to reduce test size without reducing fault coverage of a test. For DVS-enabled designs, test compaction is confined to tests in single voltage domain as test vectors from different V_{dd} settings cannot be merged as each test has to be applied at a specific voltage setting. This may negatively affect test compaction and increase test application time thereby aggravating test cost. Therefore it is important to reduce the number of test V_{dd} settings to reduce test cost.

There is no reported work on minimizing the number of test V_{dd} settings for multi-V_{dd} designs. The first part of this thesis addresses it and proposes two effective techniques to reduce the number of V_{dd} settings without compromising the fault coverage of the original test employing multiple V_{dd} settings. First, this thesis demonstrates that test point insertion (TPI) can be used to reduce the number of V_{dd} settings during test, without affecting the fault coverage of the original test, thereby reducing test cost. Experiments conducted using ISCAS and ITC benchmarks show that test V_{dd} settings are reduced minimizing test cost. A drawback with the TPI technique is that it does not guarantee a single V_{dd} test and usually results in more than one test V_{dd} setting. Therefore, this thesis proposes a more effective technique for reducing test cost of multi-V_{dd} designs, through gate-sizing (GS). It targets defects that cause faulty logic behavior to appear at more than one test V_{dd} setting, and uses gate sizing (GS) to expose the defect at a single test V_{dd}. The number of test voltages is then reduced, minimizing test cost. We show that unlike TPI, it is possible to achieve single V_{dd} test without affecting the fault coverage of the original test.

The second part of this thesis deals with the diagnosis of Multi-V_{dd} designs in the presence of resistive bridge defects that manifest themselves (as error) at more than one V_{dd} setting. All existing diagnosis techniques use a single V_{dd} setting for fault localization and therefore diagnosis for multi-V_{dd} designs imposes a challenge for defects exhibiting supply voltage dependent behavior. Single V_{dd} diagnosis for multi-V_{dd} designs may lead to imprecise diagnosis affecting failure analysis. This means new and cost-effective diagnosis strategies are required for efficient diagnosis of multi-V_{dd} designs in the presence of bridge defects. The development of such a novel diagnosis technique can be devised by answering the following questions: 1) Is diagnosis resolution affected by different voltage settings? 2) If so, what voltage setting achieves the best level of diagnosis? 3) Is it possible to improve diagnosis resolution further by carrying out diagnosis at more than one voltage setting? 4) For designs operating at more than one voltage setting, it is desirable to reduce diagnosis cost by achieving the minimum possible Test Application Time, without affecting diagnosis accuracy. Therefore, it is important to determine the most useful V_{dd} settings or combination of Vdds, which may yield the desired outcome by omitting tests at some voltage settings.

This is the first investigation that considers diagnosing bridge defects in multi-V_{dd} designs and present results to show the following four findings: 1) The lowest supply voltage provides the best resolution for single voltage diagnosis. It is however different for hard-shorts (bridges with 0 Ω resistance) as experimental results show that diagnosis accuracy has little variation across different voltage settings and therefore any V_{dd} setting can be used without a negative implication on diagnosis accuracy of hard-shorts. 2) The diagnosis resolution can be improved by carrying it out at more than one V_{dd} setting. 3) This work exploits the additional information from other voltage settings to improve the diagnosis accuracy up to 72% over single voltage diagnosis. 4) Finally, we show experimental results using different V_{dd} pairs and identify the most useful V_{dd} pair, such that high diagnosis accuracy is achieved using reduced TAT, thereby reducing diagnosis cost without affecting its accuracy.

2.5 Relevant Contemporary Research

Low power design techniques present potential challenges to test and reliability of digital designs. At the time of thesis compilation, there are continuing research efforts world-wide focusing on addressing these challenges. In the following two emerging research problems are highlighted that need to be addressed, to generate high quality and cost effective test solutions for reliable low power designs.

2.5.1 Impact of Voltage and Process Variation on Test Quality

Previous sections have examined the impact of power supply variation on the behavior of manufacturing defects. It appears that test quality is also compromised due to another type of variation, i.e., due to fabrication process. Whilst the impact of process variation on timing and power performance has been extensively investigated in the literature [Bhunia et al., 2007a], its effect on test quality is an emerging area of research. In this section we summarize two recent studies that take process variation into account using static and delay test techniques and motivate the need for joint voltage and process variation test.

In [Ingelsson et al., 2008] and [Ingelsson, 2009], the impact of process variation on static test quality has been investigated for resistive bridge. It is shown that process variation has a negative impact on test quality of such defects leading to test escapes. A Robustness matrix is developed to quantize the impact of process variation on test quality and a test generation method is developed to mitigate the impact of process variation and reduce test escapes. Experiments are conducted using ISCAS 85' and 89' benchmarks and synthesized using 45 nm CMOS technology. Results show that test generation method covers up to 18% more process variation induced logic faults than tests generated without consideration of process variation. In [Lu et al., 2005] the influence of process variation on the longest path of the design has been investigated, while considering structural elements of the design (logic elements and interconnects). The method aims to reduce test cost without compromising on test quality, i.e., fault coverage. This is achieved by identifying minimum number of longest path candidates in polynomial time. Experiments conducted on ISCAS 85' and 89' circuits show that the number of testable paths are up to 6% of those found by [Tani et al., 1998]. In addition it is 300-3000 times faster than the method proposed in [Tani et al., 1998].

High quality test for next generation Multi-V_{dd} devices require improved static and delay test techniques capable of mitigating the impact of power supply and fabrication process variation. Such test techniques will need to be developed that will require realistic fault models, for both resistive bridge and resistive open, that mimic actual behavior at the physical level in the presence of voltage and process variation. Such fault models will be used for voltage and process variation aware test generation leading to higher test quality and therefore improve in-field product reliability of future Multi-V_{dd} devices.

2.5.2 Voltage Scaling for Nanoscale SRAM

The above open problem is related to test for low-power devices. Recent research indicates that low-power design also affects reliability of the device. One such work that determines optimal voltage setting to operate SRAMs in the presence of soft errors and gate oxide degradation is presented in [Chandra and Aitken, 2009]. Nanoscale SRAMs are vulnerable to soft errors and suffer from progressive gate oxide degradation. Soft errors are faults induced by particle hit (alpha particle or neutrons), which can flip the stored data bit. These events are called Single Event Upsets (SEU) and requires data content to be re-written. SRAMs are especially vulnerable to SEU due to small node capacitance and small bit cell size[7]. On the other hand, gate oxide thickness is continuously decreasing with technology scaling in CMOS devices, which has resulted in increased gate tunneling currents. Increased gate tunneling currents result in progressive degradation of gate oxide, which is one of the most important reliability concern in current and future technologies. In [Chandra and Aitken, 2009], the optimal voltage setting to operate nanoscale SRAM in the presence of soft errors is investigated. This work has shown following three findings: For a given technology node (65 nm or 45 nm), higher voltage level results in higher immunity of SRAM cells against soft errors in the absence of gate oxide degradation. On the other hand, gate tunneling currents increase with the increase in supply voltage, which in turn contributes to gate oxide degradation. Therefore an optimal voltage is formulated by an equation, for operating nanoscale SRAMs in the presence of gate oxide degradation and soft errors. The optimal voltage reduces with increasing level of gate oxide degradation for nanoscale SRAMs.

It is expected that analytical models will be developed to achieve highest immunity against soft-errors for a given voltage setting value and gate-oxide degradation level, thereby improving reliability of nanoscale SRAMs in future technologies.

2.6 Concluding Remarks

This chapter has presented an overview of recently reported research in testing strategies for multi-voltage designs. Such strategies aim to reduce test cost and improve fault coverage of V_{dd} dependent defects. The cost reduction has been obtained by using the least number (i.e., one) of voltage test setting for V_{dd} dependent defects (resistive bridge and resistive open) by avoiding

[7] Refer to [Baumann, 2005] for further reading on the effect of technology scaling and soft errors on memory and logic components of the circuit.

repetitive tests at several V_{dd} settings. For resistive open interconnect defect, elevated V_{dd} setting achieves better detectability using delay test and therefore repetitive tests at other voltage settings can be avoided. However, full-open interconnect defects do not show V_{dd} dependent detectability and therefore can be tested at any V_{dd} setting using static test techniques [Ingelsson, 2009]. Resistive bridge defects (RBDs) show V_{dd} dependent detectability and recent research shows that the lowest V_{dd} setting achieves highest fault coverage, nevertheless this class of defects requires more than one V_{dd} setting to achieve 100% fault coverage. The V_{dd} dependent detectability of RBDs represent multitude of problems for existing DFT and diagnosis solutions. This means that there is no available DFT technique to achieve single V_{dd} test for RBDs without affecting the fault coverage. This thesis proposes two cost-effective DFT techniques in Chapter 3 and Chapter 4 to achieve single V_{dd} test targeting resistive bridges.

The V_{dd} dependent detectability of resistive bridge defects questions the completeness of existing diagnosis techniques, as all existing techniques use single V_{dd} setting for diagnosing such defects. This may lead to reduced diagnosis accuracy with negative affect on failure analysis, which is key to improving subsequent design cycle and yield. This means that novel diagnosis solutions are required for accurate and cost-effective diagnosis of bridge defects in multi-V_{dd} designs. This issue is also dealt with in this thesis and details are available in Chapter 5.

This chapter has also outlined existing low cost scan techniques for multi-voltage design, and in this thesis, the scan architecture is assumed to be fault-free. Low cost scan is possible through various techniques. Some techniques focus on reducing implementation cost of scan chains in multi-voltage environment through clustering scan chains according to their respective voltage domain thereby reducing the number of level shifters and also by employing power-aware scan that efficiently utilize expensive tester resources (bandwidth) and reduce test cost. Other technique achieves low power test for multi-voltage devices by reusing the existing functional infrastructure for voltage scaling to reduce power consumption leading to reduced cost.

The chapter also outlines a number of worthy research problems that need to be addressed to develop high quality and cost effective test solutions for reliable low power devices. A detailed description of the proposed future work is presented in Chapter 6.

Chapter 3

Test Cost Reduction Using Test Points

This chapter discusses the motivation, methodology, tools and experimental results to reduce test cost by a novel DFT (Design for Testability) technique without affecting test quality. The negative impact of multi-V_{dd} testing on test cost is discussed in Section 3.1, followed by Section 3.2 that shows how test point insertion (TPI) can be used to reduce test cost. The details of the proposed TPI technique are presented in Section 3.3, which is followed by experimental results in Section 3.4. Finally, Section 3.5 concludes the chapter.

3.1 Introduction

Multi-V_{dd} designs operate at more than one voltage setting, it was shown in [Ingelsson, 2009, Ingelsson et al., 2007] that testing such devices for resistive bridging faults requires test application at different voltage settings to ensure 100% fault coverage. Table 3.1 shows the number of test patterns to be applied at three different voltage setting (0.8V, 1.0V and 1.2V) to achieve 100% fault coverage [Ingelsson, 2009][1]. In Table 3.1, first column shows various benchmark circuits, the second column marked with # RBF shows the non-feedback resistive bridges that were considered for each design. Column 3-5 shows the number of test patterns generated at each voltage setting i.e., 0.8V, 1.0V and 1.2V respectively. Finally the last column shows the total number of test patterns generated at all three voltage settings to achieve 100% fault coverage. As can be seen from Table 3.1 that majority of circuits (16 out of 22) require testing at more than one voltage setting. This means that the ATE (Automatic Test Equipment) will have

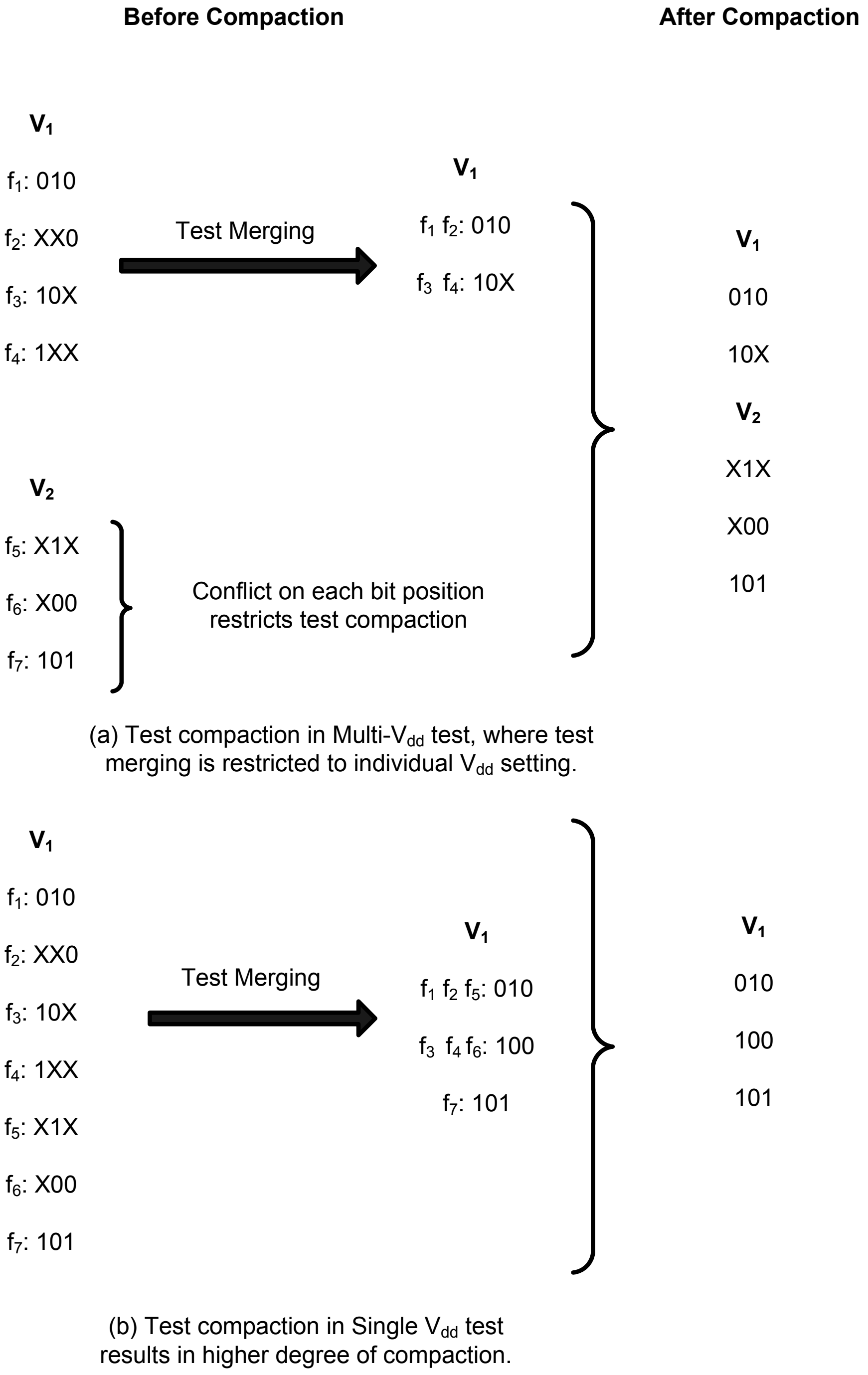
[1] It is different from Table 2.3, which shows the need for additional test vectors (after using a commercial ATPG) at different voltage settings in order to achieve 100% fault coverage. Table 3.1 shows the results without any commercial ATPG and presents results for ITC-99 benchmarks as well.

to switch between different voltage settings to apply the Multi-V_{dd} test, incurring test cost due to switching overhead and degradation of test compaction quality. The loss of compaction quality is illustrated by Figure 3.1, which shows relaxed test vectors (also called atomic components) for seven different fault and the outcome of test merging algorithm [El-Maleh and Khursheed, 2007] on two sets of test vectors to be applied at V_1 and V_2 separately. As can be seen from Figure 3.1-(a) the test merging algorithm results in reducing four test vectors to two at V_1. No reduction in test size at V_2 is possible because of conflict at each bit position of all test vectors in the test set, resulting in overall five test vectors to be applied at two V_{dd} settings separately. Figure 3.1-(b) shows the same set of test vectors but in contrast to the scenario shown in Figure 3.1-(a), all test vectors have to be applied at single V_{dd} setting, thus providing higher flexibility to reduce test set size, resulting in more compact test set. As can be seen, it has resulted in three test vectors thereby achieving higher compaction in comparison to two test V_{dd} settings.

An experiment is conducted to investigate the loss of compaction quality due to repetitive tests at multiple V_{dd} settings; this experiment also quantize the detrimental effect on compaction quality due to multiple voltage settings. For this experiment, test merging algorithm for static test compaction proposed in [El-Maleh and Khursheed, 2007] is used with 13 ISCAS-85 and full-scanned ISCAS-89 benchmarks. For each design, a test set is generated using HITEC test generator [Niermann and Patel, 1991] targeting stuck-at faults in the design, to achieve 100% fault coverage. These test vectors are then divided into two and three partitions of equal size, where each partition mimic a test at a particular V_{dd} setting. The loss in compaction quality is primarily because of test partitioning into different voltage settings, where test vectors can not be combined from a different partition. The purpose of this experiment is to evaluate and demonstrate the detrimental affect of partitioning test vectors and its impact on test compaction. Test compaction is applied for each voltage setting individually and total test count is the sum of test vectors at all voltage settings. In this way, test compaction algorithm is applied at a single partition (representing single V_{dd} test), two partitions (representing two V_{dd} settings) and three partitions (representing three V_{dd} settings). For example a certain design requires 1500 test vectors to achieve 100% fault coverage at a single test V_{dd}. For this design, while considering two V_{dd} test, each partition gets 750 test vectors and, 500 test vectors per partition in case of three test V_{dd} settings. Compaction algorithm is applied individually at each test V_{dd} and results are tabulated in Table 3.2. The first column of Table 3.2 shows the benchmark design, followed by total number of test vectors generated by HITEC. The next three columns show the number of test vectors generated by using test merging algorithm at single test V_{dd}, two test V_{dd} settings, and three test V_{dd} settings (including individual test sizes at each V_{dd} setting in case of more than one V_{dd} setting). As can be seen, for all designs the test count is smallest in case of single V_{dd} test and it increases with each additional V_{dd} setting. This is further illustrated

TABLE 3.1: Multi-V_{dd} test generation results [Ingelsson, 2009]

		# Test Patterns			
Design	# RBF	@ 0.8V	@ 1.0V	@ 1.2V	Sum
ISCAS-85, ISCAS-89 Benchmarks					
c1355	80	39			39
c1908	98	57			57
c2670	104	67			67
c3540	363	184	6	1	191
c7552	577	281		1	282
s838	34	26	2		28
s1488	435	144	2		146
s5378	305	214			214
s9234	223	132	2		134
s13207	358	192	5	1	198
s15850	943	324	4	5	333
s35932	1170	547	50	63	660
ITC-99 Benchmarks					
b01	142	23	1	1	25
b02	33	11		1	12
b03	350	122			122
b04	7,228	1117	17	15	1149
b05	10,000	465	9	10	484
b06	203	16			16
b07	6,447	757	5	11	773
b08	1,350	176	6	2	184
b09	729	86		3	89
b10	1,923	224	1	5	230



(a) Test compaction in Multi-V_{dd} test, where test merging is restricted to individual V_{dd} setting.

(b) Test compaction in Single V_{dd} test results in higher degree of compaction.

FIGURE 3.1: Test compaction for multi-V_{dd} test in comparison with compaction for single V_{dd} test.

TABLE 3.2: Impact of Multi-V_{dd} test settings on compaction.

	Original	**Single Vdd**	**Two Vdd test**			**Three Vdd test**			
Circuit	**test**	**test**	**V1**	**V2**	**Total**	**V1**	**V2**	**V3**	**Total**
c2670	154	98	43	69	112	30	41	46	117
c3540	350	75	61	21	82	48	22	29	99
c5315	193	80	52	44	96	46	36	32	114
s13207	633	238	55	191	246	44	64	156	264
s15850	657	144	63	103	166	54	87	54	195
s38417	1472	130	100	97	197	62	89	56	207
s38584	1174	138	95	107	202	79	51	103	233
s4863	132	47	39	20	59	30	19	13	62
s5378	359	119	70	79	149	46	61	66	173
s6669	138	36	32	17	49	23	16	13	52
s9234	620	170	64	156	220	48	72	120	240

by Figure 3.2, which shows the comparison of test sizes for all designs, in each of the three cases. Higher test reduction with lower number of test V_{dd} settings is because test compaction algorithm gets higher flexibility when combining test vectors resulting in overall smaller test sizes. In comparison to single V_{dd} test, the percentage increase in test size with two and three test V_{dd} settings is tabulated in Table 3.3. It can be seen that the test size is lowest at single test V_{dd} and highest at three test V_{dd} settings. In case of s38584 the increase in test size is as much as 69% in comparison to single test V_{dd}, while considering two test V_{dd} settings the test size has increased by up to 52% as in case of s38417. This experiment clearly shows the detrimental affect of multi-V_{dd} setting on test compaction thereby increasing test cost.

Test cost constitutes a substantial percentage of total manufacturing cost [Bedsole et al., 2001]. Switching between supply voltage settings during test is not a trivial task and increases the cost of test, mainly due to the switching time overhead and loss in test compaction quality. Therefore it is important to reduce the number of test V_{dd} settings to reduce test cost, which is the aim of this chapter.

TABLE 3.3: Impact of Multi-V_{dd} settings on test compaction.

Circuit	%incr. at Two Vdd settings	%incr. at Three Vdd settings
c2670	14%	19%
c3540	9%	32%
c5315	20%	43%
s13207	3%	11%
s15850	15%	35%
s38417	52%	59%
s38584	46%	69%
s4863	26%	32%
s5378	25%	45%
s6669	36%	44%
s9234	29%	41%

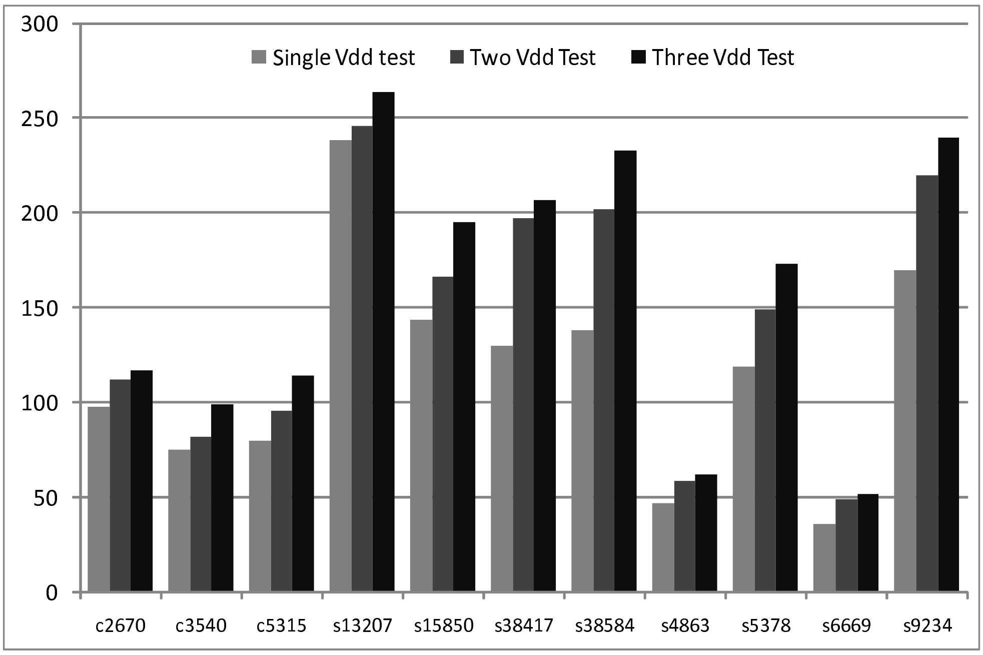


FIGURE 3.2: Impact on test compaction.

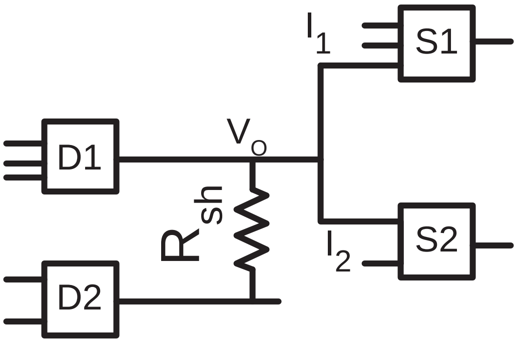


FIGURE 3.3: Resistive Bridge

Figure 3.3 and 3.4 are used to explain why MVTG [Ingelsson et al., 2007, Khursheed et al., 2008] generates tests at more than one V_{dd} setting. Figure 3.3 shows a resistive bridge feeding two different gates of a circuit, which is assumed to be operating at three voltage settings. By analysing the analog behavior of the bridge (using the same procedure as explained for Figure 2.5, 2.6, 2.7) at three different voltage settings we can determine the resistance intervals detected at each one of the three voltage settings. The covered resistance intervals at three voltage settings are shown in Figure 3.4. Figure 3.4 marks the redundant (gray bars) and non-redundant (black bars) intervals of a resistive bridge at three different voltage settings. For a certain bridge *Essential* V_{dd} setting is the one at which the highest resistance interval is detected, which is V_3 in this case. From test generation point of view, essential V_{dd} has to be included in test generation as highest resistance interval exists at essential voltage setting(s). This means that any of the resistance intervals targeted at non-essential test V_{dd} setting(s) by the test generation algorithm can be detected at one of the essential test V_{dd} setting(s), subject to suitable controllability and observability at the bridge site. On the other hand non-essential voltage settings are included in test generation only because some non-redundant intervals are detectable at non-essential voltage setting(s), these intervals are referred as NRINEV (Non-Redundant Intervals at Non-Essential V_{dd}). Two such NRINEV intervals, marked by A and B are shown in Figure 3.4. It should be observed that these two NRINEV intervals are redundant at V_3 (essential Vdd) and require additional controllability and observability for detection at essential V_{dd}. The additional controllability and observability is achieved by the help of test point insertion that uses additional test points to detect logic fault corresponding to resistance intervals “A” and “B” at V_3, as shown in Figure 3.4. The need of test generation at either V_1 or V_2 is then reduced resulting in test cost reduction.

Previously, test point insertion (TPI) has been used for increasing the fault coverage [Touba and McCluskey, 1997] and test compaction [Geuzebroek et al., 2000]. The next section shows

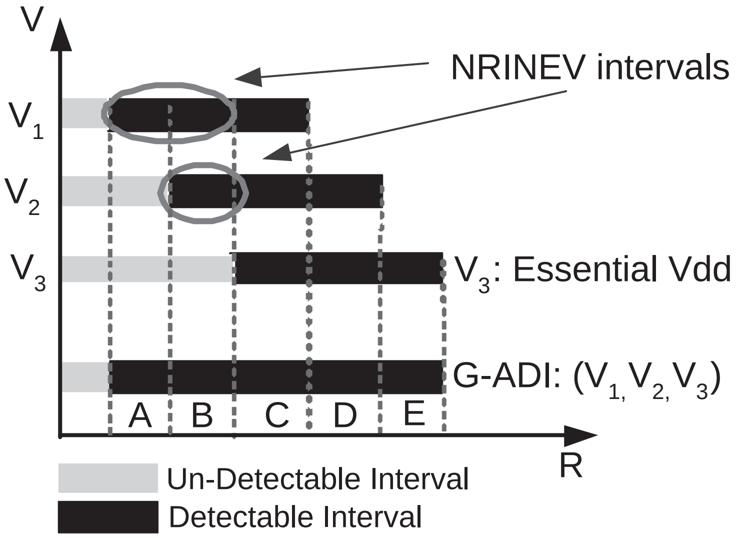
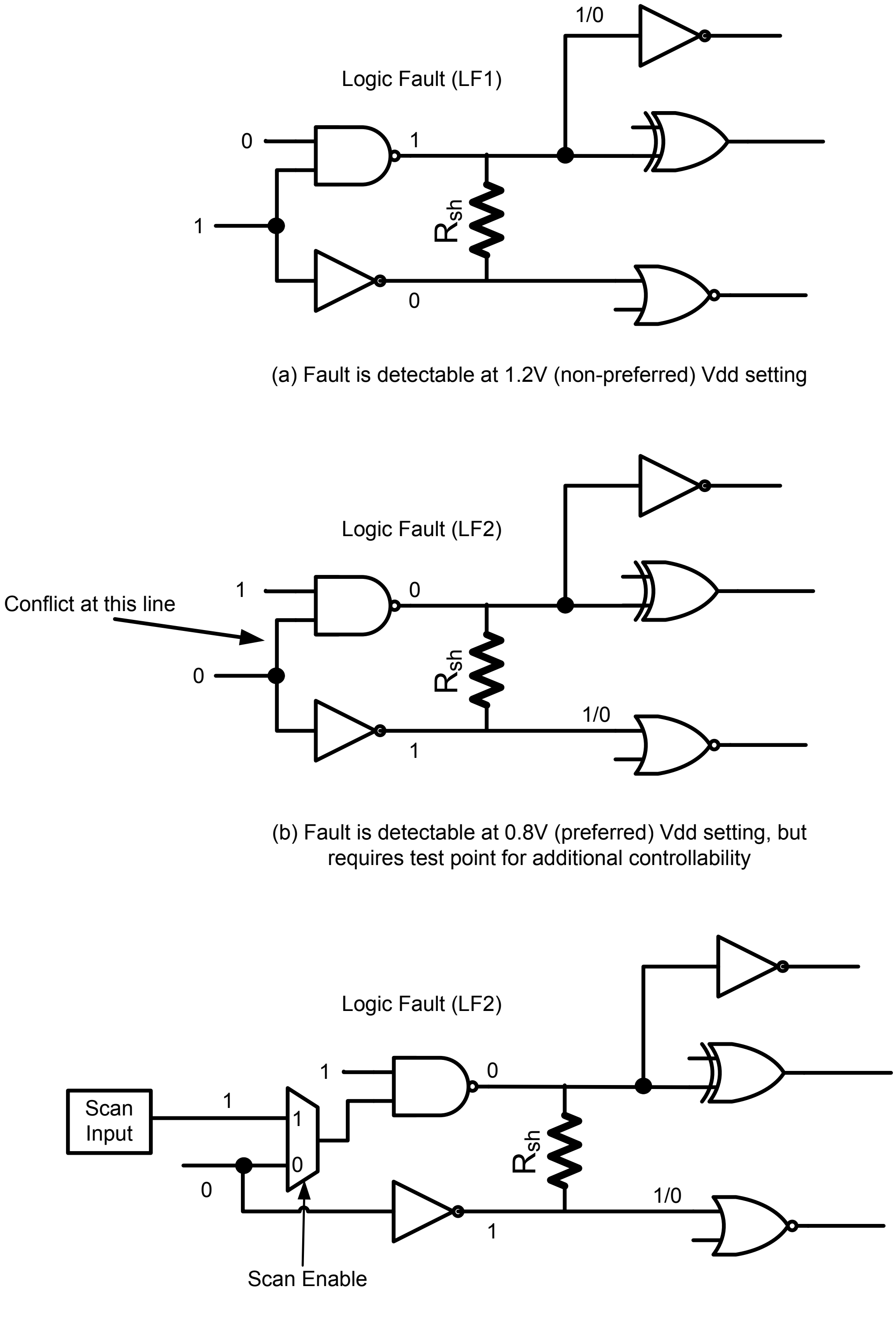


FIGURE 3.4: Reasons for test generation at more than one V_{dd} setting

how TPI can be used to reduce the number of different V_{dd} settings required during test without affecting the fault coverage.

3.2 Motivation

Test point insertion (TPI) is a well-known design for test (DFT) technique to provide additional controllability and observability [Hayes and Friedman, 1974, Hayes, 1974]; test points are added in a design to achieve higher fault coverage and have been used for test compaction as well [Geuzebroek et al., 2000, Touba and McCluskey, 1997]. The principle of using test points is discussed in detail in Section 1.6.1. Figure 3.5 shows how test points are used for reducing the number of test V_{dd} settings. A bridge location has a number of logic faults (referred as “Logic State Configuration” LSC) that comprises of the following four parameters: 1) inputs to the gates feeding the bridge, 2) resistance range covered, 3) boolean values interpreted by the gates fed by the bridge and 4) V_{dd} setting at which the fault appears. For a given bridge, the test generator [Ingelsson, 2009] targets minimum number of logic faults to cover maximum detectable resistance of the bridge and selects test patterns accordingly. Test points are used where an un-detectable logic fault at the lowest V_{dd} setting (preferred V_{dd}) covers higher bridge resistance than detectable logic faults at higher V_{dd} setting(s). This is explained in Figure 3.5 that shows two logic faults, LF1 and LF2. Logic fault (LF1) is shown in Figure 3.5-(a), the resistance range is detected at 1.2V by fault propagation through the inverter. The other logic



(a) Fault is detectable at 1.2V (non-preferred) Vdd setting

(b) Fault is detectable at 0.8V (preferred) Vdd setting, but requires test point for additional controllability

(c) Additional test point to achieve desired controllability

FIGURE 3.5: Use of test points for additional controllability and reduction in test V_{dd} settings

TABLE 3.4: Proof of concept: test point insertion to reduce the number of test voltage settings

	V_{dd} settings		Control	Observation	Total
Design	bf TPI	af TPI	Points	Points	Test Points
c432	0.8V, 1.2V	0.8 V	2	2	4
c499	0.8V	0.8V	0	0	0
c880	0.8V, 1.0V, 1.2V	0.8V, 1.0V	2	0	2
c1355	0.8V, 1.2V	0.8 V	10	3	13
c1908	0.8V, 1.2V	0.8 V	8	1	9
Total			22	6	28

fault (LF2) is shown in Figure 3.5-(b), it covers higher resistance range but as can be seen, it is undetectable due to a conflict on the net feeding the two driving gates. As can be seen, logic-0 is required at the output of nand gate, which is only possible by applying logic-1 at both the inputs of nand gate. At the same time, logic-0 is required at the input of inverter to activate the bridge. This conflict is resolved by an additional test point at this input and the resultant circuit is shown in Figure 3.5-(c). The added test point at the input of nand gate allows bridge activation[2] and fault effect is propagated through the nor gate.

This idea is further investigated by conducting an experiment by using ISCAS-85 benchmarks and three test voltage settings, 0.8V, 1.0V, and 1.2V. The aim of this experiment is to identify bridge locations that need a test at more than one voltage setting. For these bridge locations, test point insertion (TPI) is then used to target redundant (un-detectable) faults to cover the same resistance interval at the lowest V_{dd} setting, which is followed by test generation to ensure that a test pattern can be generated at the lowest V_{dd} setting. Thus this experiment serves as a proof of concept that TPI can be used to reduce the number of test V_{dd} settings.

For this experiment, the benchmarks are synthesised using a 0.12μm ST Microelectronics gate library using Synopsys design compiler and for each design only non-feedback bridges are considered[3]. Test patterns are generated using Multi-Voltage test generator (MVTG) proposed in [Ingelsson, 2009], the details of which are given by Appendix B. For each design, MVTG aims to achieve 100% fault coverage using minimum number of test patterns. The results are tabulated in Table 3.4, which shows benchmark designs in the first column, followed by the test V_{dd} settings to achieve 100% fault coverage, as generated by MVTG. Third column shows the impact of TPI on each of the design in reducing the number of test V_{dd} settings. The next two columns show the number of control points and observation points used by the TPI, and finally

[2] A resistive bridge is activated by setting opposite logic values on the two nets.

[3] Appendix D shows SPICE description of three gates from the gate library

the last column shows the total number of control and observation points. As can be seen, TPI is able to reduce the number of test V_{dd} settings for all designs requiring more than one test V_{dd} setting. In case of c880, test voltage 1.0V has a detectable resistance interval, which can not be covered at the lowest V_{dd} setting resulting in two test V_{dd} settings. This experiment provides initial results to show that TPI can be effective for reducing the number of test V_{dd} settings. The next section provides implementation details of the proposed TPI technique with emphasis on reducing the number of control and observation points.

3.3 Proposed Test Point Insertion Technique

The algorithmic flow for reducing the number of V_{dd} settings during test is outlined in Figure 3.6. The key steps of this technique are further detailed in Figure 3.9, 3.10, 3.11 and 3.12.

The algorithm starts (Figure 3.6) by computing the set of essential test V_{dd} setting(s) for the given voltage settings and bridge list. To achieve this, for each bridge B, the algorithm determines the highest detectable bridge resistance value across all available V_{dd} settings and marks the V_{dd} setting corresponding to the highest resistance value as essential V_{dd}. In line 2, the algorithm determines for each bridge the set of resistance intervals which cause faulty behavior at a non-essential V_{dd}, but are fully or partially undetectable at any of the essential V_{dd} setting(s) due to lack of suitable controllability or observability. These resistance intervals are referred to as Non Redundant Interval at Non-Essential Voltage (NRINEV). Next, in lines 3 to 6, for each NRINEV, the algorithm determines a set of test points needed to make the resistance interval detectable at an essential V_{dd} setting. For this purpose, a set of LSC which fully cover the NRINEV interval is identified. Since in most cases, more than one set of LSCs can be used to cover the same NRINEV, the algorithm selects the LSC set which is likely to require the least number of test points to become detectable. The LSC selection algorithm used for this purpose is detailed in the following section. Once all NRINEV intervals have been covered, in lines 7 and 8 an attempt is made to reduce the number of required test points by identifying test points which can be shared among two or more selected LSCs. The algorithm then inserts the resulting set of test points into the original netlist and invokes MVTG [Ingelsson, 2009] to generate the test sets corresponding to the set of essential V_{dd} setting(s). The flow chart of this procedure is shown in Figure 3.7.

```
Compute set of Essential Test V$_{dd}$ setting(s) ($V_{ess}$)
Compute set of NRINEV
for all $NRINEV$ do
    LSC Selection(NRINEV, $V_{ess}$)
    Determine a preliminary set of test points at the defect site boundary for detecting the selected LSCs
end for
Minimize set of observation points
Control Point Minimization ()
Generate Essential V$_{dd}$ Test Sets for the modified netlist
return ($netlist$, $Test\ Sets$)
```

FIGURE 3.6: Test Point Insertion

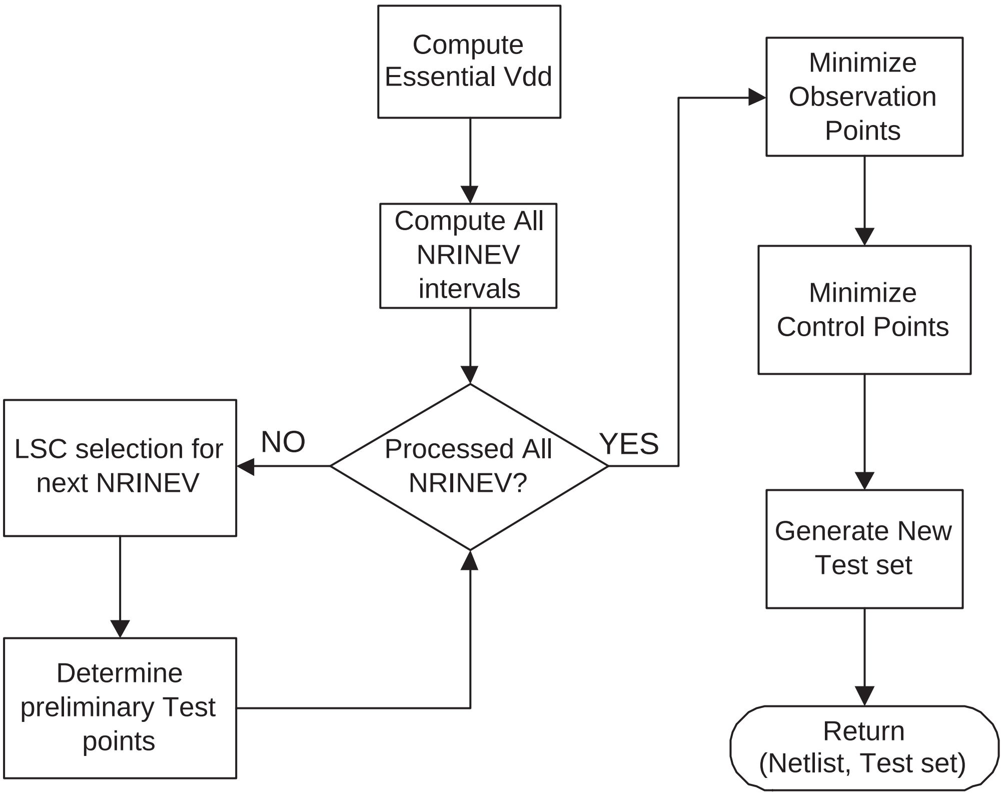

Compute Essential Vdd
Compute All NRINEV intervals
Processed All NRINEV?
NO
YES
LSC selection for next NRINEV
Determine preliminary Test points
Minimize Observation Points
Minimize Control Points
Generate New Test set
Return (Netlist, Test set)

FIGURE 3.7: Algorithmic flow of the proposed TPI technique

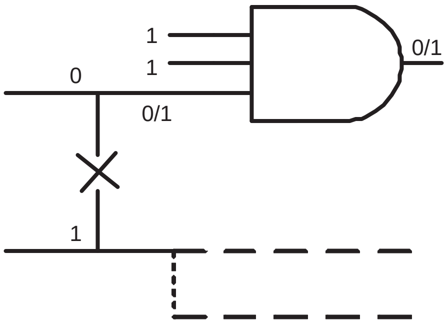


FIGURE 3.8: Observability calculation

3.3.1 LSC Selection

LSC selection aims to determine a set of LSC covering a given NRINEV which is likely to require the least number of test points[4]. The algorithm, illustrated in Figure 3.9, uses signal probabilities to quantify the effort required to control the logic values required by a LSC on the corresponding nets. In our experiments, signal probabilities were determined by simulating 5000 pseudo random patterns, however other analytical methods for estimating signal probability could be used for this purpose just as well. The algorithm continues by identifying all LSCs which expose resistance intervals fully or partially overlapping with the target NRINEV interval. A probabilistic estimate of the controllability and observability (PECO) is computed for each candidate LSC (steps 3 to 5) as follows:

$$PECO(LSC) = C(LSC) \cdot O(LSC) \tag{3.1}$$

where C(LSC) is a probabilistic measure of the LSC controllability and O(LSC) is a probabilistic measure of the observability of the defect at the outputs of the gates fed by the bridge.

$$C(LSC) = \prod_{i=1}^{n} (Prob(i)) \tag{3.2}$$

where n is the cumulated number of inputs of the two gates driving the bridged nets and *Prob(i)* is the probability of logic value required by the LSC on input i.

$$O(LSC) = \sum_{i=1}^{m} (f(X)) \tag{3.3}$$

[4] Section 2.1.1 provides details of LSC data structure, which is also referred as logic fault.

```
Input: NRINEV interval
    Essential V_dd settings $V_{ess}$
Output: Set of LSCs covering NRINEV with minimum number of required test points
Compute signal probabilities on all nets
Generate a list of LSC candidates sets which cover partially or completely the NRINEV at $V_{ess}$
for all LSC candidates do
    Compute PECO(LSC)
end for
Determine the set of LSC covering NRINEV with maximum overall PECO
return LSC selection
```

FIGURE 3.9: LSC Selection

where m is the number of gates fed by the bridged nets which propagate the faulty value to their outputs and $f(X)$ is the probability that the fault effect is propagated through gate X, computed as follows:

$$f(X) = \frac{\sum_{j=1}^{k} \prod_{i=1}^{l} SP_{i,j}}{2^l} \tag{3.4}$$

where k is the number of input combinations which propagate the fault effect to the output of successor gate X, l is the number of inputs of gate X which are not fed by the bridge, and $SP_{i,j}$ is the probability of having the value corresponding to input combination j on input i. For example, for a 3-input AND gate fed by the bridge (as shown in Figure 3.8) there is one input configuration which will propagate the fault (0/1) to its output out of the 4 possible combinations on the two inputs which are not fed by the bridge. Assuming the “1” probabilities of the inputs which are not driven by the bridge to be 0.4 and 0.7 respectively, the probability of this gate propagating the fault to its output is $\frac{(0.4*0.7)}{4} = 0.07$. In this way O(LSC) provides a probabilistic estimate to help compare various LSCs and favor the one which is likely to require lesser number of observation points.

PECO(LSC) is then used as weight in a set covering linear programming formulation to determine the LSC set covering NRINEV which is likely to require the fewest number of test points.

At this point, the selected LSCs can be made controllable and observable by inserting appropriate test points at the defect site boundary.

3.3.2 Preliminary Test Point Insertion at the Defect Site Boundary

The method proposed for determining the preliminary set of test points at the defect site boundary for a given LSC is shown in Figure 3.10. The algorithm starts by checking whether the driving gates' input assignments required by the LSC can be satisfied. If the required input assignments can be satisfied, it means there is at least one test pattern which activates the fault. Otherwise the algorithm attempts to determine a set of control points necessary for activating the fault (lines 2-15). This is achieved by using incremental bit-flipping on the driving gates' input assignments until a satisfiable combination is found. The input nets corresponding to the bit-flips in the LSC represent control point candidates and are added to the Exclusive Control Point Candidate list (ECL). At this point (step 17), the algorithm attempts to generate a test pattern which detects LSC and returns on successful generation of a test pattern. If a test pattern detecting the LSC could not be found, it means that although the fault can be activated, it is not observable at the primary outputs. At this point, the following two scenarios are possible: the faulty behavior can be observed at the output of at least one of the successor gates, or, the faulty behavior does not propagate through any of the successor gates. In order to differentiate between these two issues, a stimulus is generated for fault activation. This stimulus is applied to the circuit and all the successor gates are checked to see if the faulty behavior is observable at the output of any of these gates. If the fault is observable at the output of these gates, then the algorithm structurally traverses the circuit and marks all the nets that observes the faulty behavior as potential observation point candidates (step 22). If the fault effect is not observable at the output of any of the successor gates, the algorithm uses the logic values on all the nets, set by the stimulus generated in step 20 of the algorithm, and identifies the successor gate which observes the faulty value and requires the least number of control points in order to propagate it to its output. The nets corresponding to these control points are then added to ECL. In lines 28 to 34 the algorithm repeats steps 17 to 23 to mark all the nets that observe the faulty values for later observation point minimization, if a test pattern cannot detect the defect even after inserting control points for observability.

3.3.3 Test Points Minimization

The TPI algorithm (Figure 3.6) minimizes the number of observation points, after processing all the NRINEV intervals. The optimum set of observation points will be the minimum set cover of the nets marked as observation point candidates in lines 22 and 33 of Figure 3.10. This is similar to the method proposed in [Touba and McCluskey, 1996].

```
Input: LSC Candidate
       Bridge $b$
for all Gates driving the bridge do
    if LSC input assignment not satisfiable then
        CPCount = 1;
        SATISFIED = FALSE;
        while NOT SATISFIED do
            for all LSCIA = LSC input assignment with CPCount bit-flips do
                if LSCIA is satisfiable then
                    SATISFIED = TRUE;
                    add nets corresponding to bit-flips in LSCIA to ECL
                    BREAK;
                end if
            end for
            CPCount = CPCount + 1
        end while
    end if
end for
if LSC non-redundant then
    return $(success)$
end if
Generate a stimulus to activate the fault
if Fault is observable at the output of the gates fed by the bridge then
    Mark all the nets which observe the fault effect as OP candidates
else
    Use the logic values set by the stimulus at the inputs of the gate
    Identify a gate, from all the gates which see a fault, that require min. no. of CPs to
    propagate the fault
    add control point candidates to ECL
end if
if LSC non-redundant then
    return $(success)$
end if
Generate a stimulus to activate the fault
if Fault is observable at the output of the gates fed by the bridge then
    Mark all the nets which observe the fault effect as OP candidates
end if
```

FIGURE 3.10: Preliminary test point identification at the defect site boundary

```
1:  for all NRINEV do
2:      for all ec $\in$ ECL do
3:          Compute FIC(ec)
4:      end for
5:  end for
6:  for all pair $(ec_i, ec_j)$ where $ec_i$, $ec_j \in$ ECL do
7:      CN$(ec_i, ec_j)$ = FIC$(ec_i) \bigcap$ FIC$(ec_j)$
8:      VC$(ec_i, ec_j)$ = Find Valid CP Candidates (CN$(ec_i, ec_j)$)
9:  end for
10: Find minimum number of CPs as a minimum set cover on $\{$VC$(ec_i, ec_j)\}$
11: Insert CPs into netlist
```

FIGURE 3.11: Control Point Minimization

The TPI algorithm calls control point minimization algorithm in step 8 of Figure 3.6, to reduce the number of control points in the modified circuit. This is achieved by finding pairs of control point candidate nets which can be replaced by a single control point while still achieving the required controllability. The algorithm (shown in Figure 3.11) starts by determining the fan-in cone (FIC) sets for each net added to the ECL set in lines 9 and 26 of Figure 3.10. FIC(ec) consists of all nets in the fan-in logic cone of ec, starting from the primary inputs. Basically, FIC(ec) contains all nets which may affect the logic value on ec. Next, the algorithm finds the Common Nets (CN) for the FIC of all possible pairs of nets in ECL, i.e., CN(ec_i, ec_j) holds the nets which appear in both FIC(ec_i) and FIC(ec_j). For every set of common nets CN(ec_i, ec_j), the algorithm attempts to determine a list of valid candidates (VC) shown in line 8, where every valid candidate is able to provide the required controllability on (ec_i and ec_j), thus reducing two control points to one. These valid candidates are generated by algorithm shown in Figure 3.12 (Find Valid CP Candidates) for every pair of control points in ECL. The algorithm then determines the minimum set of control points as a minimum set cover for all VC sets. The resulting set of control points are then inserted in the netlist.

The algorithm shown in Figure 3.12 starts by creating a copy of the netlist without any control points, but with the optimized observation points at their respective locations. For every pair of control point candidates (ec_A and ec_B) the algorithm inserts all control points necessary to detect LSC(A) (using information stored in ECL), with the exception of ec_A and ec_B, where LSC(A) is the LSC corresponding to ec_A. It then tries all the common nets CN(ec_A, ec_B), one-by-one and attempts to generate a stimulus using both types of control points CP-1 and CP-0. For all candidates that detect LSC(A) a tuple consisting of the net, fanout and CP-type is placed in *First Valid Candidates, FVC*. The algorithm then moves to LSC(B) and repeats the above procedure but this time it uses the members of *FVC* instead of common nets' members. It then

```
Input: $ec_A, ec_B$, CN($ec_A, ec_B$), LSC(A), LSC(B)
Create a copy of the original circuit
Insert all the CPs required by LSC(A) with the exception of $ec_A, ec_B$
for all cn $\in$ CN($ec_A, ec_B$) do
    for all cptype $\in$ CP-0, CP-1 do
        Insert a control point (cptype) at cn
        if LSC(A) is non-redundant then
            FVC = FVC $\bigcup$ {cn}
        end if
    end for
end for
if FVC $\neq \emptyset$ then
    Insert all the CPs required by LSC(B) with the exception of $ec_A, ec_B$
    for all fvc $\in$ FVC do
        Insert a control point of type cptype(fvc)
        if LSC(B) is non-redundant then
            VC = VC $\bigcup$ fvc
        end if
    end for
end if
return VC
```

FIGURE 3.12: Find Valid CP Candidates

adds all those members of *FVC* which are able to detect LSC(B) to *Valid Candidates*, *VC* list and returns the list to the calling Algorithm (Figure 3.11).

3.4 Experimental Results

The TPI algorithm (Figure 3.6) has been validated using a experimental set up, utilizing ISCAS-85, ISCAS-89, and ITC-99 benchmark circuits, see Appendix C for detailed description of all benchmark designs. The sequential circuits are treated as combinational by assuming full-scan design and only non-feedback bridges are targeted. The benchmark circuits are synthesised using a 0.12μm ST Microelectronics gate library. Synopsys Design Compiler^{TM} is used for synthesis, as well as, to evaluate timing, area and power. Default options of DC are used for synthesis without specifying any time constraints on any design. The generated netlist is then used for test point insertion to reduce the number of test V$_{\text{dd}}$ settings. All experiments are conducted using three V$_{\text{dd}}$ settings: 0.8V, 1.0V, and 1.2V. The selection of V$_{\text{dd}}$ settings is similar to a commercial microprocessor (TransMeta Crusoe TM5800) [TM5, 2009], that varies V$_{\text{dd}}$ settings from 0.9V to 1.3V and is synthesized using 0.13 μm cell library. The test point insertion flow on the layout extracted bridge list required only a very small number of test points, only 3 out of 12 circuits required test points. This is why, an exhaustive bridge list is generated by considering all possible pairs of nets in the netlist, up to a maximum of 10,000 pairs. This increases the total number of bridges for all the circuits and therefore, creates more challenging test cases than coupling capacitance based post-layout extracted bridge list. The number of bridge locations using coupling capacitance based extraction for ISCAS designs vary from 47 to 943, for c432 and s15850 respectively using the same gate library (see Table 5.4 on page 129). It should be noted that the number of extracted bridges depend on the type of gates available in the gate library that are used during synthesis. The use of compound gates (with upto 9 inputs) reduces the gate count (in comparison to 2 input AND/OR gates) resulting in reducing the number of extracted bridge locations. For the same reason, in a recent study reported in Engelke et al. [2009] the experimental setup uses the number of gates multiplied by 10 to determine the total number of random bridges to be considered.

All benchmarks along with their respective number of gates and total number of bridges are shown in Table 3.5. The experimental data is available at (TPI: Experimental data) [TPI, 2007] to enable fair comparison with this work. This setup is used to conduct two sets of experiments. The first set of experiment shows the impact of the proposed test point insertion (TPI) technique in reducing the number of test V_{dd} settings and the second set of experiment demonstrate the impact of TPI on timing, area and power (dynamic and leakage) in comparison to the original design. The tool flow for reducing the number of test V_{dd} settings, using the proposed test point insertion technique, is shown in Figure 3.13. It should be noted that a new list of bridges is generated to take in to account the additional bridge locations after inserting test points in the original design. These additional bridge locations have a test point(s) as a driving/driven gate(s) and should be taken in to account before generating final test set to ensure that they do not require higher V_{dd} test.

3.4.1 Test V_{dd} Reduction Using TPI

The objective of this experiment is to show the impact of the proposed test point insertion (TPI) technique to reduce test V_{dd} settings. It also shows the benefit of using control point and observation point minimization algorithm to reduce the number of test points by comparing the number of test points with the preliminary version of this work published in [Ingelsson et al., 2007].

Table 3.6 shows the number of test V_{dd} setting(s) required to achieve 100% fault coverage in the original design and after inserting test points using the proposed TPI technique. As can be

Table 3.5: Benchmarks

ISCAS-85 and ISCAS-89		
CKT.	# Gates	Total Bridges
c1355	226	6,563
c1908	205	7,986
c2670	269	10,000
c3540	439	10,000
c7552	731	9,998
s344	62	469
s382	74	1,146
s386	63	1,625
s838	149	5,737
s5378	578	9,933
s9234	434	10,000
s13207	1064	10,000
s15850	1578	10,000

ITC-99		
Ckt	# Gates	# Bridges
b01	26	142
b02	15	33
b03	63	350
b04	208	7,228
b05	315	10,000
b06	33	203
b07	170	6,447
b08	86	1,350
b09	75	729
b10	88	1,923

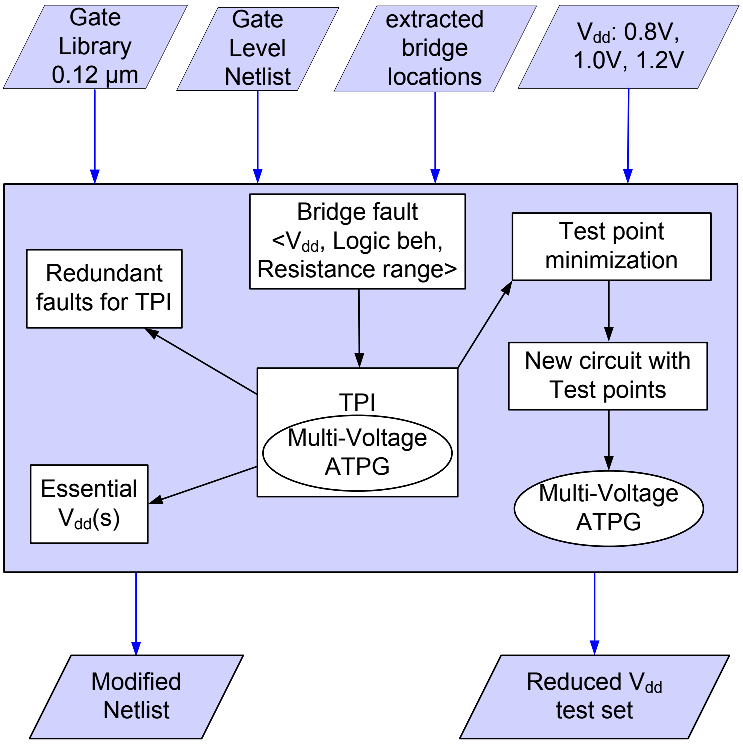


FIGURE 3.13: Tool flow of the proposed TPI technique

seen, the proposed TPI technique has reduced the number of test V_{dd} settings to single V_{dd} for 10 designs and two V_{dd} settings for 10 designs. This is achieved without affecting the fault coverage of the original test. It is only for s9234 where the number of test V_{dd} settings could not be reduced, this is because it has bridges with highest critical resistance at all three test voltages, i.e., they are all essential. The number of control and observation points added to each design is also shown in Table 3.6. It should be noted that total number of test points (including OPs and CPs) are ten or less for a large majority of circuits, it is only in case of c2670, s838 and b05 that additional test points are used. For all designs, on average TPI has added 6.7% additional gates.

The number of test points used by the proposed TPI technique is next compared with an earlier TPI implementation that was published in [Ingelsson et al., 2007]. The TPI implementation in [Ingelsson et al., 2007] does not use any minimization algorithm for control and observation points. This comparison is shown in Table 3.7 for all the circuits presented in [Ingelsson et al., 2007]. It can be noticed that the number of control points have reduced by more than 40% and this effect is even more pronounced for c1908. Similarly it achieves more than 66% reduction in the number of observation points, for the same set of circuits and bridge list. This clearly shows the effectiveness of test point minimization algorithms shown in Figure 3.6.

TABLE 3.6: Results of Test Point Insertion algorithm

ISCAS-85 and ISCAS-89 Benchmarks					
	V_{dd}(s)	V_{dd}(s)			Total
Design	bf TPI	af TPI	CP(s)	OP(s)	Test Point(s)
c1355	0.8v, 1.2v	0.8v	6	0	6
c1908	*All	0.8v, 1.2v	2	1	3
c2670	All	0.8v, 1.2v	19	0	19
c3540	All	0.8v, 1.0v	6	1	7
c7552	0.8v, 1.2v	0.8v	0	1	1
s344	All	0.8v	5	0	5
s382	All	0.8v, 1.2v	7	2	9
s386	All	0.8v, 1.0v	9	1	10
s838	All	0.8v, 1.0v	26	11	37
s5378	All	0.8v, 1.0v	5	1	6
s9234	All	All	0	0	0
s13207	All	0.8v, 1.0v	3	0	3
s15850	All	0.8v, 1.0v	3	0	3

ITC-99 Benchmarks					
	V_{dd}(s)	V_{dd}(s)			Total
Design	bf TPI	af TPI	CP(s)	OP(s)	Test Point(s)
b01	All	0.8v	1	0	1
b02	1.2v, 0.8v	0.8v	2	0	2
b03	0.8v	0.8v	0	0	0
b04	All	0.8v	1	3	4
b05	All	0.8v	30	12	42
b06	0.8v	0.8v	0	0	0
b07	All	1.2v 0.8v	10	0	10
b08	All	0.8V	6	2	8
b09	1.2v, 0.8v	0.8V	2	0	2
b10	All	0.8V	5	0	5

*All = 0.8v, 1.0v, 1.2v

TABLE 3.7: Comparison of the number of test points

	Total CP(s)		Total OP(s)	
Design	TPI*	TPI Alg. 3.6	TPI	TPI Alg. 3.6
c432	2	3	2	1
c499	0	0	0	0
c880	2	2	0	0
c1355	10	6	3	0
c1908	8	2	1	1
Total	22	13	6	2

*TPI technique presented in an earlier version of this work in [Ingelsson et al., 2007]

The fault coverage achieved at single voltage setting (0.8V) is shown in Table 3.8, which can be used to understand the trade-off between test cost and fault coverage. The table shows fault coverage at 0.8V after inserting test points for all the circuits. As can be seen, the TPI achieves very high fault coverage at 0.8V for a large majority of designs, which means that small number of test patterns are generated at other voltage settings (1.0V and 1.2V) after inserting test points.

TABLE 3.8: Fault coverage at 0.8V after inserting Test Points

ISCAS-85 and ISCAS-89 Benchmarks	
CKT.	Fault coverage at 0.8V
c1355	100%
c1908	99.99%
c2670	99.99%
c3540	99.37%
c7552	100%
s344	100%
s382	99.99%
s386	99.69%
s838	99.99%
s5378	99.99%
s9234	90%
s13207	84.62%
s15850	89.54%

ITC-99 Benchmarks	
CKT.	Fault coverage at 0.8V
b01	100%
b02	100%
b03	100%
b04	100%
b05	100%
b06	100%
b08	100%
b09	100%
b10	100%

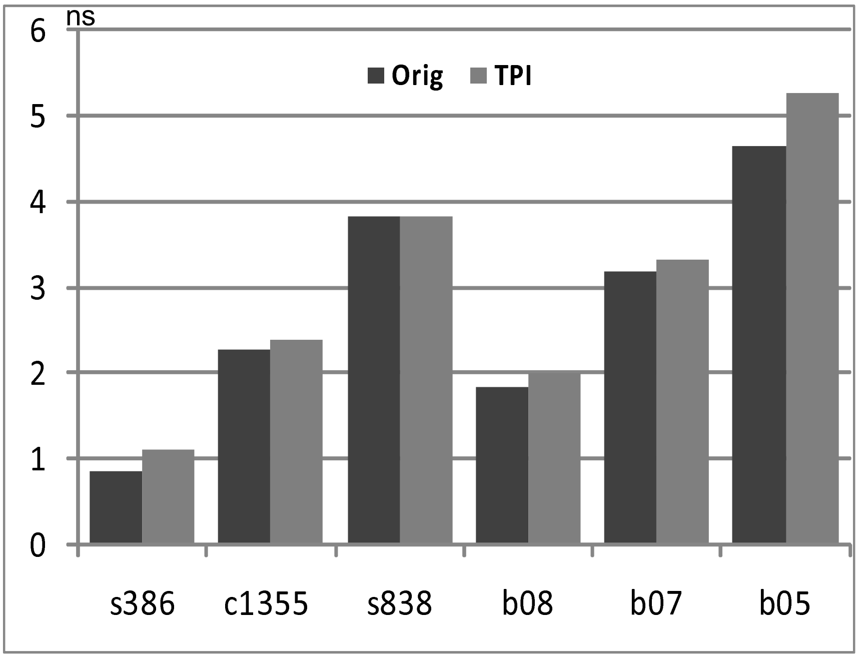


FIGURE 3.14: Impact of TPI on timing in comparison with the original design.

3.4.2 Impact on Timing, Area and Power

The second set of experiment compares timing, area and power (dynamic and leakage) of the proposed TPI technique with the original design. Figure 3.14 shows the timing comparison. As can be seen, the proposed TPI technique has a negative effect on timing when compared to the original design. This is because of the test points inserted in the critical path. For example, in case of s386, TPI has inserted 1 test point, and converted a non-critical path into a critical path resulting in increased timing. Similarly, comparison of area overhead is shown in Figure 3.15 for the two designs. The proposed TPI technique results in a higher area overhead in comparison to original designs for all circuits, which is because of additional test points. Finally, comparison of dynamic and leakage power is shown in Figure 3.16 and Figure 3.17 respectively. It can be seen that the proposed TPI technique increases the power budget in comparison to the original design. High power consumption of the TPI is because of additional switching activity, load capacitance and leakage power of added test points.

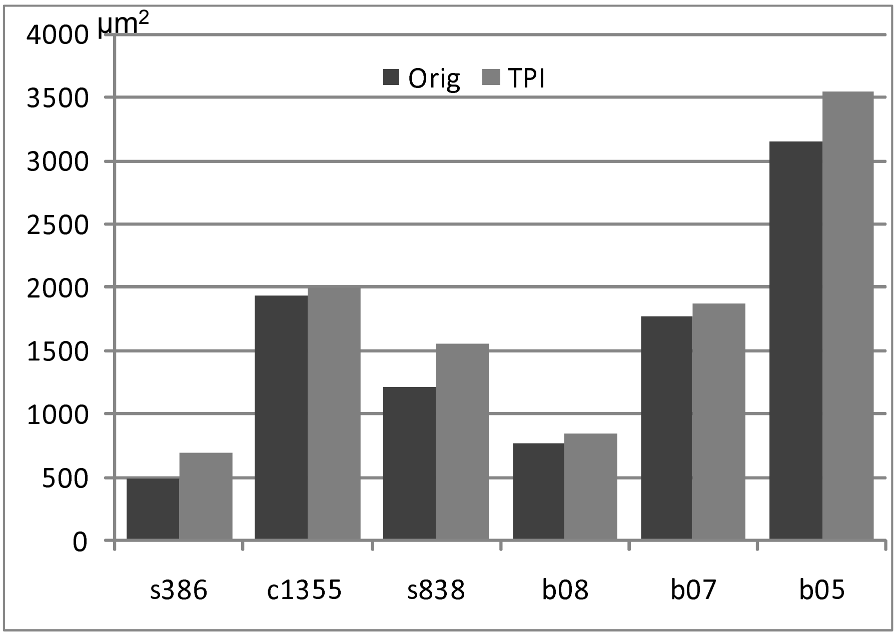


FIGURE 3.15: Impact of TPI on area in comparison with the original design.

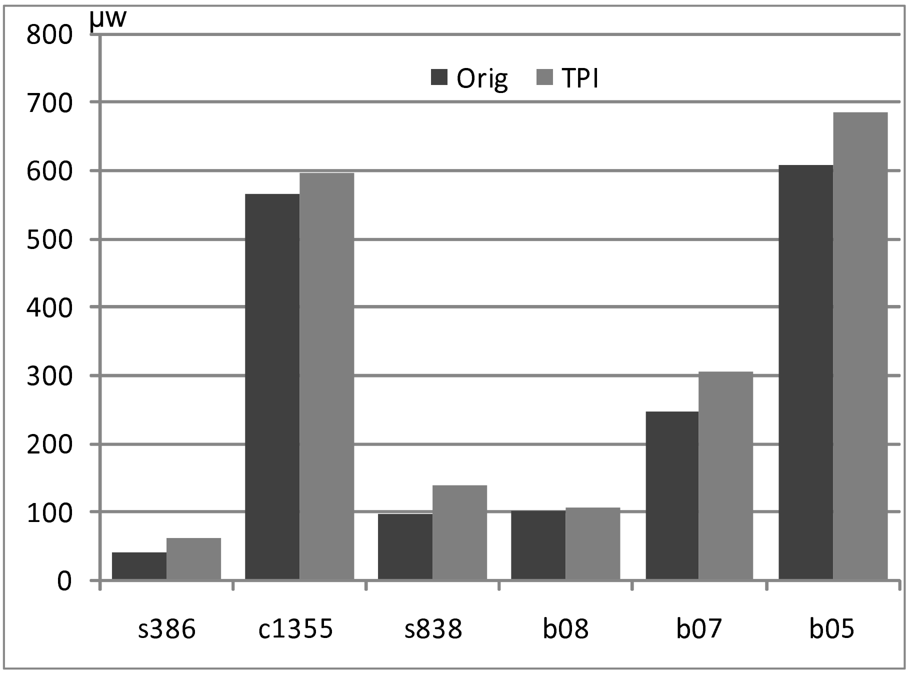


FIGURE 3.16: Impact of TPI on dynamic power in comparison with the original design.

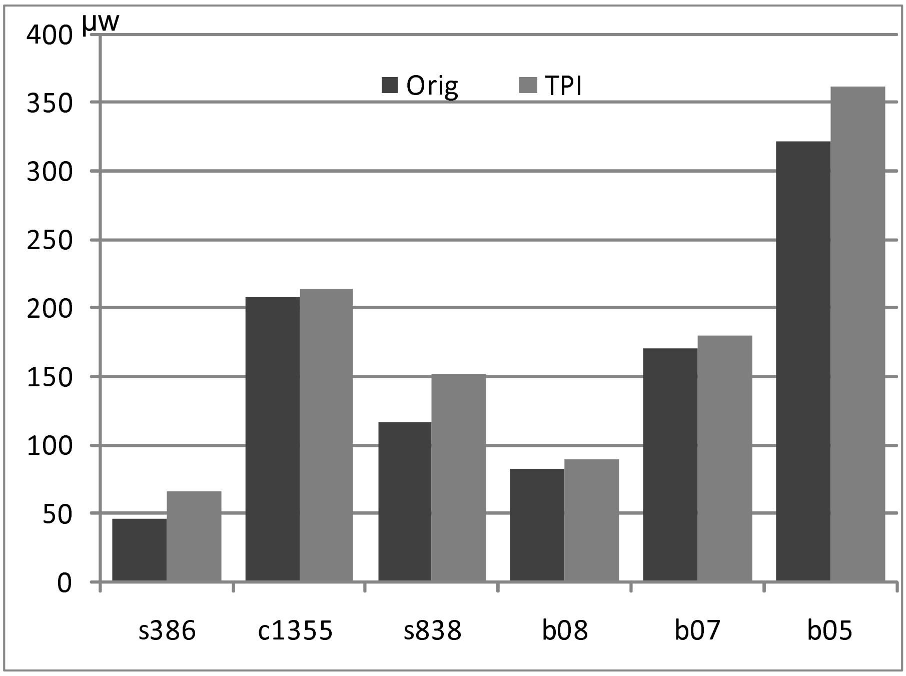


FIGURE 3.17: Impact of TPI on leakage power in comparison with the original design.

3.5 Concluding Remarks

Low power consumption and low cost manufacturing test are key constraints in today's competitive microelectronics industry. The employment of multi-V_{dd} design presents a number of challenges that need to be addressed to achieve high test quality at low cost. This chapter has addressed one of these challenges through test point insertion technique which can be employed to reduce the number of V_{dd} settings required during test without affecting the test quality.

This is the first investigation that addresses test cost reduction through minimizing the number of test V_{dd} settings for multi-V_{dd} designs. It demonstrates that test point insertion (TPI) can be used to reduce the number of V_{dd} settings during test, without affecting the fault coverage of the original test, thereby reducing test cost. Test points are used to provide additional controllability and observability at the fault-site to detect NRINEV intervals at essential V_{dd}, which are otherwise redundant (at essential V_{dd}) and therefore help reducing the number of test V_{dd} settings. A drawback with the TPI technique is that it does not guarantee a single V_{dd} test and resulted in more than one test V_{dd} setting for many designs. Other than that TPI has some well-known limitations (not limited to the proposed work) that to increase the fault coverage and to reduce test cost it may be necessary to introduce extra overhead on timing, area and power as is the case with [Abramovici et al., 1998, Touba and McCluskey, 1996, Pomeranz and Reddy, 1998] and demonstrated by the second experiment discussed in this chapter, Figure 3.14 to 3.17.

This chapter sets what is possible in terms of reducing test cost of multi-V_{dd} designs with bridge defects, by using a novel test point insertion technique. The next chapter aims to improve it further by targeting resistive bridges that cause faulty logic behavior, to appear at more than one test V_{dd} setting, and uses gate sizing (GS) to expose the same physical resistance of the bridge to minimize test cost.

Chapter 4

Test Cost Reduction Using Gate Sizing

4.1 Introduction

Resistive bridging faults (RBF) represent a major class of defects for deep submicron CMOS and can constitute 50% or more, of total defect count [Ferguson and Shen, 1988]. A bridge is defined as an un-wanted metal connection between two lines of the circuit, which may deviate the circuit from its ideal behavior. Resistive bridges have received increased attention on modeling, simulation and test generation [Renovell et al., 1996, Sar-Dessai and Walker, 1999, Engelke et al., 2004, 2006b, Renovell et al., 1999, Maeda and Kinoshita, 2000, Chen et al., 2005, Engelke et al., 2006a, Ingelsson, 2009]. Typically, a multi-V_{dd} design has a set of discrete supply voltage settings it can switch between depending on the current workload and power saving mode [Keating et al., 2007]. Manufacturing test needs to ensure that such a design operates correctly over the entire set of supply voltage settings, while keeping the overall cost of test low.

It has been shown in [Engelke et al., 2004] and more recently in [Ingelsson, 2009] that the fault coverage of a test set targeting RBF can vary with the supply voltage used during test. This means that, depending on the operating V_{dd} setting, a given RBF may or may not affect correct operation of the design. Consequently, to ensure high fault coverage for a design that needs to operate at a number of different V_{dd} settings, it is necessary to perform testing at more than one V_{dd} to detect faults that manifest themselves only at particular V_{dd}. It was shown in [Ingelsson, 2009] that the majority of circuits (8 out of 12) require testing at more than one voltage setting to achieve 100% fault coverage, which means that the ATE (Automatic Test Equipment) will have to switch between different voltage settings to apply the test. Switching between different

V_{dd} settings during test is not desirable and can impact the cost of test. Therefore it is important to reduce the number of test V_{dd} settings to one V_{dd} leading to reduction in test cost.

The only investigation that addresses test cost reduction through minimizing the number of test V_{dd} settings for multi-V_{dd} designs has been presented in Chapter 3 and published in [Khursheed et al., 2008]. It demonstrates that test point insertion (TPI) can be used to reduce the number of V_{dd} settings during test, without affecting the fault coverage of the original test, thereby reducing test cost. A drawback with the TPI technique (Chapter 3) is that it does not guarantee a single V_{dd} test and usually results in more than one test V_{dd} setting. In this chapter, a more effective technique is proposed for reducing test cost of multi-V_{dd} designs with bridge defects. It targets resistive bridges that cause faulty logic behavior, to appear at more than one test V_{dd} setting, and uses gate sizing (GS) to expose the same physical resistance of the bridge at a single test V_{dd}. The number of test voltages is then reduced, minimizing test cost. This chapter provides experimental results to show that unlike TPI, it is possible to achieve single V_{dd} test without affecting the fault coverage of the original test.

In this chapter, a gate sizing technique is presented with two different algorithms to identify bridges requiring multiple V_{dd} settings for detection. The first algorithm is *Deterministic* that utilizes only SAT-based test generation procedure [Ingelsson, 2009] to identify bridges that require multiple V_{dd} settings for detection and marks their driving gates for replacement. The second algorithm is *Probabilistic* that is motivated by an observation discussed in [Ingelsson, 2009] that SAT-based test generation can take up to 71% of total time inside SAT engine and attempts to reduce the number of times SAT engine is invoked thereby reducing computation time. These two algorithms present a trade-off between accuracy and speed; experimental results show an improvement of up to 50% in computation time. This chapter also evaluates, the impact on timing, area and power of the proposed technique, and comparison with the TPI shows that the proposed gate sizing technique performs better in terms of these three parameters. In comparison to the original design, the proposed technique has minimal impact on area and power, while timing has improved for many designs.

The chapter is organized as follows: Section 4.2 gives an overview of resistive bridge behaviour in multi-V_{dd} design. The motivation for using gate sizing to reduce the number of test V_{dd} settings is discussed in Section 4.3. Section 4.4 presents the proposed gate sizing technique. Experimental results are reported in Section 4.5, and Section 4.6 concludes the chapter.

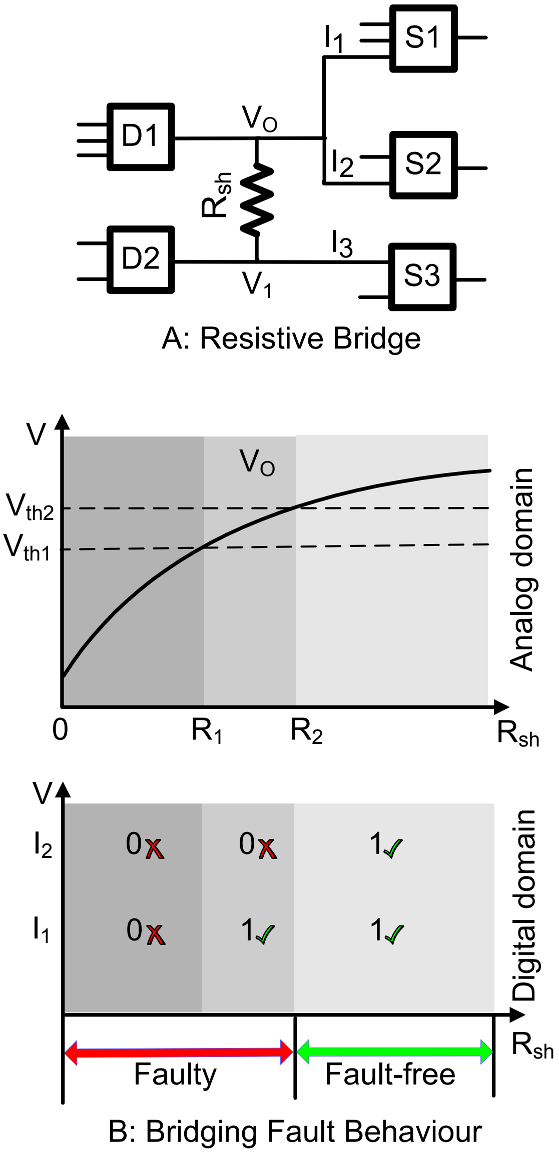


FIGURE 4.1: Bridge fault example and its behaviour in analog and digital domain

4.2 Preliminaries

To explain the proposed gate sizing technique, it is necessary to discuss some concepts related to resistive bridging faults and their behavior in the context of multi-V_{dd} designs. In this section, the behaviour of resistive bridge is briefly discussed, followed by an example to show, why TPI may not achieve single V_{dd} test for all designs, as demonstrated by the experimental results presented in Chapter 3.

A typical bridge fault behavior is illustrated in Figure 4.1. Figure 4.1-A shows a resistive bridge, D1 and D2 are the gates driving the bridged nets, while S1, S2 and S3 are successor gates; the output of D1 is driven high and the output of D2 is driven low. The dependence of the voltage level on the output of D1 (V_O) on the equivalent resistance of the physical bridge is shown in Figure 4.1-B (based on Spice simulation with 0.12μm library). To translate this analog behavior into the digital domain, the input threshold voltage levels V_{th1} and V_{th2} of the successor gates S1 and S2 have been added to the V_O plot. Each interval $[R_i, R_{i+1}]$ shown in Figure 4.1-B, corresponds to a distinct logic behavior occurring at the bridge fault site. This distinct logic behavior at the fault site is referred to as *Logic Fault* (or LSC as discussed in Sec. 2.1.1) and constitutes the following: boolean input to driving gates, resistance range coverage, $\mathrm{V_{dd}}$ setting and boolean values interpreted by driven inputs of successor gates.

Next, an explanation is provided to understand why test point insertion (TPI) does not guarantee single $\mathrm{V_{dd}}$ test for all designs, as demonstrated by experimental results presented in Chapter 3. Test points are used to provide additional controllability and observability at the fault-site to detect NRINEV[1] (Non-Redundant Intervals at Non-Essential $\mathrm{V_{dd}}$) intervals at essential $\mathrm{V_{dd}}$, which are otherwise redundant (at essential $\mathrm{V_{dd}}$) and therefore help reducing the number of test $\mathrm{V_{dd}}$ settings. TPI has shown reduction in the number of test $\mathrm{V_{dd}}$ setting(s) but it has some limitations. Experimental results presented in [Khursheed et al., 2008] show that the TPI is unable to reduce to single test $\mathrm{V_{dd}}$ for the majority of circuits (10 out of 13 circuits require more than one test $\mathrm{V_{dd}}$). This is because TPI cannot reduce the number of test $\mathrm{V_{dd}}$ setting(s) below the number of essential $\mathrm{V_{dd}}$ setting(s). This can be understood from the following explanation. In Figure 4.1-A, the gates used for driving the bridge (D1, D2) and the driven gates (S1, S2, S3) influence the number of essential $\mathrm{V_{dd}}$ setting(s) in a circuit. For the same circuit, assume that D1 is driving high and D2 is driving low, the output of D2 (V_1) on the equivalent resistance of the physical bridge is shown in Figure 4.2, which shows that higher resistance range is covered at 1.2V (non-preferred test $\mathrm{V_{dd}}$) than at 0.8V (preferred test $\mathrm{V_{dd}}$) as $R_{1.2V} > R_{0.8V}$. This means that 1.2V becomes essential test $\mathrm{V_{dd}}$ and TPI has to include it for 100% fault coverage, as the resistance range covered at 1.2V cannot be covered at 0.8V. The TPI has some limitations (not limited to the technique proposed in Chapter 3) that to increase the fault coverage and to reduce test cost it may be necessary to introduce extra overhead on timing, area and power as is the case with [Abramovici et al., 1998, Touba and McCluskey, 1996, Pomeranz and Reddy, 1998].

[1] The concept of NRINEV is discussed in detail in Section 3.1.

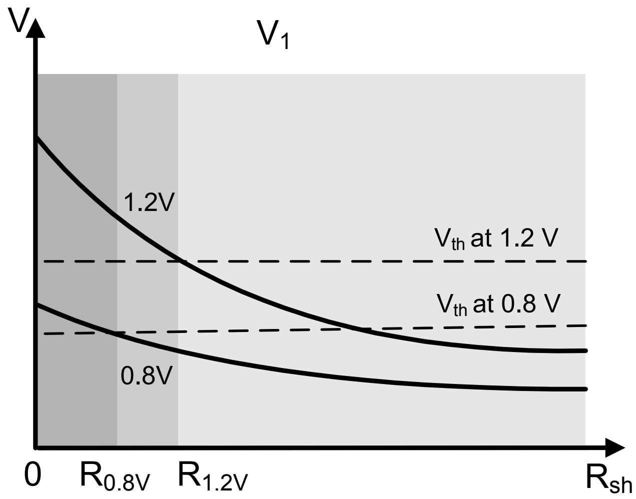


FIGURE 4.2: Resistance range detection at different voltage settings

4.3 Impact of Gate Sizing on Test V_{dd} Reduction

Gate sizing has been used to enhance timing performance of designs and more recently to tackle soft error rate in logic circuits [Zhou and Mohanram, 2006]. It was shown in [Rodriguez-Montanes et al., 2006] that bridges driven by gates with equal drive strength are likely to be detected at higher V_{dd} settings. We investigate the effect of gate sizing on the behavior of resistive bridging faults, and how it can be used to propagate faulty behavior, such that a higher physical resistance is exposed at a single V_{dd} setting (thereby reducing the number of essential test V_{dd} settings to one). The limitations of TPI can be addressed by adjusting the driving gates (D1, D2) or driven gates (S1, S2, S3) at the fault-site. The driving/driven gates can be adjusted by two approaches, which include the following:

- Modifying logic threshold of driven gates,
- Modifying drive strength of driving gates.

4.3.1 Modifying Logic Threshold of Driven Gates

In this case, the logic threshold of the driven gate is adjusted such that a higher resistance range is detectable at the lowest V_{dd} setting. This observation is further elaborated in Figure 4.3, where the logic threshold of the same gate inputs as for Figure 4.2, is reduced by gate-sizing. Therefore, the highest resistance interval is exposed at the lowest V_{dd} setting since $R_{0.8V} > R_{1.2V}$, which facilitates test generation at the lowest V_{dd} setting.

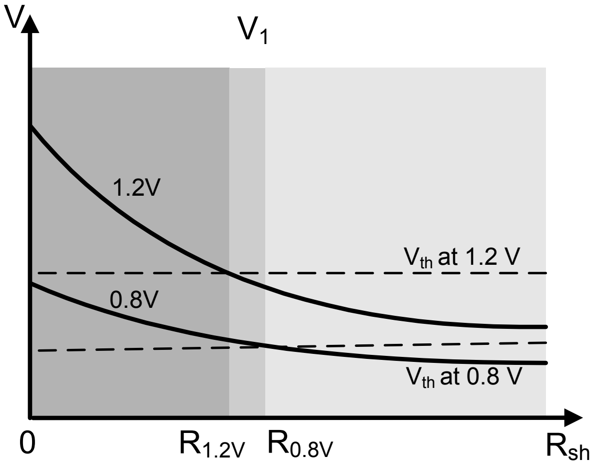


FIGURE 4.3: Resistance range detection after adjusting logic thresholds of the driven gates

The logic threshold can be adjusted by altering the width/length of the PMOS/NMOS transistor connected to the particular gate input, or by using the body bias effect. For an inverter it is given by [Weste and Eshraghian, 1994]:

$$V_{in} = \frac{V_{DD} + V_{tp} + V_{tn}\sqrt{\frac{\beta_n}{\beta_p}}}{1 + \sqrt{\frac{\beta_n}{\beta_p}}} \tag{4.1}$$

where, V_{in} is the voltage at the input of the gate, V_{DD} is the supply voltage, V_{tp} is the threshold voltage of the PMOS transistor, V_{tn} is the threshold voltage of the NMOS transistor.

$$\beta = \mu C_{ox}\left(\frac{W}{L}\right) \tag{4.2}$$

where, β is the MOS transistor gain factor, μ is the effective surface mobility of the carriers, C_{ox} is the gate oxide capacitance. From (4.1), it can be seen that a variation in W_p and W_n can alter the logic thresholds of a given gate input. This observation was used to conduct some experiments using 0.12μm ST Microelectronics library. The transistor widths (connected to the gate input of interest) are varied to reduce the logic threshold, while operating at 0.8V V_{dd}. For all the considered cases, the targeted change in logic threshold was -80 mV or less to detect the fault at the lowest V_{dd} setting, as that exposes higher resistance at the lowest V_{dd} setting. The resultant widths for some of the transistors are shown in Table 4.1, where the first column shows the gate for which the logic threshold is varied, followed by the (W_p/W_n) ratios of the original design and that of the re-designed gates. The last column shows the difference in logic thresholds

TABLE 4.1: Transistor width modification for altering logic threshold

Gate (Input)	W_p/W_n *		**Logic Th. Diff.**
	Original	**Re-designed**	**@ 0.8V V$_{\text{dd}}$**
4 Input NAND (C)	0.64/0.46	3.09/0.24	-80 mV
4 Input NAND (B)	0.64/0.46	5.22/0.24	-100 mV
5 Input AND-NOR (B)	0.94/0.64	6.79/0.24	-140 mV

* Width is in μm

as a result of gate-sizing. It can be seen that for all the cases the ratio between (W_p/W_n) is much higher than usually suggested design rule ratio of $(W_p/W_n) \approx 1.5 - 2.5$ [Weste and Eshraghian, 1994]. The ratios (in Table 4.1) result in unbalanced charging/discharging time (t_{phl} and t_{plh}) and violate design rules. For these reasons, modification of logic thresholds of the driven gates is not further considered to achieve single V$_{\text{dd}}$ test. Body biasing to vary the logic threshold was also examined but preliminary examination did not provide sufficient variations. For the cases considered, it resulted in $\approx$ 20 mV variation in logic threshold (operating at 0.8V V$_{\text{dd}}$) at the targeted gate input. Therefore logic threshold modification either by changing (width, length) ratios or by body biasing did not provide sufficient change in logic threshold voltages, and therefore these two methods are not pursued further to achieve single V$_{\text{dd}}$ test.

4.3.2 Modifying Drive Strength of Driving Gates

The drive strength of the gates driving the bridged nets can be adjusted to increase the voltage on the bridged nets (V_1 Figure 4.1-A, where D1 is driving high and D2 is driving low). This increase in voltage level can help expose higher resistance at the lowest V$_{\text{dd}}$ setting thereby reducing the number of essential V$_{\text{dd}}$ settings; additionally it can also be used to cover NRINEV (Non-Redundant Interval at Non-Essential V$_{\text{dd}}$) intervals at the lowest V$_{\text{dd}}$ setting. This concept is illustrated in Figure 4.4, which shows the same pair of bridged nets as Figure 4.2, i.e., the logic thresholds of the driven gates remain the same. It can be seen that the voltage level V_1 has increased such that $R_{0.8V} > R_{1.2V}$, as a result of increasing the drive strength of the gates driving the bridge. This means that during test pattern generation, logic fault at 0.8V will be targeted leading to single V$_{\text{dd}}$ test.

The drive current of an NMOS transistor operating in *active* region is [Weste and Eshraghian, 1994].

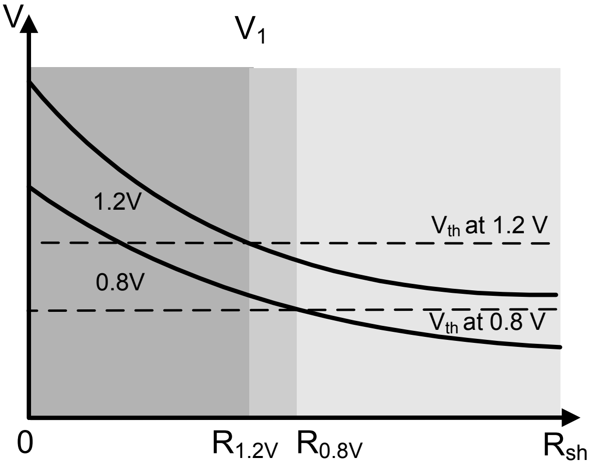


FIGURE 4.4: Resistance range detection after adjusting the drive strength of the gates driving the bridge

$$I_{ds} = \beta \left[(V_{gs} - V_t)\, V_{ds} - \frac{V_{ds}^2}{2} \right] \tag{4.3}$$

where, I_{ds} is the drain-source current, β is the gain factor expressed by (4.2), V_{gs} represents the gate-source voltage and V_t is the transistor threshold voltage.

From 4.3, it can be observed that the drive current I_{ds} is directly proportional to the gain factor β (in saturation and active modes), which in turn is directly proportional to the W/L of the transistor. Thus replacing a gate with another having higher value of β (for transistors feeding the output) results in higher drive strength. This is feasible since, different versions of functionally equivalent gates are usually available in the gate library.

An experiment is conducted to analyze the impact of increasing drive strength of gates driving the bridged nets on resistance coverage of bridge defects. For this purpose 10 circuits were synthesized using 0.12 μm STMicroelectronics gate library and Synopsys design compiler. A fault simulator and test pattern generator from [Ingelsson, 2009] is used to determine the detectable resistance range at three V$_{\mathrm{dd}}$ settings, i.e., 0.8V, 1.0V, and 1.2V. For each design, a bridge is inserted at a location that requires one or more V$_{\mathrm{dd}}$ setting for complete resistance coverage; unique resistance range at each V$_{\mathrm{dd}}$ setting is recorded that is not detectable at other V$_{\mathrm{dd}}$ settings. This is followed by replacing the gate with another having higher drive strength and repeating the procedure to determine the change in resistance coverage at each V$_{\mathrm{dd}}$ setting. The results are shown in Table 4.2. As can be seen, the resistance range for all the circuits has increased and for each design, 0.8V V$_{\mathrm{dd}}$ setting alone covers maximum resistance range, which

TABLE 4.2: Resistance range coverage at 3 different V_{dd} settings by increasing the drive strength of gates driving the bridge

Ckt	**Before Gate Sizing**			**After Gate Sizing**
	1.2V (Ω)	1.0V (Ω)	0.8V (Ω)	0.8V (Ω)
ISCAS-85 Benchmarks				
c432	0-1 kΩ			0-4.3 kΩ
c1355	0.9-1.3 kΩ		0-1.2 kΩ	0-4.5 kΩ
c1908	0-1.8 kΩ			0-6.3 kΩ
c2670	0.2-0.5 kΩ	0-0.4 kΩ		0-3.4 kΩ
c3540	0-0.6 kΩ			0-3.3 kΩ
ITC-99 Benchmarks				
b01	0.9-1.1 kΩ	0.8-1 kΩ	0-0.9 kΩ	0-4.3 kΩ
b02	0.5-1.5 kΩ		0-1.3 kΩ	0-4.6 kΩ
b03			0-7.3 kΩ	0-7.9 kΩ
b04	1.8-2.2 kΩ	2.2-2.6 kΩ	0-1.3 kΩ	0-8.3 kΩ
			2.9-3.3 kΩ	
b05	0-0.8 kΩ			0-1.7 kΩ

is not covered at any other V_{dd} setting. For instance, a bridge in the design c2670 covers 0 to 0.4 kΩ at 1.0V and 0.2 to 0.5 kΩ at 1.2V in original design. After increasing the drive strength of the driving gate the resistance range at 0.8V increased substantially from 0 to 3.4 kΩ; resistance coverage at 1.2V is covered completely at 0.8V and this is why it is not shown in the table. A similar trend is observed for the rest of the benchmarks shown in Table 4.2.

From this experiment two key observations are made:

- The detectable resistance range of a bridge defect can be increased by increasing the drive strength of driving gate. This is further shown in Figure 4.5, which shows higher defect resistance range is covered by replacing a gate (driving high, D1 as in Figure 5.1) with higher drive strength gate I_{ds2}, which is greater than I_{ds1},
- This increase is much higher at 0.8V than other voltage settings and for all the cases 0.8V alone captures the unique detectable resistance range.

These observations are exploited by the proposed gate sizing technique to achieve single V_{dd} test.

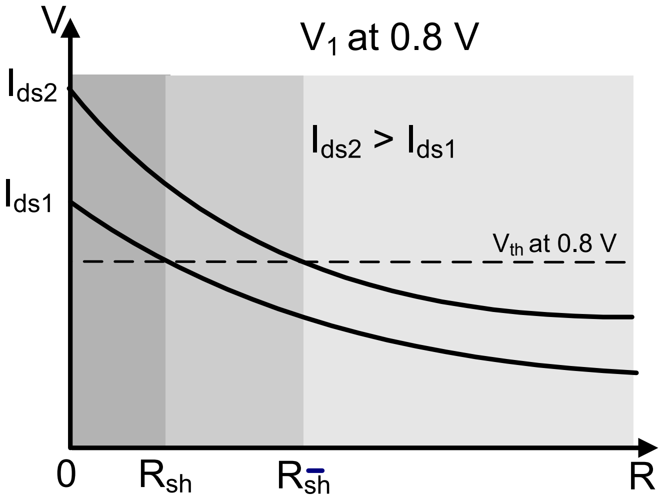


FIGURE 4.5: Higher drive strength results in increasing the covered resistance range of a bridge defect at same V_{dd} setting

4.4 Proposed Technique for Single V_{dd} Test

This section presents two gate sizing algorithms to reduce the number of test V_{dd} setting(s) for resistive bridge defect. Both algorithms consist of two phases: gate(s) identification and replacement, during which they identify the gates that should be replaced (for single V_{dd} test), followed by test generation phase on the modified circuit to achieve single V_{dd} test set. The process of gate identification for replacement distinguishes the two proposed algorithms. The first algorithm capitalizes on test generation method from [Ingelsson, 2009] to identify bridges that require more than one V_{dd} setting for complete fault coverage and is referred to as the *Deterministic Algorithm*. The second algorithm is based on a probabilistic method to identify bridge location(s) that may need more than one V_{dd} setting and is referred to as the *Probabilistic Algorithm*. The two algorithms show a trade-off between accuracy and speed as discussed in Section 4.5.

4.4.1 Deterministic Algorithm

The *Deterministic Algorithm*[2] (DA) is briefly described. It is included because the *Probabilistic Algorithm* (PA) uses the same flow and the two algorithms are compared in Section 4.5 demonstrating the trade-off between accuracy and speed.

[2] This algorithm was presented in our earlier publication in DATE 2009 and can be downloaded from http://eprints.ecs.soton.ac.uk/17047.

The algorithm (Figure 4.6) starts by test generation (test generation follows the method presented in [Ingelsson, 2009]) and marks all the bridges, which require test generation at higher than the lowest V_{dd} setting. All such bridges are placed in $TargetBridgeList$ and all the driving gates of the respective bridges are marked as potential candidates for gate replacement. The algorithm then solves a minimum set covering problem that identifies the minimum number of driving gates, such that all the bridges are covered. The selected gates are placed in $minGatesList$ (step-2). The algorithm then takes each selected gate in $minGatesList$ and replaces it with another having higher drive strength from the gate library (step 3-5). After updating the netlist, the algorithm generates a test set considering complete bridge list and finally returns with an updated netlist and a new test set.

```
Input: Netlist
Output: Test Set, Modified Netlist
1: Compute $TargetBridgeList$ by running test generation using the netlist
   // Mark the bridges that require test at additional
   // voltage setting(s)
2: Compute minimum number of driving gates $minGatesList$ across complete
   $TargetBridgeList$ by solving a minimum set cover
3: for all $minGatesList$ do
4:     Replace the selected gate with another having higher drive strength.
5: end for
6: Generate Test Set for the modified netlist using complete bridge list.
7: return $(Modified\,netlist,\ Test\ set)$
```

FIGURE 4.6: Deterministic Algorithm

4.4.2 Probabilistic Algorithm

This algorithm reduces run time to identify bridge locations for gate replacement. An experiment conducted using 12 different ISCAS-85 and ISCAS-89 benchmarks and the SAT-based test generator [Ingelsson, 2009] used in this work show that on average, 49% of total time spent during test generation is taken by the SAT engine [zChaff, 2007], and it can take as much as 71% of total time [Ingelsson, 2009]. The SAT-solver has exponential worst-case complexity [Ingelsson, 2009] and therefore the purpose of the *Probabilistic* algorithm is to restrict its usage thereby reducing run time. In the *Deterministic Algorithm* bridge locations for gate replacement are identified by invoking test generator [Ingelsson, 2009] in step-1, as shown in Figure 4.6. The *Probabilistic Algorithm* (PA) aims at reducing run time by selectively using the test generator (and therefore SAT-solver) and does not use it by default. The PA is invoked as step-1 of gate sizing technique (to compute $TargetBridgeList$) without affecting rest of algorithmic flow, shown in Figure 4.6.

As discussed in Section 4.2, a bridge defect consists of a number of logic faults at each $\mathrm{V_{dd}}$ setting; all logic faults per bridge constitute its fault domain. Since bridges requiring higher $\mathrm{V_{dd}}$ settings for detection are only targeted for gate replacement, the probabilistic algorithm categorizes the $\mathrm{V_{dd}}$ setting of each bridge location, which is used to decide whether gate replacement is required or not. Each bridge location is categorized by computing the probability of detecting logic faults at higher $\mathrm{V_{dd}}$ settings in comparison to those at the lowest $\mathrm{V_{dd}}$ setting. Therefore, a bridge with higher probability of fault detection at the lowest $\mathrm{V_{dd}}$ setting is not targeted for gate replacement. Probability based categorization and comparison of logic faults reduces the need of invoking test generator, thereby speeding up the gate sizing technique.

This is achieved by assigning a detection value (DV) to each logic fault in the fault domain that represents the probability of fault detection. It is assigned by computing probability of fault activation and fault effect observation at the output of gates fed by the bridge. It also takes into account the observability of a net by measuring minimum distance of each net from primary output(s) [Chandra and Patel, 1989]. The categorization of bridge defect to a specific $\mathrm{V_{dd}}$ is shown in Figure 4.7. It shows $\mathrm{V_{dd}}$ specific logic faults, with respective resistance range and detection value, where $DV \in [0, 1]$. Figure 4.7-a shows all logic faults of a bridge, including one at the highest $\mathrm{V_{dd}}$ setting (black bar) and the lowest $\mathrm{V_{dd}}$ setting (gray bar) with their respective DV. As can be seen, the resistance range covered at the highest $\mathrm{V_{dd}}$ setting has lower DV than 3 overlapping logic faults at the lowest $\mathrm{V_{dd}}$ setting. It means that the probability of this bridge to be detected at the highest $\mathrm{V_{dd}}$ setting is 3 times lower than that of the lowest $\mathrm{V_{dd}}$ setting. Similarly, a bridge resistance at the highest $\mathrm{V_{dd}}$ is shown in Figure 4.7-b, which shows the complete resistance range overlap by 2 logic faults, each with higher and lower DV, at the lowest $\mathrm{V_{dd}}$ setting. The Probabilistic algorithm uses this type of comparison to ease bridge identification (requiring gate replacement) without invoking computationally expensive test generator [Ingelsson, 2009].

Since logic circuits have different depths, topologies and design styles, a challenge is to establish a generic set of criteria to categorize bridges according to their $\mathrm{V_{dd}}$ setting – more importantly the criteria should hold on a wide variety of benchmarks and cover the worst case scenario for each design. For this reason, we performed a detailed analysis using 23 benchmarks, with various gate counts, design styles (ISCAS 85, 89 and ITC 99) and in total more than 110,000 bridge locations. After detailed analysis, a set of criteria is formulated to categorize a bridge to the lowest $\mathrm{V_{dd}}$ test setting. A bridge is referred to as *Low V_{dd} Bridge*, if its resistance ranges across all logic faults at higher $\mathrm{V_{dd}}$ settings are completely overlapped by those at the lowest $\mathrm{V_{dd}}$ setting, using one of the following criteria:

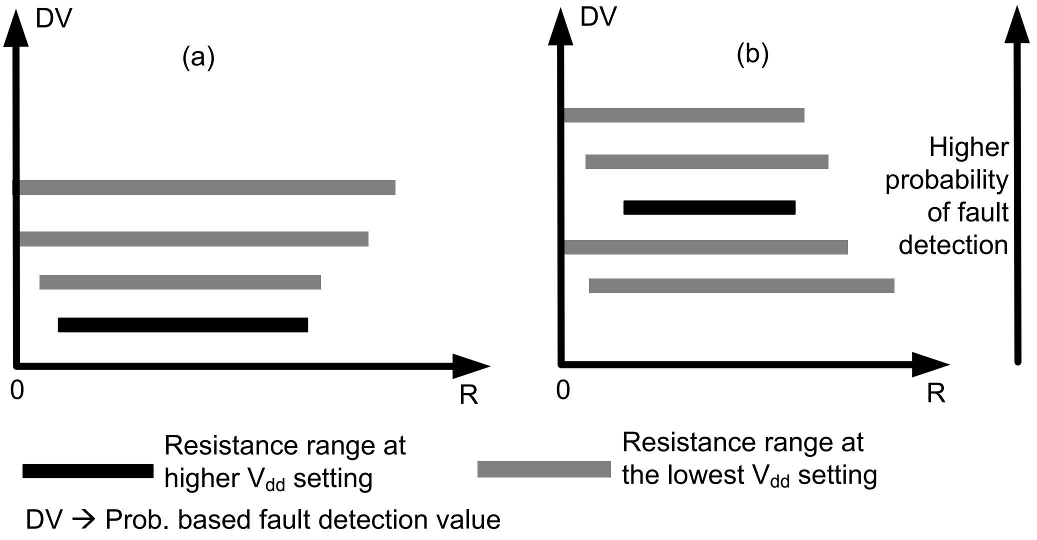

DV
(a)
0
R
DV
(b)
Higher probability of fault detection
0
R
Resistance range at higher V_{dd} setting
Resistance range at the lowest V_{dd} setting
DV → Prob. based fault detection value

FIGURE 4.7: Logic faults comparison using probability based detection value

1. Resistance range at higher V_{dd} is covered by at least 3 logic faults, at the lowest V_{dd} setting, with higher detection value, as shown in Figure 4.7-a.

2. Resistance range is covered by 2 logic faults with higher detection value, AND at least 2 logic faults with lower detection value, as shown in Figure 4.7-b.

3. Resistance range covered by 1 logic fault with higher detection value, AND at least 15 logic faults with lower detection value.

4. Resistance range covered by at least 20 logic faults with lower detection value.

The above listed criteria is developed after detailed experimentation using benchmarks shown in Table 4.3. The number of overlapping logic faults represent the worst case scenario over all bridges per design. It is used by the algorithm to cover a subset of bridges requiring the lowest V_{dd} test. Such bridge locations do not need gate replacement and therefore reduce the number of calls to the SAT-solver made by the test generator. The rest of the bridges are categorized as *Gray Zone* bridges as they may need gate replacement to achieve single V_{dd} test. Only for these bridges, the algorithm uses test generation to determine the exact V_{dd} test setting for detection of each such bridge. As a result of test generation, the Gray Zone bridges are categorized as either requiring High V_{dd} or Low V_{dd} test for detection. The above list of criteria serves as a useful filter to distinguish the bridges requiring High V_{dd} test and results in speeding up the process of gate identification for replacement by reducing the use of test generator.

To further evaluate the above criteria, experiments were conducted, using benchmarks shown in Table 4.3, with higher limits on the number of overlapping logic faults. In all cases, the experiments resulted in only converting Low $\mathrm{V_{dd}}$ bridges to Gray Zone bridges, without affecting those requiring High $\mathrm{V_{dd}}$ test. This and the above mentioned reasons indicate that the developed set of criteria for bridge identification, employed by the *Probabilistic* algorithm, is expected to hold on other benchmarks just as well. However, there is still a non-zero probability of missing out a bridge (Gray Zone bridge identified as low $\mathrm{V_{dd}}$) in another experimental setup. Such corner cases will be identified by step-6 of the algorithm presented in Figure 4.6 resulting in an additional iteration of the algorithm.

The Probabilistic Algorithm (PA) is shown in Figure 4.8. It uses signal probabilities to quantify the effort required by a logic fault for detection. In our experiments, signal probabilities are calculated by simulating pseudo-random patterns, however other analytical methods for estimating signal probability can be used for this purpose just as well. In a given circuit, signal probability per net is found by assigning a probability of 1(0) by carrying out logic simulations on the circuit using pseudo-random test patterns, until the probability of 1(0) do not change in last 200 iterations on any net. The number of iterations (200) is found by experimenting with different number of iterations from 50 to 300, and with 200 iterations, probability values are stable for all benchmarks. In step-3 of PA (Figure 4.8), the algorithm generates all logic faults per bridge and in step-4, it removes non-unique logic faults that are completely covered by identical logic fault at another $\mathrm{V_{dd}}$ setting. Two logic faults at different $\mathrm{V_{dd}}$ settings are identical if the input assignments to gates feeding the bridge are same along with the logic values interpreted by gates driven by the bridge. Such logic faults are distinguished by resistance range and $\mathrm{V_{dd}}$ setting at which it appears. This step reduces total candidate logic faults and is used to speed up the search and bridge categorization process. A probabilistic estimate of controllability and observability, referred to as detection value DV(LF), is computed for each candidate logic fault (steps 5 to 7) as follows:

$$DV(LF) = C(LF) \cdot O(LF) \tag{4.4}$$

where C(LF) is a probabilistic measure of the logic fault controllability, O(LF) is a probabilistic measure of observability of the fault at the outputs of gates fed by the bridge

$$C(LF) = \prod_{i=1}^{n}(Prob(i)) \tag{4.5}$$

where n is the cumulative number of inputs of the two gates driving the bridged nets and *Prob(i)* is the signal probability of logic value required by the LF on input i

```
Input: Netlist, Bridge locations
Output: Categorize all bridge locations to either of the two categories: Low V_dd, High V_dd
Compute signal probabilities on all nets
for all Bridge locations do
    Generate a list of logic faults candidates at each V_dd setting
    Retain unique logic faults at all V_dd settings
    for all LF candidates do
        Compute DV(LF)
    end for
    Sort all logic faults using their respective DV(LF)
    Categorize bridge location to either Low V_dd or Gray Zone
    Invoke test generator for Gray Zone Bridge and categorize it as either High V_dd or Low
    V_dd Bridge
    Update $(TargetBridgeList)$
end for
return $(TargetBridgeList)$
```

FIGURE 4.8: LF Ranking and Bridge Categorization of PA

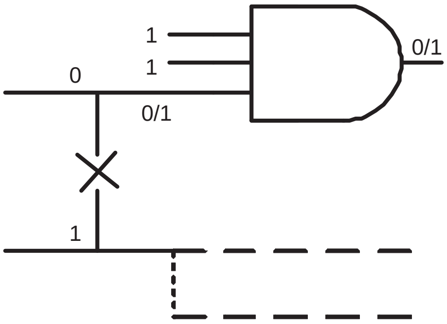


FIGURE 4.9: Observability calculation

$$O(LF) = \sum_{i=1}^{m} \frac{(f(X)) \cdot G_i}{D(PO)_i} \tag{4.6}$$

where m is the number of gates fed by the bridged nets, which propagate the faulty value to their outputs, G is the number of gates fed by each such gate, and $D(PO)$ is the minimum distance of fault observing gate fed by the bridge from primary output(s). $f(X)$ is the probability that the fault effect is propagated through gate X, computed as follows:

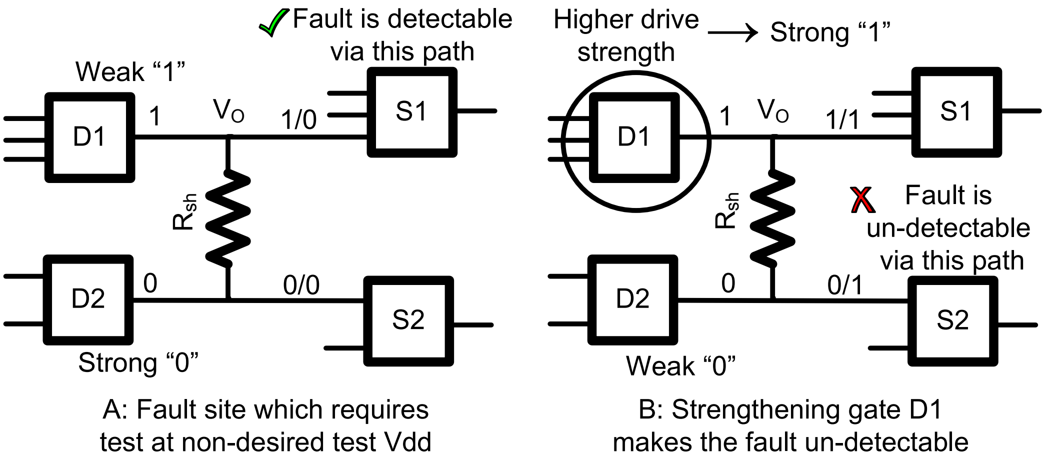


FIGURE 4.10: Fault Redundancy due to gate selection by minimum set cover

$$f(X) = \frac{\sum_{j=1}^{k}\prod_{i=1}^{l} SP_{i,j}}{2^l} \tag{4.7}$$

where k is the number of input combinations which propagate the fault effect to the output of successor gate X, l is the number of inputs of gate X which are not fed by the bridge, and $SP_{i,j}$ is the probability of having the value corresponding to input combination j on input i. For example, a 3-input AND gate fed by the bridge (as shown in Figure 4.9) there is one input configuration which will propagate the fault (0/1) to its output out of the 4 possible combinations on the two inputs which are not fed by the bridge. Assuming the "1" probabilities of the inputs which are not driven by the bridge to be 0.4 and 0.7 respectively, the probability of this gate propagating the fault to its output is $\frac{(0.4*0.7)}{4} = 0.07$.

Logic faults are sorted using their respective *Detection Value* (DV), and are then categorized into two different categories (Low V$_{\text{dd}}$ or Gray Zone) using the above mentioned set of criteria (Figure 4.7). For bridges that falls into "Gray Zone", test generator [Ingelsson, 2009] is invoked, which identifies exact V$_{\text{dd}}$ setting of each bridge location in Gray Zone. Bridges requiring higher V$_{\text{dd}}$ test are marked by $TargetBridgeList$, and this process is repeated for all bridge locations. Finally the PA returns to step-2 of the algorithm shown in Figure 4.6 with updated $TargetBridgeList$ that is used to compute minimum number of gates for replacement using set covering technique.

It should be noted that the minimum set covering technique (step-2, Figure 4.6) is useful for area minimization and has shown positive results for almost all the cases considered. However, in a few cases (less than 10), increasing the drive strength of a gate may make the fault redundant (un-detectable) at all V$_{\text{dd}}$ settings. This is explained using Figure 4.10, which shows a fault-site with driving gates D1 (driving high), D2 (driving low) while S1 and S2 are successor gates. Consider Figure 4.10-A and assume that the output of D1 is a weak “1” and the output of D2 is a strong “0”. This results in S1 reading a faulty logic value at its input (shown as 1/0), while S2 reads the correct logic value in both good/faulty circuits. Furthermore, assume that the fault effect is propagated to the primary output via S1 and results in test generation at a non-desired voltage setting. Now consider Figure 4.10-B, which shows that gate D1 is selected by the minimum set cover and is replaced by a gate with higher drive strength. Due to this change in drive strength, D1 outputs a strong “1” and D2 outputs a weak “0”, which results in S2 reading a faulty logic value (shown as 0/1) but this faulty logic value does not reach the primary output and therefore the fault becomes un-detectable. In such cases, the drive strength of both the driving gates (D1 and D2) is adjusted, such that higher resistance is exposed at the lowest $\mathrm{V_{dd}}$ setting (Figure 4.4) while ensuring that the fault is detectable. Therefore it is worth mentioning that for a few bridges, gate replacement and test generation may be repeated for fault detection at the lowest $\mathrm{V_{dd}}$ setting.

4.5 Experimental Results

The experimental setup used to validate the proposed gate sizing technique is similar to the one used to validate TPI (Chapter 3), which is briefly discussed next.

The proposed technique for reducing test $\mathrm{V_{dd}}$ settings is validated using ISCAS’85, ’89 and ITC 99 full scan circuits, see Appendix C for detailed description of all benchmark designs. The benchmark circuits are synthesized using ST Microelectronics 0.12 μm cell library. Synopsys Design Compiler^{TM} (DC) is used for synthesis, as well as, to evaluate timing, area and power. Default options of DC are used for synthesis without specifying any time constraints on any design. The generated netlist is then used for gate identification and replacement to achieve single $\mathrm{V_{dd}}$ test. The setup uses non-feedback bridges only and an exhaustive bridge list is generated by considering all possible pairs of nets in the netlist, up to a maximum of 10,000 pairs. All experiments are conducted using three $\mathrm{V_{dd}}$ settings: 0.8V, 1.0V, and 1.2V. The test generation flow used by the proposed gate sizing technique is shown in Figure 4.11. It should be noted that in case of TPI, as dicussed in Sec 3.4, a list of additional bridges is generated after inserting test points in the original design. However, this is not required in case of the GS technique because of the following two reasons: 1) the set up uses exhaustive bridge list, instead of coupling capacitance based extraction using physical layout; 2) the gates are only replaced and not added (as in case of TPI) by the proposed GS technique. The benchmarks used, total number of gates

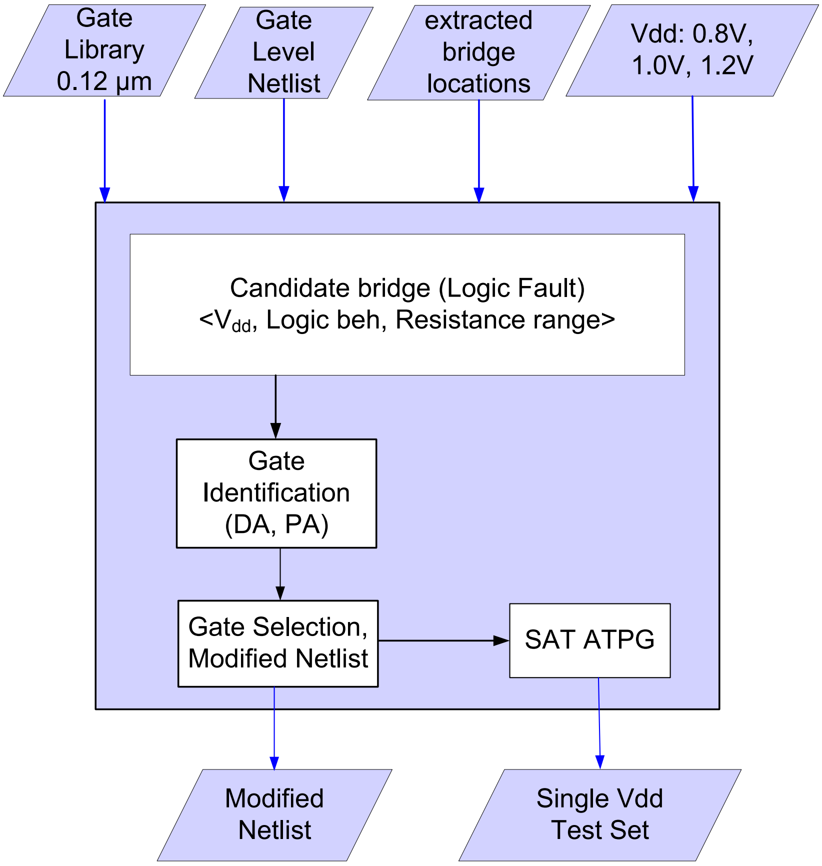

Gate Library 0.12 μm
Gate Level Netlist
extracted bridge locations
Vdd: 0.8V, 1.0V, 1.2V
Candidate bridge (Logic Fault)
<V_{dd}, Logic beh, Resistance range>
Gate Identification (DA, PA)
Gate Selection, Modified Netlist
SAT ATPG
Modified Netlist
Single Vdd Test Set

FIGURE 4.11: Test generation flow

and extracted bridges for each circuit are tabulated in Table 4.3. This setup is used to conduct two sets of experiments.

4.5.1 Test V_{dd} Reduction Using GS

These experiments have two objectives: firstly, to show the impact of the proposed gate sizing (GS) algorithms *Deterministic* and *Probabilistic*, to achieve single V_{dd} test. Secondly, to compare the two proposed algorithms in terms of the number of gates replaced and respective runtimes. Table 4.4 tabulates the total number of test V_{dd} setting(s) required by the original design (labeled as Orig.) and compares it with those generated by TPI [Khursheed et al., 2008]

TABLE 4.3: Benchmarks

ISCAS 85, ISCAS 89		
Ckt	# Gates	# Bridges
c432	93	1,094
c1355	226	6,563
c1908	205	7,986
c2670	269	10,000
c3540	439	10,000
c7552	731	9,998
s344	62	469
s382	74	1,146
s386	63	1,625
s838	149	5,737
s5378	578	9,933
s9234	434	10,000
s15850	1578	10,000

ITC 99		
Ckt	# Gates	# Bridges
b01	26	142
b02	15	33
b03	63	350
b04	208	7,228
b05	315	10,000
b06	33	203
b07	170	6,447
b08	86	1,350
b09	75	729
b10	88	1,923

(labeled, TPI)[3], and the proposed gate sizing algorithms (labeled DA, PA). As can be seen, the proposed algorithms (DA, PA) are able to achieve 100% fault coverage at a single test V_{dd}. This is unlike TPI, which requires two or more test V_{dd} settings for a large number of circuits. Furthermore, TPI is unable to reduce any test V_{dd}, in case of c432 and c1908. The last column of Table 4.4 shows the number of gates replaced by the proposed algorithms (DA, PA) and the number of test points (control/observation points) added by the TPI[4]. The number of gates replaced by the two algorithms ranges from 1-18, while the TPI has added up to 42 test points. The total number of gates replaced by the two GS algorithms (or added by the TPI) is shown in the last row of Table 4.4. The computation time of the proposed gate sizing algorithms is less than the TPI as it uses a simple set covering algorithm (Step-2, Figure 4.6) for reducing the number of gates to be replaced, while the TPI uses a complex control point minimization algorithm. The number of gates replaced by the PA is higher for certain circuits than the DA, as in case of c432. This is because of step-4 of the *Probabilistic* algorithm (PA) (Figure 4.8) that removes non-unique logic faults to speedup the algorithm. To investigate the increased gate count, a bridge in c432 is analyzed that is marked for gate replacement by the PA. The bridge has the following three logic faults: LF1@1.2V (0-1000 Ω), LF2@0.8V (0-800 Ω) and LF3@0.8V (800-1200 Ω). Furthermore, LF1 and LF2 are identical in terms of input assignments to the gates feeding the bridge and the logic values interpreted by the gates fed by the bridge. Since LF1 covers higher resistance than LF2, the algorithm removes LF2. With the removal of LF2, the bridge is marked for gate replacement, as 1.2V V_{dd} setting is required for complete resistance coverage.

The detectable resistance of neighboring nets (potential bridges) that may be affected by re-sizing of gates was analyzed by comparing the detectable resistance range before and after gate sizing. It was found that around 75% of the bridges sharing the net driven by the re-sized gates has their detectable resistance range increased, while the resistance range has reduced for the rest of 25% bridges, however it is always $\geq$ 1KΩ of detectable resistance after re-sizing. These bridges are not further re-sized because it was reported in Rodriguez-Montanes et al. [1992] that around 96% of the bridges have their resistance range $\leq$ 1KΩ, however the proposed gate sizing technique can be repeated for such bridge locations, if higher detectability is required. The detectable resistance range is increased for a large majority of bridges because a bridge location consists of a large number of logic faults, where total number of logic faults depends on the number of possible combinations to activate the bridge and the number of gates fed by the bridge. For each bridge location, the test generator determines the total detectable resistance range using all possible logic faults. Therefore resistance range covered by an individual logic fault is less important than the total detectable resistance considering all logic faults. From

[3] TPI results may vary from those reported in [Khursheed et al., 2008] because of using different logic threshold values

[4] The number of test points is the sum of control and observation points

the experimental results, it is evident that the proposed gate sizing technique guarantees single V_{dd} test for all designs, while increasing the detectable resistance range for a large majority of bridges.

Table 4.5 shows the categorization of bridges to *Low V_{dd}* and *Gray Zone* by step-9 of the *Probabilistic* algorithm (Figure 4.8). As can be seen, for all the circuits, on average 45% and up to 71.6% of total bridges are identified as “Low V_{dd}” without using computationally expensive (SAT-solver based) test generator. These bridges are accurately identified by using probability based bridge identification criteria (Figure 4.7).

To get an insight into the computation time of the proposed algorithms *(Probabilistic, Deterministic)*, see Table 4.6, which shows the comparison of total number of SAT calls and run-time of the two algorithms. The *Probabilistic* algorithm has significantly reduced the total number of computationally expensive SAT calls, for all benchmark designs, and on average it achieves 2.6X reduction in the total number of SAT calls in comparison to the *Deterministic* algorithm. The run-time (of PA and DA) is shown in column 3 of Table 4.6, and the last column shows the relative run-time by the PA in comparison to the DA. The last two rows show the sum and average of the number of SAT calls and run-time for all designs.

As can be seen, the PA results in a significant speed up for a large majority of circuits (upto 50% time reduction, in case of c2670), this is especially noticeable for larger circuits for e.g., b04, c2670, c3540, c7552, s9234, and s15850 that show significant speed up. However, because of the setup time of PA (step-1 and steps 5-7 shown in Figure 4.8), it is more time efficient for larger designs and smaller designs do not show improvement, as is the case with s382, s386, b01, b08, and b10. The results presented in Table 4.6 has shown encouraging results in terms of reducing SAT calls and minimizing run-time, which is further elaborated by Figure 4.12 that shows the comparison of the number of SAT calls made by the two algorithms (PA and DA).

TABLE 4.4: Results of the proposed Gate Sizing algorithms (DA, PA) and comparison with TPI presented in Chapter 3.

	Test V_{dd} settings			No. of Gates		
Ckt	Orig.	TPI (Chp 3)	DA, PA	DA	PA	TPI (Chp 3)
c432	All*	All	0.8V	2	3	0
c1355	All	0.8V	0.8V	4	4	10
c1908	1.2V, 0.8V	1.2V, 0.8V	0.8V	3	3	0
c2670	All	1.2V, 0.8V	0.8V	6	6	19
c3540	All	1.0V, 0.8V	0.8V	7	8	7
c7552	All	0.8V	0.8V	1	1	1
s344	1.2V, 0.8V	0.8V	0.8V	1	1	1
s382	1.2V, 0.8V	0.8V	0.8V	2	2	5
s386	All	1.2V, 0.8V	0.8V	7	7	4
s838	All	0.8V	0.8V	14	14	28
s5378	All	1.0V, 0.8V	0.8V	9	12	9
s9234	All	1.0V, 0.8V	0.8V	6	13	2
s15850	All	0.8V	0.8V	8	9	3
b01	All	0.8V	0.8V	1	1	1
b02	1.2V, 0.8V	0.8V	0.8V	1	1	2
b03	0.8V	0.8V	0.8V	0	0	0
b04	All	0.8V	0.8V	8	8	4
b05	All	0.8V	0.8V	18	18	42
b06	0.8V	0.8V	0.8V	0	0	0
b07	All	1.2V, 0.8V	0.8V	9	10	10
b08	All	0.8V	0.8V	4	4	8
b09	1.2V, 0.8V	0.8V	0.8V	2	2	2
b10	All	0.8V	0.8V	4	5	5
Total No. of Gates				117	132	163

*All = 0.8V, 1.0V, 1.2V

PA → Probabilistic Algorithm, DA → Deterministic Algorithm

TABLE 4.5: Bridge categorization by the Probabilistic algorithm

		Prob. Search Space	
Ckt.	# Bridges	Gray Zone	Low V_{dd}
c432	1,094	339	755
c1355	6,563	3762	2,801
c1908	7,986	4776	3,210
c2670	10,000	2842	7,158
c3540	10,000	3282	6,718
c7552	9,998	6203	3795
s344	469	234	235
s382	1,146	803	343
s386	1,625	751	874
s838	5,737	3916	1821
s5378	9,933	4886	5047
s9234	10,000	5363	4637
s15850	10,000	5899	4101
b01	142	78	64
b02	33	21	12
b03	350	195	155
b04	7,228	3497	3,731
b05	10,000	4468	5,532
b06	203	148	55
b07	6,447	3489	2,958
b08	1,350	860	490
b09	729	542	187
b10	1,923	1189	734

TABLE 4.6: Timing Comparison of Deterministic and Probabilistic Algorithms

	Total SAT runs		Time (min)		
Ckt.	PA	DA	PA	DA	$\frac{PA}{DA}$
c432	1816	7379	1.52	1.93	0.78
c1355	5821	19128	24.37	28.88	0.84
c1908	8940	13766	23.83	25.52	0.93
c2670	7416	50488	117.68	237.33	0.50
c3540	10790	44908	75.62	135.75	0.56
c7552	18454	32877	225.25	396.95	0.57
s382	1363	2119	1.65	1.28	1.29
s386	2190	7770	2.35	2.27	1.04
s838	6187	14586	19.82	22.28	0.89
s5378	9450	31269	310.00	336.20	0.92
s9234	12669	37064	723.60	947.60	0.76
s15850	12580	20598	4513.1	5896.18	0.77
b01	166	338	0.05	0.02	3.0
b02	44	63	0.02	0.02	1.0
b04	6884	13803	33.78	41.15	0.82
b07	8527	25631	29.32	31.45	0.93
b08	3697	9883	2.08	1.93	1.08
b10	4601	8258	2.03	1.55	1.31
Total	121595	339928	6106.1	8108.3	0.75
Avg.	6755.3	18884.9	339.2	450.5	0.75

PA → Probabilistic Algorithm, DA → Deterministic Algorithm

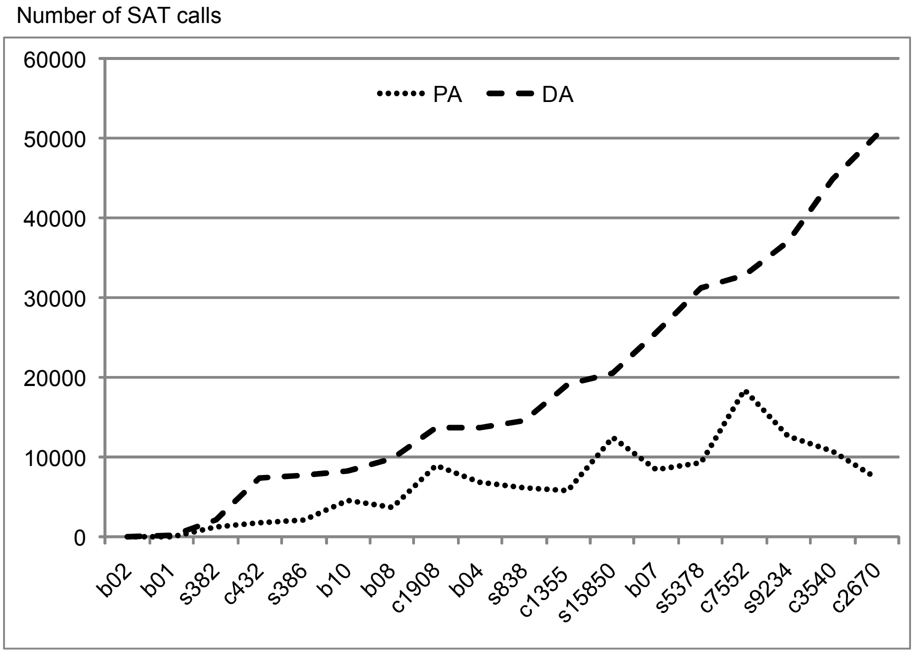


FIGURE 4.12: Total number of SAT calls: DA v PA.

4.5.2 Impact on Timing, Area and Power

The second set of experiments compares timing, area and power (dynamic and leakage) of the original design, the proposed Gate Sizing, and the TPI. Figure 4.13 shows the timing performance. As can be seen, the proposed GS technique has little effect on timing when compared to the original design. This is because it replaces small number of gates. On average, for circuits shown in Table 4.4, it has replaced only 3% of the total number of gates. For some circuits the proposed GS technique has improved timing due to larger and faster gates. This is unlike the case with the TPI, where the timing was negatively affected because of the test points inserted in the critical path. For example, in case of s386, TPI has inserted 1 test point, and converted a non-critical path into a critical path, while the GS technique has replaced a gate with a bigger gate, in the critical path, resulting in reduced timing. On comparing the delay of the longest path in the original and the GS modified designs, it was found that the longest path in the original design has a delay of 0.85 ns. On the other hand, the GS has replaced a gate in the longest path with a bigger gate thereby reducing the delay of the longest path to 0.78 ns (from 0.85 ns in the original design). As a result the second longest path in the original design with a delay of 0.82 ns, became the longest path in the GS modified design.

Similarly, comparison of area overhead is shown in Figure 4.14 for the three designs. The proposed GS technique results in a slightly higher area overhead in comparison to original designs; however, it is less than the TPI for all circuits. Finally, comparison of dynamic and leakage power is shown in Figure 4.15 and Figure 4.16 respectively. It can be seen that the proposed gate sizing technique slightly increases the power budget in comparison to the original design; however, it is less than the TPI in all cases. High power consumption of the TPI is because of additional switching activity and leakage power of added test points. In case of the GS, switching activity does not change in comparison to the original design but load capacitance and leakage power increases due to bigger gates, leading to higher dynamic and leakage power. The impact on leakage power can be reduced by using high-V_t transistors in non-critical paths of the design [Keating et al., 2007].

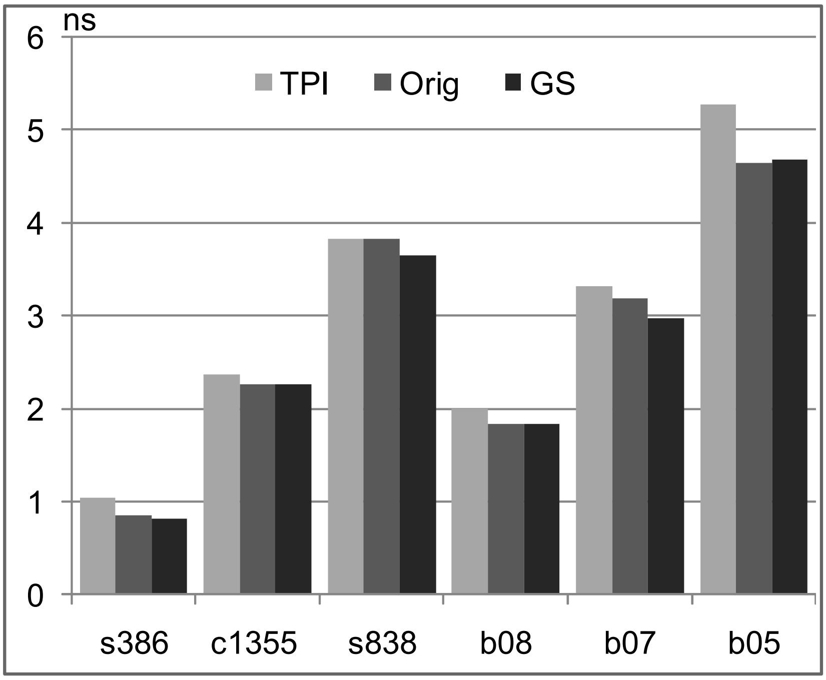


FIGURE 4.13: Impact of Gate Sizing on timing performance and comparison with the original and the TPI presented in Chapter 3.

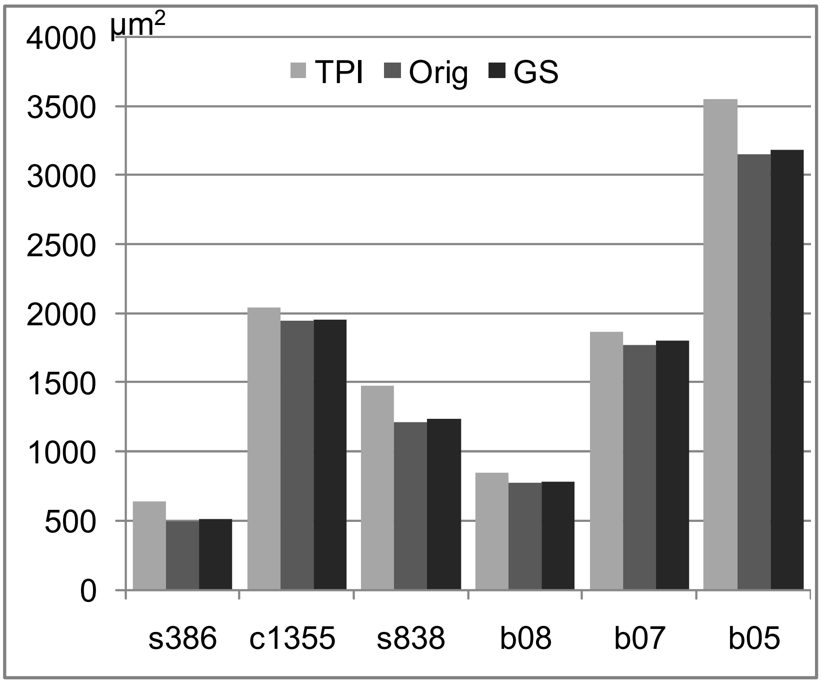


FIGURE 4.14: Impact of Gate Sizing on area overhead and comparison with the original and the TPI presented in Chapter 3.

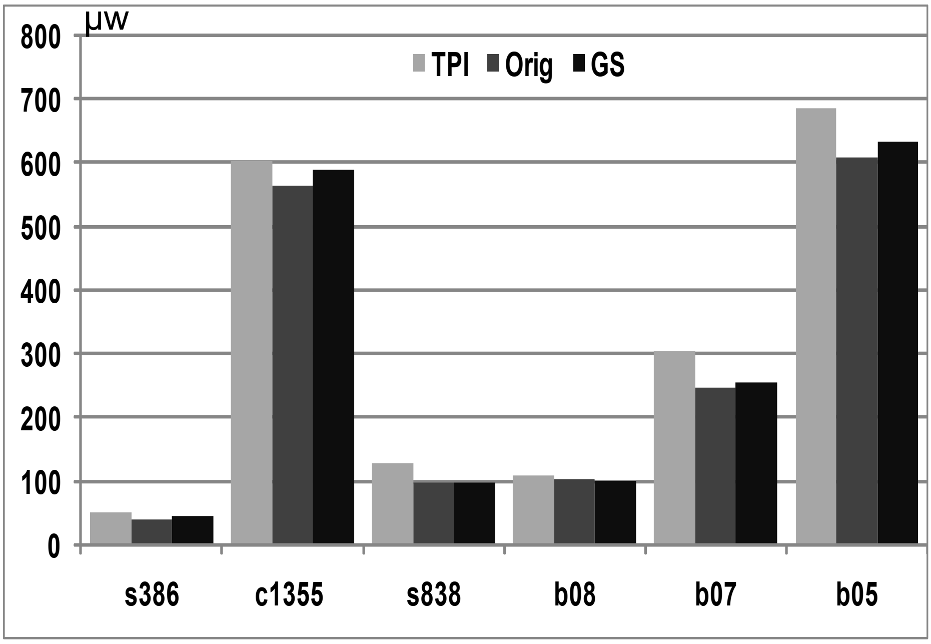


FIGURE 4.15: Impact of Gate Sizing on dynamic power and comparison with the original and the TPI presented in Chapter 3.

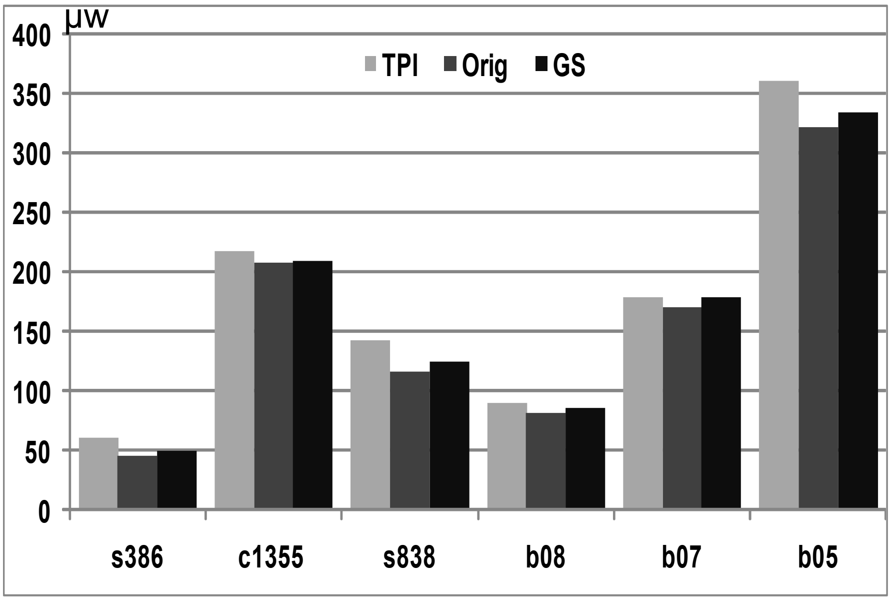


FIGURE 4.16: Impact of Gate Sizing on leakage power and comparison with the original and the TPI presented in Chapter 3.

4.6 Concluding Remarks

Resistive bridge defects (RBDs) show Vdd dependent detectability and research studies conducted across the world show that the lowest Vdd setting achieves highest fault coverage. Nevertheless this class of defects requires more than one Vdd setting to achieve 100% fault coverage. The Vdd dependent detectability of RBDs represent new problems for existing DFT solutions, i.e., it may be necessary to repeat the test at more than one voltage setting to achieve 100% fault coverage. Test repetition at more than one voltage setting is un-desirable because of its negative implication on test cost[5]. This chapter has addressed this problem by achieving single V_{dd} test for RBDs without affecting the fault coverage of the original test set.

In this chapter, gate sizing technique is proposed to reduce test cost of multi-V_{dd} designs with bridge defects, by reducing the number of test voltage settings. It has been shown, that it is possible to achieve 100% fault coverage using a single V_{dd} test setting. This represents an improvement on the TPI technique (presented in Chapter 3 and published in [Khursheed et al., 2008]) which mostly requires two or more test V_{dd} settings to achieve complete fault coverage. In this chapter, two algorithms are presented to identify gates for replacement and to achieve single V_{dd} test, these algorithms show a trade-off between accuracy and speed. The proposed gate sizing technique has little effect on timing, area and power when compared with the original design (prior to gatesizing) and performs better than the TPI in terms of these three parameters.

The Vdd dependent detectability of resistive bridge defects not only affects existing DFT solutions but also questions the existing diagnosis techniques, as all existing diagnosis techniques use single Vdd test setting for diagnosing such defects. This may lead to reduced diagnosis accuracy with negative affect on failure analysis, which is key to improving subsequent design cycle and yield. This means that novel diagnosis solutions are required for accurate and cost-effective diagnosis of bridge defects in multi-Vdd designs, which is the aim of the next chapter.

[5] See Section 3.1 for illustrative example and more details on the impact of multi-Vdd testing on test cost.

Chapter 5

Bridge Defect Diagnosis

5.1 Introduction

Diagnosis is a systematic way to uniquely identify the defect causing malfunction in the circuit. It is critical to silicon debugging, yield analysis and for improving subsequent manufacturing cycle. There has been extensive work on modeling, detection and diagnosis of bridge defects [Abramovici and Breuer, 1980, Waicukauski and Lindbloom, 1989, Millman et al., 1990, Pomeranz and Reddy, 1992, Wu and Rudnick, 1999, Arslan and Orailoglu, 2003, Zou et al., 2005, Rousset et al., 2007, Holst and Wunderlich, 2007, Gattiker, 2008, Pomeranz and Reddy, 2008]. However these works implicitly consider only designs using a single supply voltage V_{dd}. Many modern processors allow use of multiple V_{dd} settings, which can be dynamically selected to reduce power consumed and still meet the computational requirements [Martin et al., 2002] and [Intel, 2007]. Thus it is important to investigate the effect and potential advantage of using multiple V_{dd} settings to improve diagnosis accuracy for such designs.

A bridge is defined as an un-wanted metal connection between two lines of the circuit, which may deviate the circuit from its ideal behavior. In considering diagnosis of bridge defects we used a cause-effect diagnosis procedure which uses dictionaries [Abramovici et al., 1998][1]. The amount of information stored in a dictionary is a trade off between storage space and diagnostic resolution. A study reported in [Narayanan et al., 1997] compares these parameters for full response dictionary (that holds the detailed output response for each fault per test vector), pass-fail dictionary (which stores one bit, indicating pass or fail of a test, per test per fault) and frequency based dictionary (that holds the detection count of each fault over the entire test set).

[1] Section 1.5 provides more details on different diagnosis techniques other than cause-effect diagnosis technique.

The study shows that pass-fail dictionary provides high diagnostic power (much higher than frequency based dictionary but slightly lower than full response dictionary) and higher space compaction (much higher space compaction than full response dictionary). Therefore in order to conserve storage requirements for the dictionaries we used a pass-fail dictionary [Pomeranz and Reddy, 1992]. However conclusions drawn through the experiments reported in this work are expected to hold if other diagnosis procedures are used (including full response dictionary or effect-cause diagnosis procedure [Abramovici and Breuer, 1980, Abramovici et al., 1998]).

A study comparing between better fault models or better diagnosis algorithms revealed that using a simple diagnosis algorithm on a better fault model achieves higher diagnosis accuracy [Aitken and Maxwell, 1995]. It was shown by Zou *et al.* [Zou et al., 2005] that using an advanced parametric bridge fault model [Renovell et al., 1995, 1996], diagnosis resolution can improve over algorithms that use simpler fault models. This work also uses the same parametric fault model [Renovell et al., 1996].

The nature of bridge defects in multi-V_{dd} designs is such that they manifest themselves at one or more voltage settings [Engelke et al., 2004, Ingelsson, 2009]. Existing diagnosis techniques use a single V_{dd} setting and therefore diagnosis for multi-V_{dd} designs imposes a challenge as bridge defects exhibit supply voltage dependent behavior. Single V_{dd} diagnosis for multi-V_{dd} designs may lead to imprecise diagnosis as shown by experimental results (Section 5.5) of this work. Furthermore, it raises the following questions: 1) Is diagnosis resolution affected by different voltage settings? 2) If so, what voltage setting achieves the best level of diagnosis? 3) Is it possible to improve diagnosis resolution further by carrying out diagnosis at more than one voltage setting? 4) For designs operating at more than one voltage setting, it is desirable to reduce diagnosis cost by achieving the minimum possible Test Application Time (TAT), without affecting diagnosis accuracy. Therefore, it is important to determine the most useful V_{dd} settings or combination of V_{dd} settings, which may yield the desired outcome by omitting tests at some voltage settings.

This is the first reported work to consider diagnosing bridge defects in multi-V_{dd} designs and present results to show that the lowest supply voltage provides the best resolution for single voltage diagnosis. This work further exploits the additional information from other voltage settings to improve the diagnosis accuracy up to 72% over single voltage diagnosis. We also analyse hard-shorts (bridges with 0 Ω resistance) and experimental results show that diagnosis accuracy has little variation across different voltage settings for this class of defects. Finally, we show experimental results using different V_{dd} pairs and identify the most useful V_{dd} pair, such that high diagnosis accuracy is achieved using reduced TAT, thereby reducing diagnosis cost.

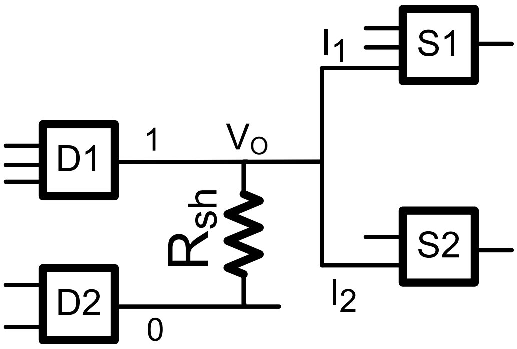
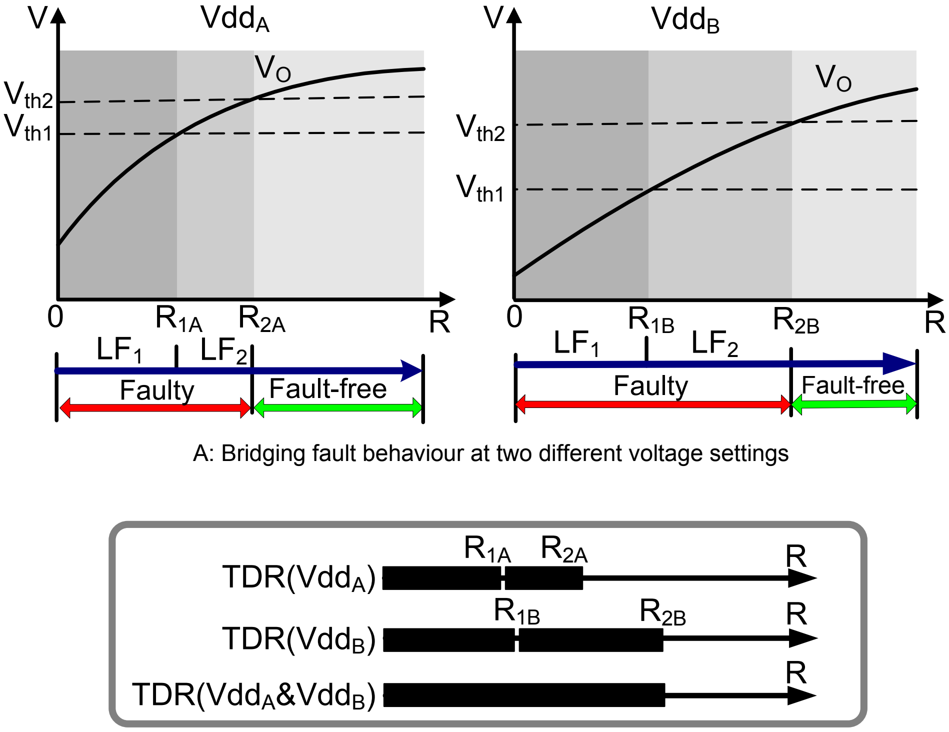


FIGURE 5.1: Resistive bridge

The chapter is organized as follows: Section 5.2 gives an overview of resistive bridge defects and their behavior in the context of multi-V$_{\text{dd}}$ design. The motivation for multi-V$_{\text{dd}}$ diagnosis is discussed in Section 5.3. In Section 5.4 we present a multi-V$_{\text{dd}}$ diagnosis algorithm for bridge defects. Experimental setup and results are reported in Section 5.5, and finally Section 5.6 concludes the chapter.

5.2 Preliminaries

Figure 5.1 shows a resistive bridge, D1 and D2 are the gates driving the bridged nets, while S1 and S2 are successor gates, i.e., gates having inputs driven by one of the bridged nets. Let us consider the case when the output of D1 is driven high and the output of D2 is driven low. The dependence of the voltage level on the output of D1 (V_O) on the equivalent resistance of the physical bridge is shown in Figure 5.2, for two supply voltage settings (based on Spice simulation with 0.12μm library). Figure 5.2-A show the relation between the voltage on the output of gate D1 (Figure 5.1) and the bridge resistance for two different supply voltages Vdd_A and Vdd_B. Figure 5.2-A also shows how the analog behavior at the fault site translates into the digital domain. We can see that two distinct Logic Faults LF1 and LF2 can be identified for each V$_{\text{dd}}$ setting. Figure 5.2-B shows the Total Detectable Resistance (TDR) for the LFs detected at two voltage settings separately and combined as well. This V$_{\text{dd}}$ behavior of defect also means that a test pattern targeting a particular logic fault will detect different ranges of physical defects when applied at different supply voltage settings. For example, at Vdd_A, a test pattern targeting LF2 will detect bridges with $R_{sh} \in [R_{1A}, R_{2A}]$, while at Vdd_B it will detect a much wider range of physical bridges ($R_{sh} \in [R_{1B}, R_{2B}]$). Furthermore, this means that same defect can be covered at more than one voltage setting.



A: Bridging fault behaviour at two different voltage settings

B: Total Detectable Resistance (TDR)

FIGURE 5.2: Effect of supply voltage on bridge fault: Analog/Digital domain

A sub-class of resistive bridging faults is hard-short, which is observed when the nets connected with one another are at 0 Ω. The behavior of hard-shorts in the context of multiple voltage settings can be understood from Figure 5.1 and Figure 5.2. In Figure 5.1, since the value of R_{sh} is 0 Ohms, the logic behavior at the fault site does not vary at two different $\mathrm{V_{dd}}$ settings (LF1 at both $\mathrm{V_{dd}}$ settings). In general, this similarity in logic behavior at two $\mathrm{V_{dd}}$ settings suggests that fault detection (for hard-shorts) may have lesser dependence on voltage setting used, in comparison to bridges with higher resistance values.

From a diagnosis point of view it is interesting to analyze the impact of covering the same defect (specially, bridges with higher resistance values) at more than one voltage setting and to analyze its effect on diagnosis resolution, i.e., can it help to improve the diagnosis resolution over single voltage diagnosis? The next section uses illustrative examples to show that combining the information gathered by diagnosing at different voltage settings may help improve the diagnosis accuracy over single voltage diagnosis.

5.3 Motivations for Multi-V_{dd} diagnosis

The last two chapters (Chapter 3 and Chapter 4) have shown how existing DFT solutions are affected by the V_{dd} dependent behaviour of resistive bridges. It gives the motivation to re-evaluate existing diagnosis strategies as all of them use single V_{dd} test setting for diagnosis. This may lead to reduced diagnosis accuracy and negative impact on failure analysis.

This section presents two illustrative examples to highlight the possible improvement in diagnosis by carrying it out at multiple voltage settings, using a simple pass/fail test. As discussed in section 5.2, defects caused by a resistive bridge consists of resistance interval(s) detectable at one or more voltage settings. The resistance range (at each voltage setting) corresponds to a faulty logic behavior in digital domain. Total detectable resistance for the bridge comprises of union of resistance intervals detectable at each voltage setting. This is further elaborated in Figure 5.3, which shows two bridge locations (BL-A and BL-B) in a circuit structure similar to the one shown in Figure 5.1 and is found by using the same mechanism as for Figure 5.2 using three voltage settings. Figure 5.3 shows the V_o behavior of bridges at three different voltage settings in analog domain and corresponding logic faults marked by TDR(V_1), TDR(V_2), and TDR(V_3) respectively. It should be noted that two logic faults exist for each bridge at each voltage setting (shown by TDR(V_1) etc), but only one is assumed to be detectable. Logic faults shown in Figure 5.3 are magnified and re-drawn in Figure 5.4, which shows the total detectable resistance for the two bridges by combining information from all three voltage settings. For instance, in case of BL-A, resistance range marked by interval-A is detectable at V_1 only, similarly resistance range marked by interval-B is detectable at both V_1 and V_2.

The illustrative examples show the possible improvement by multiple voltage diagnosis over single voltage diagnosis. The two examples inject two different defects and are based on the following assumptions: 1) Single defect can be active at a given time. 2) There is only one Failing Pattern (FP) in the diagnostic test set, which detects the two defects. Figure 5.4 shows all the intervals that are detectable at different supply voltages by the same FP. Table 5.1 maps the Detected/Not-Detected (D/ND) status of all intervals shown in Figure 5.4 for the two bridges.

5.3.1 Combining Diagnosis Information

In the first case, we inject a defect consisting of resistance value from interval *C* of bridge-A (Figure 5.4). In this scenario the diagnostic test applied at each voltage setting would result in the following response: (V_1, V_2, V_3) = (D, D, D), i.e., the defect is detected at all three voltage settings.

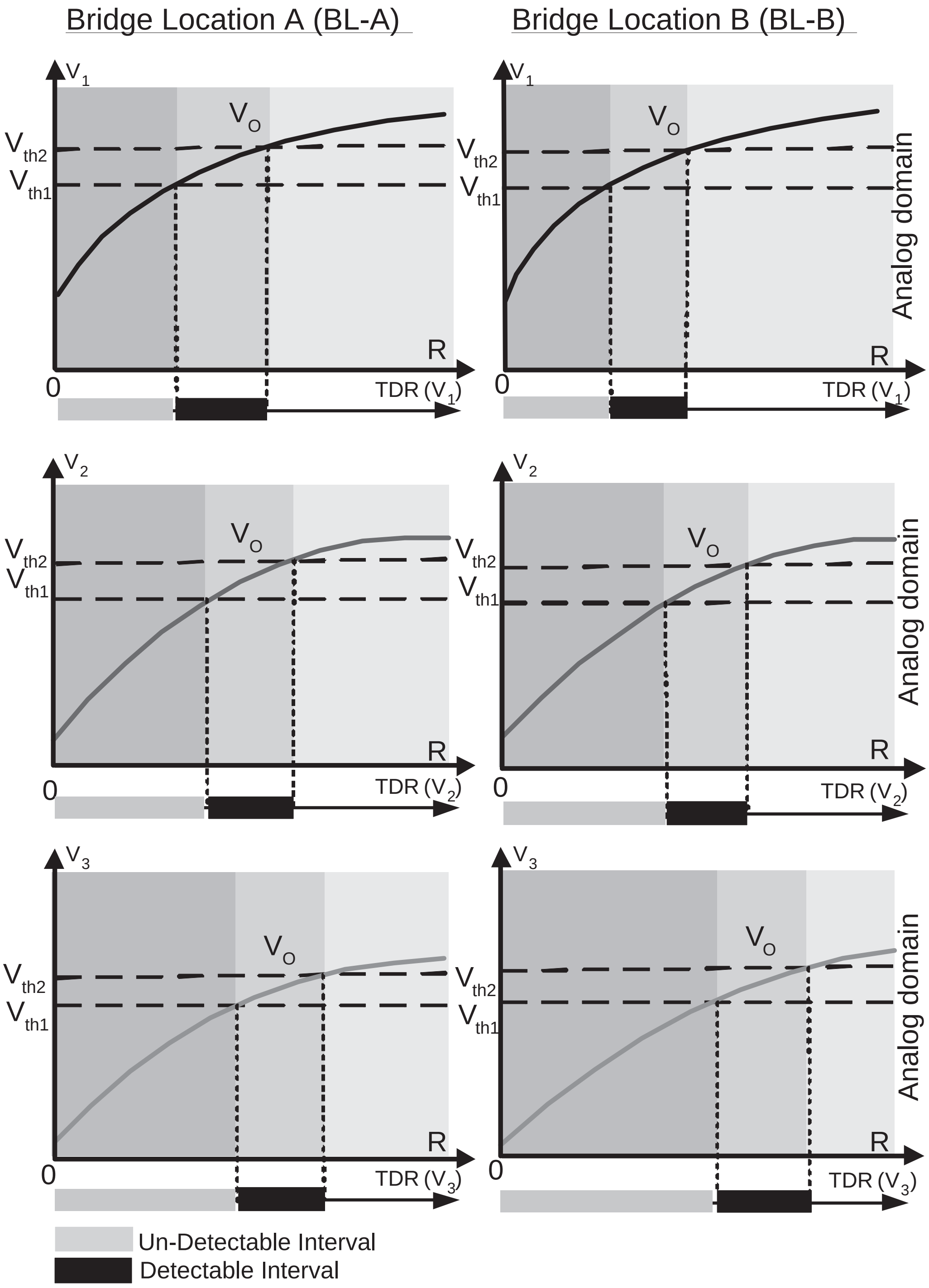


FIGURE 5.3: Analog behavior of resistive bridges at three different voltage settings

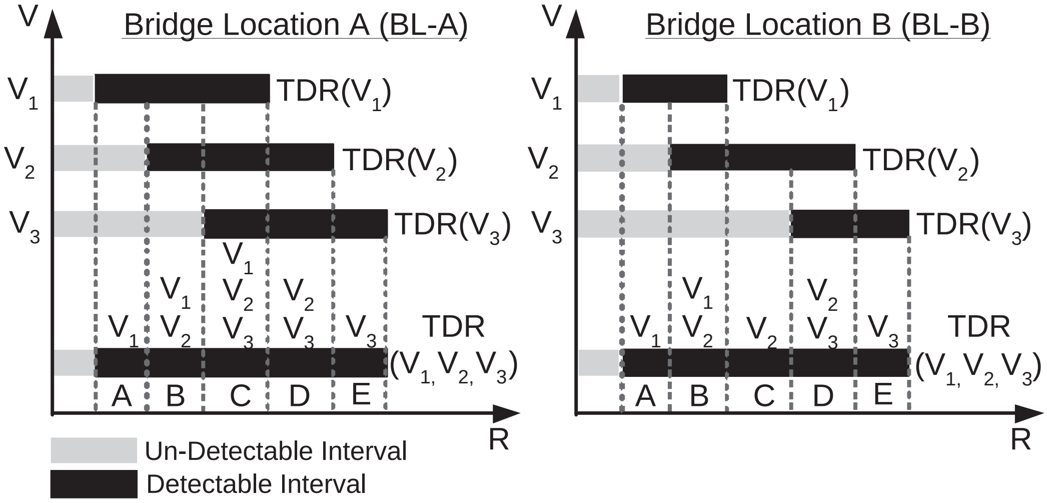


FIGURE 5.4: Two bridges detected by the same test pattern

TABLE 5.1: Resistance intervals exposed by single failing test at different voltage settings

	Bridge Location-A					Bridge Location-B				
	A	B	*C*	D	E	A	B	*C*	D	E
$\mathbf{V_1}$	D	D	*D*	ND	ND	D	D	*ND*	ND	ND
$\mathbf{V_2}$	ND	D	*D*	D	ND	ND	D	*D*	D	ND
$\mathbf{V_3}$	ND	ND	*D*	D	D	ND	ND	*ND*	D	D

We first carry out diagnosis at each voltage setting separately and then at all three voltage settings, using the information provided by Table 5.1 and the tester response. As mentioned earlier Table 5.1 shows the (D/ND) status of each interval of the two bridges, as detected by the only FP. The tester response at V_1 is “D”, which means that the diagnosis callout at V_1 is: bridge-A (intervals A, B, C) and bridge-B (intervals A, B). At V_2 the tester response is “D”, which means that the diagnosis callout at V_2 is: bridge-A (intervals B, C, D) and bridge-B (intervals B, C, D) and finally at V_3 the tester response is “D”, and the diagnosis callout is: bridge-A (intervals C, D, E) and bridge-B (intervals D, E). Next, we take into account the tester response at all three voltage settings, which is (D, D, D) and by combining the diagnosis callout at each voltage setting, we can identify the bridge and resistance interval that is common across all three voltage settings, i.e., bridge-A (interval C), which is indeed the actual inserted defect.

From this example, we can see that it is possible to improve the diagnosis callout by combining the information obtained from diagnosing the defect at three different voltage settings.

5.3.2 Passing Resistance Interval

This step further exploits the additional information, which is only available by diagnosing the design using multiple voltage settings. The diagnostic test applied at multiple voltage settings may detect a defect at one voltage setting but it may not detect it at another voltage setting. This concept is shown in Figure 5.2-B, a resistance range $R_{sh} \in [R_{2A}, R_{2B}]$ of Total Detectable Resistance (TDR) (Vdd_A & Vdd_B) can only be covered at Vdd_B. This means that a test pattern can detect this defect at Vdd_B only and will not be able to detect it at Vdd_A. Such test patterns that show a Detected "D" status at one voltage setting and Not-Detected "ND" status at other(s) are referred to as Partially Passing (PP) patterns.

The following example shows the effect of using PP patterns to improve diagnosis resolution. For this example we assume that interval *C* of bridge-B is causing malfunction and only one test pattern is a failing test pattern (FP). In this case, the tester response at three voltage settings (V$_1$, V$_2$, V$_3$) is (ND, D, ND). The diagnosis is carried out using the information available in Table 5.1 and the tester response. Table 5.2 shows the progressive reduction in the list of suspected bridges as a result of each diagnosis step. The left most column shows the voltage setting, the next column shows the Bridges (Resistance Intervals) detected by the FP at the particular voltage (as shown in Table 5.1) and the last column shows the D/ND status, using the Tester Response (TR). We first carry out diagnosis at V$_2$ as that has the detected status alone. The tester response at V$_2$ is "D", which means that the diagnosis callout at V$_2$ is: BL-A (intervals B, C, D) and BL-B (intervals B, C, D). Next, we take into account the resistance intervals for the two bridges that are detectable at other voltage settings, i.e., V$_1$ and V$_3$. At V$_1$, the detected bridges (resistance intervals) by the FP are: BL-A (intervals A, B, C) and BL-B (intervals A, B), but since the tester response is "ND", this means that all these intervals for the two bridges can not be causing malfunction in the circuit, and therefore the common intervals (for each bridge) can be removed from the suspected bridge list. As shown in Table 5.2, after removing the common intervals, the remaining intervals for the two bridges are: BL-A (interval D) and BL-B (interval C, D). Next, we carry out the same procedure at V$_3$ and remove the common interval for the two bridges from the suspected bridge list, i.e, interval D for both BL-A and BL-B. This gives BL-B (interval C) alone as the suspected candidate list, which in turn is the exact diagnosis. Furthermore it is an improvement over single-V$_{dd}$ diagnosis (at V$_2$: BL-A (intervals B, C, D) and BL-B (intervals B, C, D)).

TABLE 5.2: Improvement by removal of passing resistance intervals

V$_{\text{dd}}$	Bridges (Resistance Intervals)		TR
V$_2$	BL-A (B, C, D)	BL-B (B, C, D)	D
V$_1$	BL-A (A, B, C)	BL-B (A, B)	ND
Suspected Bridges: BL-A (D), BL-B (C, D)			
V$_3$	BL-A (C, D, E)	BL-B (D, E)	ND
Suspected Bridges: BL-A (∅), BL-B (C)			

The above example shows the usefulness of Partially Passing patterns in improving diagnosis, which are not available at single voltage setting.

5.4 Multi-V$_{\text{dd}}$ Diagnosis Algorithm

This section presents the diagnosis algorithm that carries out diagnosis at single/multiple voltage settings using a simple pass/fail (D/ND) test. The algorithm uses dictionary and tester response; the flow is shown in Figure 5.5. The dictionary holds the resistance range of each bridge, which is detected by a Test Pattern (TP) when it is applied at a certain voltage setting V_i, where V_i could be V_1, V_2, or V_3. From now onwards, we will refer to it as (TP, V_i) pair. Every bridge with its complete resistance range is fault-simulated separately by each one of the (TP, V_i) pair. The detected resistance interval(s) of each bridge is stored in the dictionary, against the (TP, V_i) pair that detects it. Fault simulation is performed using the procedure outlined in [Ingelsson, 2009]. The tool flow for generating dictionaries is shown by Figure 5.8 and further explained in section 5.5. The diagnosis algorithm also uses emulated tester response using the fault simulator presented in [Ingelsson, 2009]. It provides all the Failing Patterns (FP), corresponding voltage setting V_i on which the defect is detected, and the observed primary output response of the design, i.e., all (FP, V_i, PO) tuple(s). This diagnosis algorithm consists of three types of intersection and primary output matching scheme, which are explained next:

5.4.1 Bridge Intersection (BI)

The diagnosis algorithm starts by reading all the (FP, V_i) pairs generated by the tester. Using the dictionary and each (FP, V_i) pair, it retrieves all the bridges along with their resistance intervals that are detected by the particular (FP, V_i) pair. It then identifies the common bridges that each

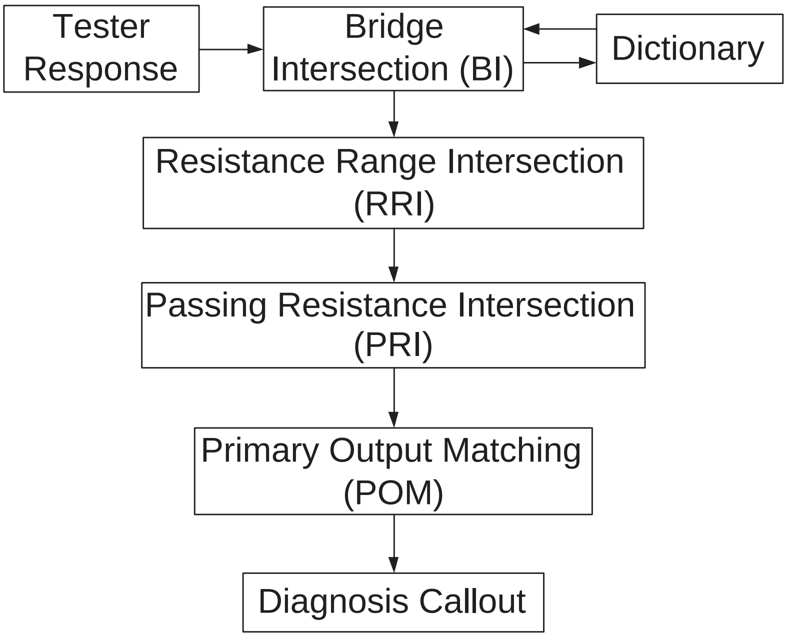

Tester Response
Bridge Intersection (BI)
Dictionary
Resistance Range Intersection (RRI)
Passing Resistance Intersection (PRI)
Primary Output Matching (POM)
Diagnosis Callout

FIGURE 5.5: Flow of the proposed Multi-V_{dd} Diagnosis Algorithm

one of the (FP, V_i) pair detects. The list of common bridges across all the (FP, V_i) pairs gives the "first suspected candidates list".

5.4.2 Resistance Range Intersection (RRI)

The size of "first suspected candidates list" can be further reduced by using the fact that resistive bridge defects manifests themselves at a single resistance value. This means that a defect should show a common resistance interval across all the failing patterns, otherwise it can be removed from the suspected candidate list. This idea is illustrated by Table 5.3. The table lists the two bridges (BL-A and BL-B) and their respective resistance intervals, detected by each one of the (FP, V_i) pair. It can be seen that only resistance interval "C" of BL-A is common to all three (FP, V_i) pairs and there is no resistance interval of BL-B that is common across all FPs. This means BL-B can be removed from the suspected candidates list. RRI removes the bridges with inconsistent resistance intervals and returns the "second suspected candidates list".

5.4.3 Passing Resistance Intersection (PRI)

The purpose of Passing Resistance Intersection (PRI) is to remove the resistance interval(s) (for each bridge in the "second suspected candidate list"), which is not causing malfunction in the

Table 5.3: Diagnosis improvement by resistance range intersection

	Suspected Bridges (Resistance Intervals)	
(FP, V_1)	BL-A (A, B, C)	BL-B (C, D, E)
(FP, V_2)	BL-A (B, C, D)	BL-B (A, B, F)
(FP, V_3)	BL-A (A, C, D)	BL-B (A, E)
RRI	BL-A (C)	

circuit, thereby narrowing the suspected list of bridges. This is achieved by using the PP Patterns (test patterns that pass at one voltage setting but fail at another), dictionaries and the “second suspected candidate list”. Dictionaries hold the detectable resistance interval(s) of all bridge locations, detected by a test pattern when applied at a certain voltage setting. Test patterns that pass at a certain voltage setting are referred as (PP, V_i) pair. This means that (PP, V_i) pair holds the resistance interval(s) (for respective bridges) that is not causing malfunction in the circuit and can be safely removed from the resistance range of suspected bridges. Bridges with empty list of resistance intervals can be removed from the suspected candidates, thereby improving diagnosis accuracy. The algorithm for this diagnosis step is outlined in Figure 5.6.

The algorithm starts by first finding the passing voltage(s) for all the (FP, V_i) pairs and storing the corresponding (PP, V_i) pairs. It then fetches the list of all detected bridges with their corresponding resistance interval(s), for all the (PP, V_i) pairs, from the dictionary. These two steps are shown in lines 1-4. In line 5, the algorithm compiles the “PP Bridge List” by combining the resistance interval(s) of each bridge, detected by (PP, V_i) pair, i.e., “PP Bridge List” holds the non-faulty resistance interval(s) of each bridge.

The algorithm goes over each bridge in Suspected Bridge list (one-by-one) and identifies the overlapping resistance interval(s) of the same bridge in PP Bridge list. This overlapping resistance interval(s), marked as ORI, is removed from the list of resistance interval(s) of the particular bridge in Suspected Bridge list. This process is repeated for all the bridges in Suspected Bridge list and is shown by lines 6-14. Next, it removes bridges with empty list of resistance intervals, from Suspected Bridge list. This step is shown by lines 15-17. Finally, the algorithm returns the “Final Bridge List”, which holds all the bridges with their resistance intervals.

5.4.4 Primary Output Matching (POM)

Primary Output Matching (POM) improves diagnosis accuracy further by removing resistance intervals (for each suspected bridge), which produce a different output response than produced

```
Input: List of (FP, $V_i$) pairs, Suspected Bridge List
 1: Using the (FP, $V_i$) pair, compile the list of (PP, $V_i$) pair.
 2: for all (PP, $V_i$) pairs do
 3:   Fetch the detected resistance interval for each bridge from the dictionary.
 4: end for
 5: PP Bridge List = Compute the overall passing resistance interval(s) for each bridge in all of
    (PP, $V_i$) pairs.
 6: for all $BL_i \in$ Suspected Bridges do
 7:   $RI_i =$ Resistance Interval(s) of $BL_i$
 8:   for all $BL_j \in$ PP Bridge List do
 9:     if $BL_i = BL_j$ then
10:       $RI_j =$ Resistance Interval(s) of $BL_j$
11:       $ORI = RI_i \bigcap RI_j$
12:       $RI_i = RI_i - ORI$
13:     end if
14:   end for
15:   if $RI_i = \emptyset$ then
16:     Remove $BL_i$ from Suspected Bridge List
17:   end if
18: end for
19: return Suspected Bridge List
```

FIGURE 5.6: Passing Resistance Intersection

by the defect. The improvements achieved by this step are demonstrated by experimental results, as discussed in section 5.5. As mentioned earlier, the emulated tester response stores the primary output values for each failing pattern in the form of (FP, V_i, PO) tuple. POM is accomplished by applying failing pattern(s) in presence of each resistance interval (of every bridge) and comparing the observed output response with the one recorded by the tester for the particular (FP, V_i, PO) tuple. The resistance intervals, which deviate from the expected output response (stored in the tuple) are removed from the resistance intervals of the suspected bridge. In this way suspected resistance intervals are reduced (from respective bridges); finally bridges without any suspected resistance interval are completely removed from the suspected bridge list. The procedure is outlined in Figure 5.7.

The algorithm starts by fault simulating (using the procedure in [Ingelsson, 2009]) each resistance interval of the suspected bridge list using the (FP, V_i, PO) tuple and compares the output response of the DUT (marked by OR on line 5) with PO member of the tuple. It removes resistance interval from suspected bridge in case of a mismatch and moves to the next resistance

```
Input: List of (FP, $V_i$, PO) tuple, Suspected Bridge List
Output: Final Bridge List
 1: for all $BL_i \in$ Suspected Bridges do
 2:     for all $RI_k \in$ Resistance Interval of $BL_i$ do
 3:         for all $FP_j \in$ (FP, $V_i$, PO) tuple do
 4:             fault simulate $RI_k$ using ($FP_j$, $V_i$)
 5:             OR $=$ Output of DUT in presence of $RI_k$
 6:             if OR $\neq$ PO of $FP_j$ then
 7:                 Remove $RI_k$ from $BL_i$
 8:                 Move to next $RI$ of $BL_i$ ($k$=$k$+1)
 9:                 break /* go to line 3 */
10:             end if
11:         end for
12:     end for
13:     if $BL_i = \emptyset$ then
14:         Remove $BL_i$ from Suspected Bridge List
15:     end if
16: end for
17: Final Bridge List $=$ Suspected Bridge List
18: return Final Bridge List
```

FIGURE 5.7: Primary Output Matching

interval, otherwise it applies next failing pattern, this is shown by lines 6-10. Finally the algorithm removes those bridges from the suspected bridge list which have no resistance interval, as shown by line 13-15. This process is repeated for all the suspected bridges.

It should be noted that proposed diagnostic flow outlined in Figure 5.5 applies POM as the last step. The suspected bridge list is greatly reduced by first three intersection procedures (BI, RRI, PRI) and POM is applied on reduced number of suspected bridges, which restricts the computation time of the algorithm, as fault simulation is applied only on the remaining resistance intervals of suspected bridges.

5.5 Experimental Results

Five experiments are conducted to analyze and validate the proposed Multi-V_{dd} diagnosis algorithm and to analyse the trade-offs between diagnosis cost and accuracy. These experiments use ISCAS'85 and '89 full scan circuits, details of which can be found in Appendix C. The benchmark circuits are synthesized using ST Microelectronics 0.12μm cell library. The tool flow to generate dictionaries is shown by Figure 5.8. For each design, non-feedback bridges are identified from the circuit layout. The "extractRC" tool from Cadence is used to get all the pairs of nets that are capacitively coupled. These pairs of nets are the most likely bridge locations. Feedback bridges are identified and removed. Table 5.4 shows different circuits used, along with total number of gates and extracted bridges for each circuit. The dictionaries are generated by fault-simulating 500 pseudo-random test patterns[2] at three different voltage settings (0.8V, 1.0V, 1.2V) against each bridge, as discussed in section 5.4. Same test patterns are applied at each voltage setting for fair comparison between diagnosis at different voltage settings. The tester is emulated using the fault simulator described in [Ingelsson, 2009]. A study presented in [Rodriguez-Montanes et al., 1992] on 14 wafers from different batches and different production lines concluded that 98.3% of resistive bridges are $\leq$ 5 kΩ, while considering upper bound of uncertainty. Therefore to mimic the real scenario, defects are injected by randomly selecting a resistance value between 0-5 kΩ for a randomly selected bridge. The tester applies all 500 TPs at different voltage settings and outputs the (FP, V_i, PO) tuples for the diagnosis algorithm. For each circuit, 500 such random defects are injected (one at a time). A set of parameters are defined as follows to categorise the diagnosis callout for each test case.

1. **Exact (EXT):** The test case for which the diagnosis procedure returns a single bridge location and that bridge matches with the injected random bridge.
2. **Contains (CNT):** The test case for which the diagnosis procedure returns more than one bridge location and one of them matches with the injected random bridge.
3. **Empty (EMT):** The test case for which the diagnosis procedure does not return any bridge location.

This setup is used to conduct five experiments. The first experiment analyses the voltage setting that achieves best level of diagnosis, second shows the possible improvement in diagnosis accuracy by carrying it out at multiple-voltage settings. Third experiment analyses the impact

[2] Please note that we used 1000 pseudo-random test patterns at each V_{dd} setting in the earlier version of this work presented at ETS'08, therefore diagnosis callout differ from results reported in ETS'08.

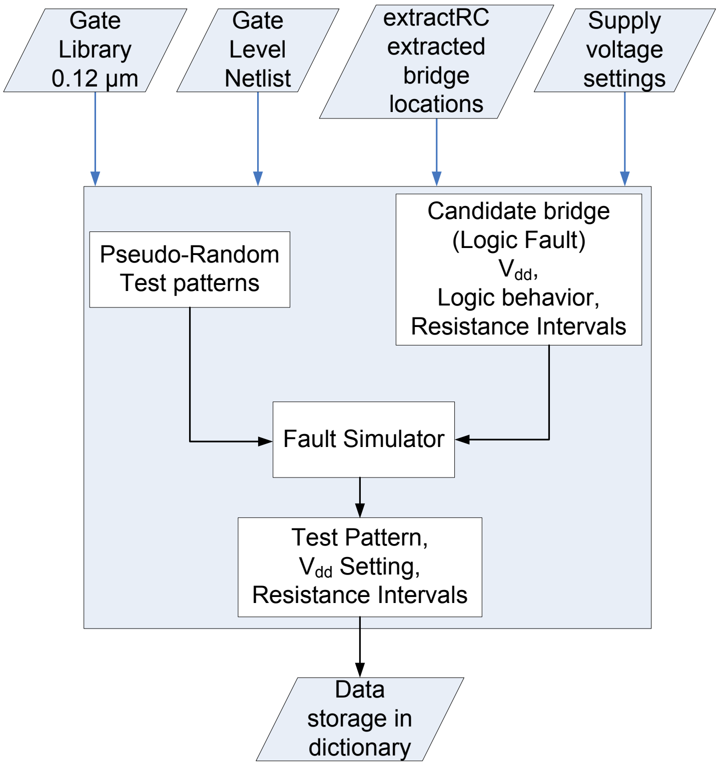

Gate Library 0.12 µm
Gate Level Netlist
extractRC extracted bridge locations
Supply voltage settings
Pseudo-Random Test patterns
Candidate bridge (Logic Fault) V_{dd}, Logic behavior, Resistance Intervals
Fault Simulator
Test Pattern, V_{dd} Setting, Resistance Intervals
Data storage in dictionary

FIGURE 5.8: Tool flow for Dictionary generation

of missing out diagnosis at one of the three voltage setting and shows the effect of conducting diagnosis on different V_{dd} pairs {(0.8V, 1.0V), (0.8V, 1.2V), (1.0V, 1.2V)}. This experiment is motivated towards saving tester time while recognizing the V_{dd} pair that achieves highest diagnosis accuracy. The fourth experiment is geared towards getting an insight into diagnosis of hard-shorts in the context of multi-V_{dd} designs, as they behave differently than bridges with higher resistance value. Last experiment shows that higher diagnosis accuracy can be achieved using larger (or high resolution ATPG generated) tests.

TABLE 5.4: Benchmarks

CKT.	# Gates	# Bridges
c432	93	47
c880	161	69
c499	187	85
c1908	205	98
c1355	226	80
s1488	281	435
s9234	434	223
c3540	439	363
s5378	578	305
c7552	731	578
s13207	1064	358
s15850	1578	943
s35932	3689	1170
s38584	5133	2937

5.5.1 Diagnosis Accuracy Using Single-V_{dd} Setting

The first experiment uses first two steps of the proposed diagnosis algorithm, i.e., Bridge Intersection and Resistance Range Intersection at each voltage setting separately. For every defect these two steps are carried out at each voltage setting independently and results are compiled to compare the diagnosis accuracy at each voltage setting.

Table 5.5 tabulates the outcome of the experiment. The first column shows the benchmark circuits, the next three main columns, marked with "@ V_{dd} $0.8V$", "@ V_{dd} $1.0V$" and "@ V_{dd} $1.2V$", show the number of test cases which fall into one of the three diagnostic categories (EXT, CNT, EMT) as a result of applying first two steps of the proposed diagnosis procedure at the particular voltage setting. It can be observed from Table 5.5 that diagnosis accuracy is highest at $0.8V$ with highest number of Exacts and least number of Empty callouts for all the circuits. It is only for s13207 that we notice higher number of Exacts at $1.2V$ in comparison to other voltage settings. It was further investigated by analyzing the detailed diagnosis callout, which shows that majority of test cases diagnosed exactly at 1.2V are included in the CNT group with 2-3 candidate bridges at other voltage setting. From this experiment we can observe that the lowest voltage setting achieves highest diagnosis accuracy for a large majority of circuits, which is similar to the findings reported recently using current based diagnosis [Arumi et al., 2007].

From Table 5.5 it can also be observed that the number of empty callouts are quite high for all the circuits. This is further probed by a small experiment using circuits with higher number of empty callouts in Table 5.5. In this experiment 500 random defects are inserted but unlike previous experiment, each defect is detectable at at-least one voltage setting and the outcome is tabulated in Table 5.6. In Table 5.6 it should be noted that the number of empty callouts are quite high at 1.0V and 1.2V in comparison to 0.8V. Empty callouts at 0.8V are very few and these defects are then detected at higher voltage settings for s9234, s5378 and s13207. This behavior can be understood from the study reported in [Rodriguez-Montanes et al., 2006], which shows that for some bridges connected by gates of equal drive strength, higher V_{dd} is more effective for fault detection. This experiment shows that logic faults have higher detectability at the lowest voltage setting (0.8V) as a defect does not show a faulty logic behavior at higher voltage settings, which is in line with previously reported research [Hao and McCluskey, 1993]. Secondly high empty callouts (in Table 5.5) is also due to using pseudo-random test patterns, which are not optimized for defect detection and are used for illustration purposes.

TABLE 5.5: Diagnosis callout at Single voltage setting

	@ V_{dd} 0.8V			@ V_{dd} 1.0V			@ V_{dd} 1.2V		
CKT.	EXT	CNT	EMT	EXT	CNT	EMT	EXT	CNT	EMT
c432	**350**	107	43	302	94	104	264	101	135
c880	**423**	41	36	355	47	98	297	45	158
c499	**330**	97	73	290	88	122	245	91	164
c1908	**263**	190	47	230	174	96	202	154	144
c1355	**372**	76	52	329	79	92	289	81	130
s1488	**228**	230	42	194	200	106	173	171	156
s9234	0	362	138	0	305	195	0	271	229
c3540	**339**	133	28	281	141	78	239	133	128
s5378	**102**	320	78	85	286	129	75	246	179
c7552	**369**	99	32	298	100	102	253	91	156
s13207	79	266	155	66	241	193	**129**	141	230
s15850	0	468	32	0	406	94	0	355	145
s35932	**276**	150	74	250	141	109	211	120	169
s38584	**180**	265	55	159	233	108	133	206	161

TABLE 5.6: Analysis for Empty Callouts

	@ V_{dd} 0.8V			@ V_{dd} 1.0V			@ V_{dd} 1.2V		
CKT.	EXT	CNT	EMT	EXT	CNT	EMT	EXT	CNT	EMT
c499	385	115	0	336	107	57	283	109	108
c1908	291	209	0	252	192	56	220	170	110
c1355	415	85	0	367	189	44	321	90	89
s9234	0	499	1	0	415	85	0	365	135
s5378	113	384	3	93	342	65	84	291	125
s13207	117	380	3	96	341	63	185	203	112

5.5.2 Diagnosis Accuracy Using Multi-V$_{dd}$ Settings

The second experiment uses the complete diagnosis algorithm across all the voltage settings. In this case, the tester response holds the failing patterns over all three voltage settings and corresponding primary output response. Table 5.7 shows the outcome of this experiment. The 2^{nd} main column marked with "RRI", shows the effect of "Resistance Range Intersection" by taking into account all bridges (with their resistance ranges) detected at all voltage settings. The 3^{rd} main column marked with "PRI", shows the effect of applying "Passing Resistance Intersection" by using the partially passing patterns. The last main column marked with "POM", shows the effect of applying "Primary Output Matching" by fault simulating the suspected bridges using (FP, V_i, PO) tuples. From Table 5.7 it can be observed that in all cases POM achieves best diagnosis accuracy with highest number of Exact callouts for all the circuits. It should also be noted from Tables 5.5 and 5.7 that "RRI" marginally improves over diagnosis at 0.8V. For majority of circuits, the number of Exact callouts at 0.8V have improved by less than 10. It is in case of c1908, s1488 and especially s13207 that it achieves significant improvement over Exact callouts at 0.8V.

The relative increase (Incr) in the number of Exact callouts by PRI and POM over other schemes are shown in 2^{nd} and 3^{rd} main columns of Table 5.8 by comparing the number of Exact callouts in each case. In 2^{nd} main column of Table 5.8, we list the relative increase in diagnosis accuracy of PRI over: A) "0.8 V" (2^{nd} column of Table 5.5) and B) "RRI" (2^{nd} column of Table 5.7). It should be noted that "PRI" achieves substantial improvement in diagnosis accuracy for all the circuits, showing up to 32.8% improvement over diagnosis callout at "0.8V" and "RRI". This clearly demonstrates the useful contribution of test patterns that pass at one voltage setting but fail at another (Partially Passing Patterns) in improving the overall diagnosis accuracy. Next, in 3^{rd} main column of Table 5.8 we list the relative increase in diagnosis accuracy of POM over: A) "RRI" (2^{nd} main column of Table 5.7) and B) "PRI" (3^{rd} main column of Table 5.7). It can be observed that "POM" achieves highest overall diagnosis accuracy for all the circuits, showing upto 72% improvement over "RRI" and 39.2% improvement over "PRI". This points to the success of POM in reducing the callouts categorized as "CNT" by PRI scheme.

From this experiment, we can observe that the Partially Passing patterns, which are not available at single voltage diagnosis can significantly improve diagnosis accuracy. The time taken by the Multi-V$_{dd}$ diagnosis algorithm ranges from a second to few minutes, depending on the size of benchmark circuit.

TABLE 5.7: Diagnosis callout at Multiple voltage settings

	RRI			PRI			POM		
CKT.	EXT	CNT	EMT	EXT	CNT	EMT	EXT	CNT	EMT
c432	357	100	43	383	74	43	**419**	38	43
c880	424	40	36	437	27	36	**441**	23	36
c499	330	97	73	376	51	73	**410**	17	73
c1908	276	177	47	326	127	47	**385**	68	47
c1355	373	75	52	396	52	52	**423**	25	52
s1488	251	207	42	347	111	42	**389**	69	42
s9234	0	363	137	109	254	137	**275**	88	137
c3540	340	133	27	395	78	27	**427**	46	27
s5378	105	320	75	250	175	75	**355**	70	75
c7552	371	97	32	400	68	32	**428**	40	32
s13207	160	188	152	200	148	152	**224**	124	152
s15850	0	468	32	164	304	32	**360**	108	32
s35932	276	151	73	295	132	73	**351**	76	73
s38584	183	262	55	303	142	55	**383**	62	55

TABLE 5.8: Diagnosis Improvement by PRI and POM

	PRI %Incr. over		POM %Incr. over	
CKT.	0.8V	RRI	RRI	PRI
c432	6.6	5.2	12.4	7.2
c880	2.8	2.6	3.4	0.8
c499	9.2	9.2	16	6.8
c1908	12.6	10	21.8	11.8
c1355	4.8	4.6	10	5.4
s1488	23.8	19.2	27.6	8.4
s9234	21.8	21.8	55	33.2
c3540	11.2	11	17.4	6.4
s5378	29.6	29	50	21
c7552	6.2	5.8	11.4	5.6
s13207	24.2	8	12.8	4.8
s15850	**32.8**	**32.8**	**72**	**39.2**
s35932	3.8	3.8	15	11.2
s38584	24.6	24	40	16

5.5.3 Diagnosis Cost Minimization

Diagnosis cost is directly affected by the time individual IC spends on the tester while running diagnostic test. For this reason, it is desirable to reduce tester time to achieve low-cost diagnosis with least compromise on diagnosis accuracy. From previous experimental results we have seen that high diagnosis accuracy is achieved by carrying out diagnosis at multiple voltage settings. The aim of this experiment is to evaluate the trade-off between diagnosis cost and accuracy. This is accomplished by investigating the most useful $\mathrm{V_{dd}}$ settings or combination of $\mathrm{V_{dd}}$ settings, which may yield the desired outcome by omitting tests at a certain voltage setting, thereby reducing diagnosis cost.

The third experiment also uses the complete diagnosis algorithm across different voltage settings. In this case, we carry out diagnosis using 3 $\mathrm{V_{dd}}$ pairs, i.e., (0.8V, 1.0V), (0.8V, 1.2V) and (1.0V, 1.2V). The outcome of this experiment is shown in 2^{nd}, 3^{rd} and 4^{th} main columns of Table 5.9. From Table 5.9 it can be observed that the diagnosis callout at “0.8V and 1.0V” achieves the highest accuracy in comparison to the other two $\mathrm{V_{dd}}$ pairs, i.e., (0.8V, 1.2V) and (1.0V, 1.2V).

It can be observed that Multi-$\mathrm{V_{dd}}$ diagnosis scheme that uses all $\mathrm{V_{dd}}$ settings (shown in 4^{th} main column of Table 5.7) achieves slightly better diagnosis accuracy than diagnosis at “0.8V and 1.0V”. In terms of the number of exact callouts found by the two, the maximum difference is 12 for all the circuits. On the other hand, the maximum difference in number of exact callouts between diagnosis at all $\mathrm{V_{dd}}$ settings and at “0.8V and 1.2V” is 44 (in case of s15850). The maximum difference is even higher, i.e., 104 (in case of s1488) in comparison to the number of exact callouts at “1.0V and 1.2V”. This experiment shows that the tester time, which is a crucial parameter in the diagnosis cost can be reduced by 33% by carrying out diagnosis at “0.8V and 1.0V” only, while achieving very high (close to the overall best) diagnosis accuracy.

TABLE 5.9: Diagnosis at different Voltage pairs

	@ 0.8V and 1.0V			@ 0.8V and 1.2V			@ 1.0V and 1.2V		
CKT.	EXT	CNT	EMT	EXT	CNT	EMT	EXT	CNT	EMT
c432	**417**	40	43	416	41	43	357	39	104
c880	**440**	24	36	438	26	36	380	22	98
c499	**409**	18	73	408	19	73	364	14	122
c1908	**383**	70	47	376	77	47	325	79	96
c1355	**423**	25	52	419	29	52	378	30	92
s1488	**377**	81	42	375	83	42	285	109	106
s9234	268	95	137	**270**	93	137	218	87	195
c3540	**420**	53	27	416	56	28	352	70	78
s5378	**347**	78	75	344	80	76	279	92	129
c7552	**426**	42	32	**426**	42	32	343	55	102
s13207	215	132	153	**220**	127	153	190	118	192
s15850	**348**	120	32	316	152	32	323	83	94
s35932	**351**	76	73	**351**	76	73	317	74	109
s38584	**371**	74	55	366	79	55	312	80	108

5.5.4 Diagnosis of Hard-Shorts Using Multi-V_{dd} Settings

The purpose of this experiment is to get an insight into diagnosis of hard-shorts in the context of multi-V_{dd} designs and make appropriate recommendations for diagnosing such defects. The same experimental set up is used for diagnosis as for the first two experiments, but instead of inserting random resistance range for each bridge, resistance value is set to 0 Ohms for all the selected bridges. In this experiment the number of test cases are limited by the number of bridges extracted by the layout tool and listed in Table 5.4, however designs with more than 500 bridges are restricted by 500 test cases.

The first part of experiment uses first two steps of the proposed diagnosis algorithm, i.e., Bridge Intersection and Resistance Range Intersection at each voltage setting separately. For every defect these two steps are carried out at each voltage setting independently and results are compiled to compare the diagnosis accuracy at each voltage setting. Table 5.10 tabulates the outcome of this experiment in the same fashion as for Table 5.5. It should be noted that the number of exact callouts are in close proximity at all voltage settings for all the circuits other than s13207. Higher number of exact callouts are observed for s13027 at 1.2V than at other voltage settings, as noted in first experiment. The number of empty callouts are also in very close proximity for all the circuits, which suggests that injected defects are in CNT group for defects that are not uniquely identified (EXT group).

The second part of the experiment uses complete diagnosis algorithm across all voltage settings. In this case, the tester response holds the failing patterns over all three voltage settings and corresponding primary output response as used for the second experiment. Table 5.11 tabulates the outcome of this experiment using RRI, PRI and POM. In case of hard-shorts, while comparing the number of EXT callouts with single voltage diagnosis (Table 5.10), PRI shows up to 8.5% improvement (in case of s5378, while comparing with diagnosis at 1.2V) over single voltage diagnosis. However in case of resistive bridges this improvement is up to 32.8%, as shown in Table 5.8. Next we analyze the impact of POM in improving the diagnosis accuracy, as it can be seen that POM shows significant improvement over PRI and other techniques, but this improvement should not be entirely attributed to using more than one V_{dd} settings, as inserted defect may be identified by POM using one of the three V_{dd} settings.

In the light of this discussion it is fair to conclude that multiple voltage diagnosis shows higher improvement for resistive bridges than for hard-shorts.

TABLE 5.10: Diagnosis callout for Hard Shorts at Single voltage setting

	@ V_{dd} 0.8V			@ V_{dd} 1.0V			@ V_{dd} 1.2V		
CKT.	EXT	CNT	EMT	EXT	CNT	EMT	EXT	CNT	EMT
c432	44	3	0	44	3	0	44	3	0
c880	67	2	0	67	2	0	67	2	0
c499	72	13	0	72	13	0	72	13	0
c1908	63	34	1	67	30	1	67	30	1
c1355	71	9	0	72	8	0	72	8	0
s1488	306	127	2	323	110	2	332	101	2
s9234	0	188	35	0	190	33	0	190	33
c3540	286	76	1	287	75	1	287	75	1
s5378	96	199	10	97	199	9	99	197	9
c7552	464	29	7	465	28	7	465	28	7
s13207	63	214	81	63	215	80	140	138	80
s15850	0	491	9	0	491	9	0	491	9
s35932	383	115	2	383	115	2	383	115	2
s38584	381	115	4	383	113	4	382	114	4

TABLE 5.11: Diagnosis callout for Hard Shorts at Multiple voltage setting

	RRI			PRI			POM		
CKT.	EXT	CNT	EMT	EXT	CNT	EMT	EXT	CNT	EMT
c432	44	3	0	44	3	0	46	1	0
c880	67	2	0	67	2	0	67	2	0
c499	72	13	0	73	12	0	84	1	0
c1908	67	30	1	67	30	1	87	10	1
c1355	72	8	0	73	7	0	80	0	0
s1488	334	99	2	343	90	2	401	32	2
s9234	0	190	33	15	175	33	147	43	33
c3540	288	74	1	301	61	1	344	18	1
s5378	101	195	9	125	171	9	277	19	9
c7552	467	26	7	469	24	7	487	6	7
s13207	143	136	79	146	133	79	191	88	79
s15850	0	491	9	39	452	9	436	55	9
s35932	383	115	2	383	115	2	477	21	2
s38584	383	113	4	383	113	4	445	51	4

TABLE 5.12: Diagnosis callout for Resistive Bridges at Single voltage setting

	@ V_{dd} 0.8V			@ V_{dd} 1.0V			@ V_{dd} 1.2V		
CKT.	EXT	CNT	EMT	EXT	CNT	EMT	EXT	CNT	EMT
c432	**422**	78	0	355	100	45	316	97	87
c499	**406**	94	0	362	81	57	309	83	108
c1908	**381**	119	0	333	119	48	285	113	102
c1355	**430**	70	0	388	68	44	340	71	89
s9234	**198**	302	0	164	256	80	137	235	128
c3540	**383**	116	1	320	129	51	281	115	104
s5378	**259**	240	1	204	237	59	168	213	119
c7552	**411**	89	0	334	96	70	286	82	132
s13207	**228**	270	2	193	251	56	187	202	111

5.5.5 Impact of High Resolution Test on Diagnosis Accuracy

The aim of this experiment is to show the impact of test size on diagnosis accuracy. In this experiment, we have used 2000 pseudo-random test patterns (4 times that of test size used in previous experiments) at each V_{dd} setting. Dictionaries are generated using the same flow as shown in Figure 5.8 and explained in section 5.5. The defects are randomly injected and are detectable at least at one voltage setting, which is what an ATPG normally aims to target during test generation.

Table 5.12 shows the results of diagnosis callout at single voltage setting using first two steps of the diagnosis algorithm, i.e., Bridge Intersection and Resistance Range Intersection. As expected, for all the circuits shown in Table 5.12 the diagnosis accuracy has improved in comparison to results shown in Table 5.5, primarily due to increased test size.

In the second part of the experiment, complete diagnosis algorithm is used and results are shown in Table 5.13. As can be seen from Table 5.13 multiple voltage diagnosis shows significant increase in the number of Exact callouts in comparison to single voltage diagnosis (shown in Table 5.12). For PRI step, the %age increase in the number of Exact callouts is up to 22.4% (as for s5378) over single voltage ($0.8V$) diagnosis. These results are further improved by the POM step, which shows up to 38.2% increase (as for s5378) in the number of Exact callouts in comparison to single voltage diagnosis.

The key observation of this experiment is that better diagnosis can be achieved with a large (high resolution) ATPG test set. It should be noted that for single voltage diagnosis highest accuracy

TABLE 5.13: Diagnosis callout for Resistive Bridges at Multiple voltage settings

	RRI			PRI			POM		
CKT.	EXT	CNT	EMT	EXT	CNT	EMT	EXT	CNT	EMT
c432	428	72	0	445	55	0	**473**	27	0
c499	406	94	0	442	58	0	**480**	20	0
c1908	398	102	0	439	61	0	**465**	35	0
c1355	431	69	0	447	53	0	**471**	29	0
s9234	198	302	0	284	216	0	**375**	125	0
c3540	389	111	0	445	55	0	**474**	26	0
s5378	263	237	0	371	129	0	**450**	50	0
c7552	412	88	0	441	59	0	**467**	33	0
s13207	246	254	0	303	197	0	**355**	145	0

is achieved at the lowest ($0.8V$) voltage setting, which can be further improved by multiple voltage diagnosis. In [Ingelsson, 2009], it was shown that for 8 out of 12 multi-V$_{\text{dd}}$ designs, 100% bridge defect coverage can't be achieved at single voltage setting. The study shows that most amount of bridge defect resistance is covered by tests at lowest V$_{\text{dd}}$ setting ($0.8V$), however for 100% defect coverage it is essential to generate tests at higher V$_{\text{dd}}$ settings. The proposed multi-V$_{\text{dd}}$ diagnosis approach capitalizes on these findings and achieves overall high diagnosis accuracy by using multiple voltage settings.

5.6 Concluding Remarks

Low power ICs employing multiple-V$_{\text{dd}}$ designs are commonly used in hand-held devices. Developing effective diagnosis capabilities for such ICs is important for today's competitive mobile electronics. This work is based on cause-effect diagnosis scheme using a simple pass/fail dictionary to minimize memory storage, however conclusions drawn through the experiments reported in this work are expected to hold if a complete dictionary that uses complete faulty responses or if an effect-cause diagnosis procedure [Abramovici and Breuer, 1980, Abramovici et al., 1998] is used. This work has addressed for the first time diagnosis of multiple-V$_{\text{dd}}$ ICs and proposed a novel multi-V$_{\text{dd}}$ diagnosis algorithm to exploit the information from all voltage settings to achieve higher diagnosis accuracy. This work provides a proof-of-concept that Multi-V$_{\text{dd}}$ diagnosis can improve diagnosis accuracy over single-V$_{\text{dd}}$ diagnosis. In addition, it recommends a way to reduce diagnosis cost by carrying it out at ($0.8V$, $1.0V$) V$_{\text{dd}}$ settings and still achieve high diagnosis accuracy. The improved diagnosis accuracy justifies the usage of test patterns at more than a single-V_{dd} setting. Lastly, it shows experimental results to establish that Multi-V_{dd} diagnosis is more effective for resistive bridges than for hard-shorts.

Chapter 6

Conclusions and Future Work

The last decade has witnessed a tremendous increase in the demand of hand-held devices, for example, PDAs, laptops and smart-phones. For these devices, battery life is an important constraint limiting the support for additional features in such devices and researchers around the world have put their efforts together to extend it using low-power design techniques. Low-power design techniques aim to increase the battery life while supporting the demand for additional features in such devices. The miniaturization of CMOS process nodes has enabled higher integration of transistors per silicon die, but unfortunately the number of pins per die available for testing these devices do not increase at the same rate, leading to higher cost of manufacturing test. In the electronics industry, a widely accepted "rule of 10" indicates that the cost of detecting a bad component increases by 10 times at each level of assembly (from wafer to package, to board, and finally to system level) putting even more stress on manufacturing test techniques demanding continuous innovation to keep the cost of manufacturing test low. This has led to the development of more sophisticated DFT techniques that aim to reduce the rising cost of manufacturing test. The contributions presented in this thesis provide new and cost-effective *test and diagnosis* solutions for designs employing multi-V_{dd} settings, which are summarized in the next section followed by proposed future work.

6.1 Thesis Contributions

Multi-V_{dd} is an effective low power design technique commonly employed in hand-held devices that varies the supply voltage and operating frequency of a design according to the workload.

Manufacturing test of such devices poses new challenges because certain physical defects become active at a specific supply voltage settings and stay dormant otherwise. Resistive bridge defect is a dominant defect type in deep submicron designs that exhibits V_{dd}-dependent detectability. The first objective of this thesis[1] is met by analysing the V_{dd}-dependent detectability of resistive bridge defect using three operating voltages. It is found that the lowest operating voltage achieves the highest detectability (fault coverage), but to achieve 100% fault coverage, majority of circuits (16 out of 22) require test at more than one voltage setting. This means that the ATE (Automatic Test Equipment) will have to switch between different voltage settings to apply the test, incurring test cost due to switching overhead and degradation of test compaction quality.

The V_{dd}-dependent detectability of resistive bridges presents challenges for the existing test and diagnosis techniques for the following two reasons: 1) Existing DFT techniques use repetitive tests at several V_{dd} settings to achieve 100% fault coverage. Repetitive tests at several V_{dd} settings are undesirable as it increases the manufacturing test cost; 2) All existing diagnosis techniques use a single V_{dd} setting for diagnosing resistive bridges that exhibit V_{dd}-dependent detectability, which may reduce the diagnosis accuracy affecting subsequent design cycle and yield. This thesis presents the first detailed investigation on reducing test cost and improving diagnosis accuracy, while considering multi-V_{dd} designs in the presence of resistive bridges.

The test cost is reduced by exploiting an observation noted by earlier publications that a high percentage (generally 80% or above) of resistive bridge defects are detectable at the lowest operating V_{dd} setting. This thesis proposes two new and cost-effective DFT techniques to further increase the percentage of detectable logic faults at the lowest V_{dd} setting, thereby reducing test cost by avoiding the need to apply repetitive test at several V_{dd} settings. The two techniques, *test point insertion* and *gate sizing*, are proposed to address the 2nd objective of the thesis. These two techniques were discussed in Chapter 3 and Chapter 4 respectively.

The test point insertion technique was discussed in Chapter 3, which is the first technique to reduce the number of test V_{dd} settings without affecting the fault coverage of the original test. TPI capitalizes on the observation that each resistive bridge defect consists of a large number of logic faults, including both detectable and non-detectable logic faults. It targets resistive bridges requiring test at higher V_{dd} settings, and converts specific un-detectable logic faults at the lowest V_{dd} setting into detectable logic faults by the help of test points. Test points provide additional controllability and observability at the fault site thereby reducing the number of test V_{dd} settings by improving the testability of the design. The proposed TPI technique performs a detailed analysis using signal probabilities of nets and this analysis is used to select the minimum

[1] All objectives are listed on page # 51

number of logic faults that are likely to cover NRINEV intervals at the lowest V_{dd} setting using a minimum number of test points. In this way, the proposed TPI technique targets all NRINEV intervals and the number of test V_{dd} settings is reduced. Once the number of test V_{dd} settings is reduced, the proposed technique attempts to reduce the number of test points. This is achieved by identifying common nets in the design to provide the same controllability and observability to all fault sites requiring test points, thus reducing the number of test points, i.e., control points and observation points are minimized without affecting the fault coverage.

Experiments are conducted using ISCAS-85, ISCAS-89, and ITC-99 benchmarks, and in total 23 designs. All experiments are conducted using three V_{dd} settings, and 21 out of 23 designs require multi-V_{dd} test settings for 100% fault coverage. Experimental results show that the proposed TPI technique is able to achieve a single V_{dd} test for 10 designs that otherwise require two or more test V_{dd} settings. Similarly, the proposed TPI technique has reduced another set of 10 benchmark designs to two test V_{dd} settings that otherwise require three test V_{dd} settings. However, it couldn't reduce any test V_{dd} setting for 1 benchmark design. In terms of area overhead due to added test points, it has added ≤ 10 test points to the large majority of circuits. On average (considering all designs) the proposed TPI technique has added 6.7% additional gates. The added test points have shown negative effect on timing, and similarly the power consumption of the design has also increased in comparison to the original design. It is well-known in the test community that added test points improve fault coverage of the design at the cost of timing, area and power [Abramovici et al., 1998, Touba and McCluskey, 1996, Pomeranz and Reddy, 1998].

The proposed TPI technique has shown encouraging results in terms of reducing the number of test V_{dd} settings. However, it couldn't achieve single V_{dd} test for all designs, therefore Chapter 4 of this thesis has proposed another more effective technique to reduce the number of test V_{dd} settings. The second technique exploits the observation that the resistance interval covered by a detectable logic fault can be increased by changing the drive strength of gates driving the bridge and uses *Gate Sizing* to increase the drive strength of gates driving the bridge. The proposed gate sizing technique works on the principle of covering (i.e., completely overlapping) the resistance range greater than or equal to that of the NRINEV interval (for definition of NRINEV, see Sec. 3.1) at the lowest V_{dd} setting. This is achieved by increasing the resistance range of a detectable logic fault at the lowest V_{dd} setting such that it covers a higher resistance range than that of the NRINEV interval. The gates driving the bridge are re-sized and the number of test V_{dd} settings are reduced.

The proposed gate sizing technique has been implemented by two different algorithms, which are distinguished by the process of gate identification for replacement. In other words, the two algorithms solve the following question differently, i.e., how to identify a bridge location requiring test at a higher V_{dd} setting from a given list of bridge locations? Just like TPI, the first *Deterministic Algorithm* invokes a test generator to identify the set of bridges requiring test at higher V_{dd} settings, this algorithm returns an accurate set of bridges but it is computationally expensive as the test generator is a SAT-based ATPG with exponential worst case complexity. Therefore the second algorithm attempts to reduce the computation effort by using a probabilistic estimate that determines the likelihood of each bridge location to be detected at a higher test V_{dd} setting. Since bridges requiring higher V_{dd} settings for detection are only targeted for gate replacement. The second *Probabilistic Algorithm* uses a set of criteria based on probabilistic estimate, which is developed by analysing 23 different designs from ISCAS-85, ISCAS-89, and ITC-99 benchmarks, which is used to categorize the V_{dd} setting of each bridge location. The algorithm selectively uses a test generator where the probabilistic estimate is inconclusive. Probability based categorization reduces the need of invoking the test generator for each bridge location, thereby speeding up the gate sizing technique. The proposed gate sizing technique (implemented by the two algorithms) reduces the number of gates to be replaced by identifying and replacing common gates across all bridge locations that require high V_{dd} test. This is followed by test generation on the modified netlist that returns single V_{dd} test.

Experimental results show that the proposed gate sizing technique is able to achieve single V_{dd} test for all designs that otherwise require two or more test V_{dd} settings, without affecting the fault coverage of the original design. In terms of computation time, the Probabilistic Algorithm results in a significant speed up for a large majority of circuits and reduces runtime by up to 50%, when compared with the Deterministic Algorithm. The reduction in runtime is especially noticeable for larger designs. In terms of timing, area and power, the proposed gate sizing (GS) technique has improved timing for some designs, in comparison to the original design due to gate replacement by bigger and faster gates; the timing performance is better when compared with TPI for all designs. The proposed GS technique has a small area overhead in comparison to the original design, but it is lower than that of TPI for all designs. This is because, on average for all designs, it replaces only 3% of total gates. Similarly, the power consumption of GS modified designs is always lower than those modified by the TPI, however it is higher than the original design. This is because of the bigger load capacitance and higher leakage current of bigger gates replaced by the proposed GS technique.

The two proposed techniques (TPI in Chapter 3 and GS in Chapter 4) provide novel solutions to reduce the number of test V_{dd} settings without affecting the fault coverage of the original test, thereby reducing the test cost of designs operating at multiple-voltage settings, while targeting a dominant defect type, i.e., resistive bridge defect. The V_{dd}-dependent detectability of resistive bridges demands revisiting existing diagnosis techniques, as all existing diagnosis techniques employ a single V_{dd} setting for fault localization and therefore diagnosis of multi-V_{dd} designs imposes a challenge for defects exhibiting supply voltage dependent behavior as that may affect failure analysis with negative implications on subsequent design cycle.

Chapter 5 addresses the 3rd objective of this thesis by proposing a new and cost-effective technique for efficient diagnosis of multi-V_{dd} designs in the presence of resistive bridge defects. Using a cause-effect diagnosis technique with pass/fail dictionaries, the proposed diagnosis technique capitalizes on resistance range detection of all suspected bridges at each of the operating V_{dd} settings to narrow down the list of suspected candidates thereby improving the diagnosis accuracy. The cost-effectiveness of the proposed technique is achieved by identifying the most useful V_{dd} settings that achieve the same accuracy, while reducing test application time (TAT).

Experiments are conducted using ISCAS-85 and ISCAS-89 benchmark designs and experimental results show that for single voltage diagnosis, the lowest V_{dd} setting is the most effective one, and achieves highest diagnosis accuracy while considering single V_{dd} diagnosis. However, the accuracy can be improved by as much as 72%, by using the proposed multi-V_{dd} diagnosis technique that takes into account resistance range detection at all operating V_{dd} settings. Diagnosis at all operating V_{dd} settings may increase diagnosis cost by increasing test application time (TAT). Therefore experiments are conducted to identify the most useful V_{dd} settings (or combination of V_{dd} settings) that show very high diagnosis accuracy using the lowest two V_{dd} settings. The lowest two V_{dd} settings achieve very high (close to the overall best) diagnosis accuracy thereby reducing diagnosis cost of resistive bridge defects. This work also analyses hard-shorts (bridges with 0 Ω resistance) and experimental results show that the diagnosis accuracy has little variation across different voltage settings for this class of defects, and therefore any V_{dd} setting can be used just as well without affecting diagnosis accuracy.

The contributions presented in this thesis provide novel, relevant and cost-effective *test and diagnosis* solutions for designs employing multi-V_{dd} settings, while targeting resistive bridge defects that exhibit V_{dd}-dependent detectability. The conclusions drawn in this thesis are supported by extensive analysis using an advanced parametric bridge fault model, state-of-the-art EDA tools, widely-used benchmark designs and in-house software specifically developed to generate realistic data to meet the last objective of this thesis. It is hoped that the test and diagnosis techniques proposed in this thesis will make useful contributions towards the development of future EDA test tools.

6.2 Future Work: PV-Aware Test

CMOS fabrication process variation has been taken for granted for years and over many scaled technology nodes. Fabrication process variation is mainly due to sub-wavelength lithography, random dopant distribution and line edge roughness and affects the transistor threshold voltage V_T, oxide thickness, and its geometry (W, L) [Bernstein et al., 2006]. As silicon manufacturing processes scale to and beyond the 65nm node, process variation can no longer be ignored [Roy et al., 2006a, Mak. and Nassif, 2006]. It has been demonstrated that it is possible to control the impact of process variation on circuit performance and power through process-tolerant design and improved fabrication techniques [Bhunia et al., 2007b]. Variation-aware test, on the other hand, which is the focus of this research is a new area that is currently receiving considerable attention world-wide. There is a general consensus between various EDA vendors that existing manufacturing test methods appear to be affected by the fabrication process and operating power supply voltage variation and is therefore less effective for testing nanoscale ICs. There is a great deal of novel work to be undertaken in this new research area and this project aims to make a significant contribution to the topic of variation-aware semiconductor test. The test research group at the University of Southampton has been an early and key player in this new area and is well-qualified to undertake the proposed research (See [Ingelsson, 2009, Ingelsson et al., 2008, 2009]).

Once a design has been functionally verified, the next stage is to fabricate it. Conventionally, an IC is tested at the time of fabrication to find manufacturing imperfections. The testing of an IC is the process of exercising the circuit with test patterns (collections of logical 1s and 0s) and analyzing its response to determine whether it behaves correctly, with the aim of preventing the delivery of defective parts to customers. This project is focused on new manufacturing test methods that minimize test escapes (undetected logic and delay faults) introduced by variation in the fabrication process and operating power supply voltage. This is needed to increase the shipped-product quality for today's high-density nanometer ICs. Faults are used to model fabrication defects that can lead to physical failures. An error is the manifestation of a fault in the circuit. The purpose of manufacturing testing is to detect the faults by forcing them to be manifested as errors, e.g. incorrect outputs. A fault model is an abstraction of a physical defect which can be used to determine the effect (error) of the corresponding defect at the output of the circuit under test. The advantage of modeling physical defects as logical and delay faults is that the problem of fault analysis and test generation is simplified by abstracting the complex analogue behaviour to binary digital behavior [Roy et al., 2006b]. Commonly used fault models include stuck-at faults, delay faults, and bridging faults. With continuous scaling in process technology, we have ICs that offer high clock frequencies, low power and high density. However, advances in technology have also led to more manufacturing defect types with the most prominent being resistive shorts or bridging (resistive connection between two nodes either within logic gate, or between outputs of different logic gates) and resistive opens (resistor between two circuit nodes that should be connected). According to the ITRS 2005 [ITR, 2005], the frequency of resistive open and resistive short (RORS) defects increases with technology scaling. As an example, an industrial study estimated RORS account for as much as 58% of all defects [Montanes et al., 2002] found in an IC fabricated using 130nm. RORS defects alter the IC delay performance and change the logic function leading to IC failures, and therefore they are aggressively targeted by industry during manufacturing test. The considerable increase of RORS defects in nanometer ICs are due to the presence of many interconnection layers, growing number of connections between each layer, and denser interconnection lines and they are likely to become more prominent in next generation process technologies [ITR, 2005].

An overview of the state-of-the-art research in RORS defects testing including our preliminary research is discussed next, which forms the basis for the proposed research. Over the last couple of years fault models and test generation methods for RORS defects have been reported in the literature, recent examples include [Engelke et al., 2006b, Arumi et al., 2008b]. As a result of growing industry concerns, commercial Automatic Test Pattern Generation (APTG) methods and tools targeting RORS defects have become available recently, e.g. Synopsys TetraMax and Mentor Graphics FastScan. Whilst significant progress has been made on how to detect RORS defects effectively, recent academic and industrial research is showing that these defects are sensitive to variation in the operating power supply voltage, compromising their detectability. One example we have demonstrated [Khursheed et al., 2008] is that resistive shorts change their logical behaviour with varying V_{dd} and unless this V_{dd}-dependent behaviour is addressed during test generation, loss of defect coverage occurs leading to reduced IC yield and reliability. Another example [Kruseman and Heiligers, 2006] from industry considered resistive opens and shown how such defects are better detected at elevated V_{dd} (higher than nominal operating power supply V_{dd}). We have also demonstrated [Ingelsson et al., 2008, 2009] how resistive shorts are sensitive to fabrication process variation (VT, W/L, TOX) leading to new logical faults induced by such variation which are missed during test. There is further evidence that process variation introduces additional delay failures as reported in [Lu et al., 2005] and more recently by IBM [Iyengar et al., 2007] and TI [Devanathan et al., 2007a]. It should be noted that current commercial ATPG methods and tools do not take into consideration the variation in fabrication process and operating power supply voltage during test generation (i.e. variation-unaware) which is the main focus of the proposed research.

Recent research including that presented in this thesis, shows that logical and delay behaviour of RORS are sensitive to V_{dd} [Khursheed et al., 2008] and to process variation [Ingelsson et al., 2008, Lu et al., 2005, Iyengar et al., 2007, Devanathan et al., 2007a] and therefore new high quality manufacturing test methods targeting such defects are needed to minimize test escapes and to increase the shipped-product quality for nanometer ICs. The key objectives of the proposed research include the following:

1. Develop structural fault models for resistive open and resistive short (RORS) defects to predict accurately their logical and timing behaviour under process and power supply voltage (PV) variation;
2. Analyse using the developed fault models the effects of PV variation on the defects' behaviour to identify the variation-induced logic and delay faults that need to be targeted during test pattern generation to improve fault coverage and reduce test data volume;
3. Develop PV variation-aware test pattern generation methods (static and dynamic) for logic and delay tests leveraging the identified variation-induced faults to improve test quality (less test escapes) and reduced test cost (less test application time);
4. Investigate the impact of PV variations on diagnosis accuracy using the developed fault models to develop a robust diagnosis technique taking PV variations into account.

The produced manufacturing defect models and test pattern generation methods from the research programme will help to establish the scientific foundation required for the development of next generation process and voltage (PV) variation aware test methods and tools for nanoscale integrated circuits. This is highly novel research since to the best of our knowledge, at present there are no reported PV variation aware fault models for resistive open and resistive short defects. Such models will facilitate the development of more efficient test generation methods in terms of higher defect coverage (better test quality) with less volume of test data (lower test time) when compared with the state-of-the-art delay test methods reported in [Lu et al., 2005, Iyengar et al., 2007, Devanathan et al., 2007a], and the only reported basic logic test method reported by us [Ingelsson et al., 2008]. This research proposal is timely and responds to present and future industrial needs. This is because the availability of effective and low-cost test methods developed specifically to mitigate the impact of PV variation are of paramount importance if the test cost of nanometre ICs is to remain acceptable for the highly competitive microelectronics industry. The outcome from this research would be practical test solutions that are attractive to both industrial exploitation and further academic research.

Appendix A

Logic Threshold Calculation

The logic threshold of a gate input is used in experiments to determine the critical resistance of a logic fault, see Section 2.1.1 for detailed description of critical resistance calculation. The logic threshold of a gate input is defined as the input voltage at which the voltage at the output reaches half of the supply voltage, while keeping all other inputs of the gate at non-controlling value(s) [Segura et al., 1998]. Figure A.1 illustrates the logic threshold calculation for one of the two inputs of a nand gate.

In this thesis all experiments are conducted using 0.12μm ST Microelectronics gate library. Logic threshold is calculated for every input of all gates in the gate library, which is then stored in a database. For a given bridge location, this pre-compiled database is used in experiments to determine the critical resistance of a logic fault. Figure A.2 shows the tool flow to calculate the logic threshold of each gate in the library. As can be seen, the gate library provides the transistor level SPICE description of a gate with 'm' inputs, which is used to calculate the logic threshold of each gate input 'i', where i $\in$ m, at each of the three $\mathrm{V_{dd}}$ settings (i.e., 0.8V, 1.0V, 1.2V) respectively. For a given gate input 'i', the tool determines all 'k' non-controlling input

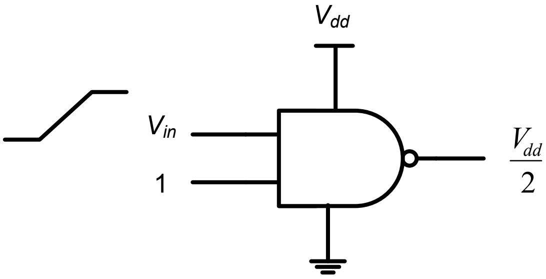


FIGURE A.1: Logic threshold calculation of a two input Nand gate.

TABLE A.1: Simulated logic threshold of gates using the tool flow shown in Figure A.2.

Gate	Input A			Input B		
	1.2V	1.0V	0.8V	1.2V	1.0V	0.8V
Two Input AND	0.52	0.44	0.36	0.54	0.46	0.37
Two Input NAND	0.52	0.45	0.37	0.55	0.46	0.38
Two Input OR	0.6	0.49	0.39	0.56	0.47	0.38
Two Input NOR	0.6	0.5	0.4	0.56	0.48	0.39
Inverter	0.55	0.46	0.38			

combinations of the gate and for each combination ‘j ∈ k’, it determines the logic threshold using Spectre simulation and records the input voltage at which the output of the gate reaches half of the supply voltage. In a similar way, logic threshold is calculated for all non-controlling input combinations ‘k’ of a gate input ‘i’. Logic threshold of the input ‘i’ is calculated by taking an average over all logic threshold values and the database is updated for input ‘i’ at a given V_{dd} setting. In this way, the tool generates the logic threshold of the gate input for all V_{dd} settings and this procedure is repeated for all inputs ‘i ∈ m’.

This tool flow is used to calculate logic threshold of 140 different gates with various number of inputs. As an illustration, Table A.1 shows the calculated logic threshold of five different gates at three V_{dd} settings using the tool flow shown in Figure A.2.

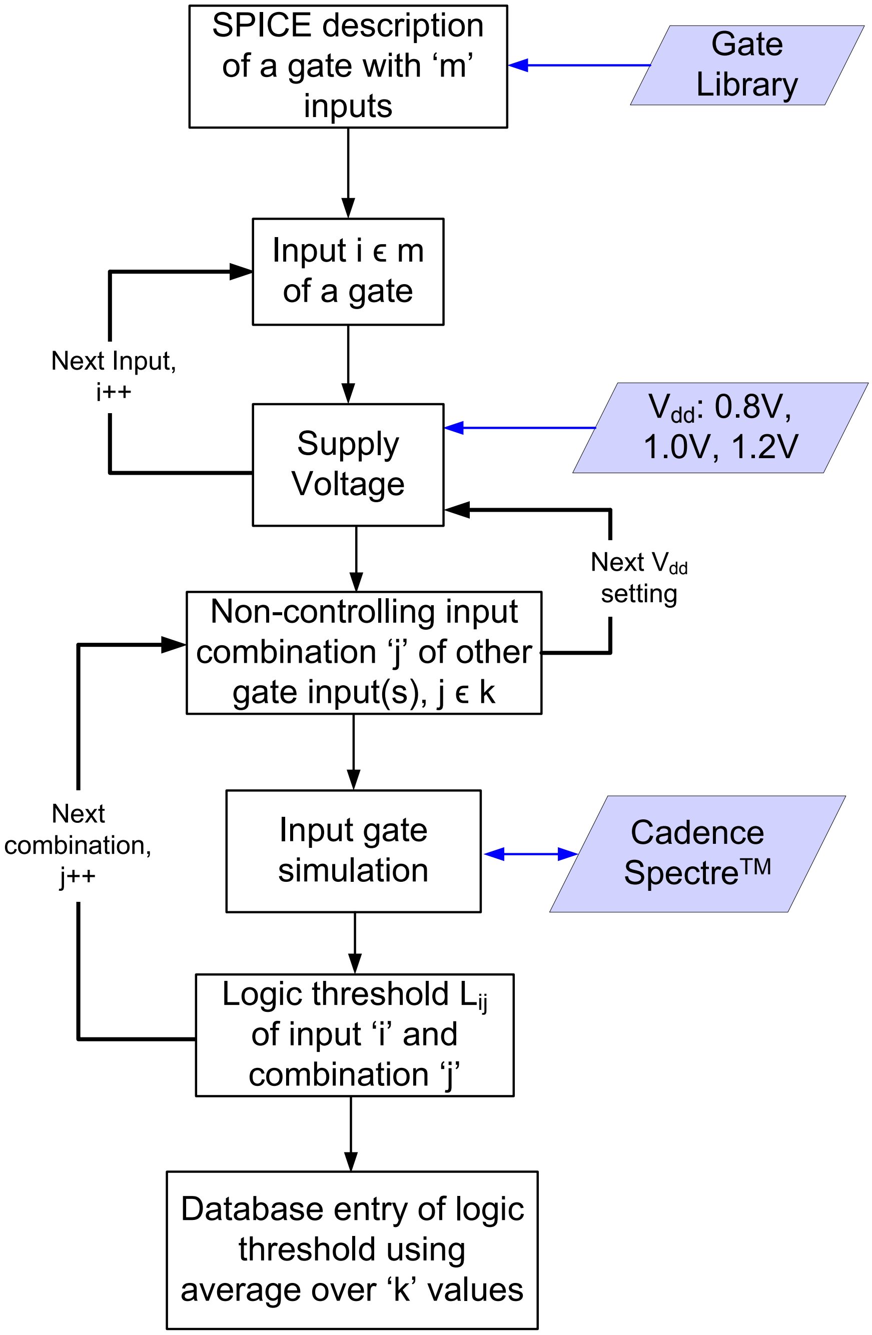

SPICE description of a gate with 'm' inputs
Gate Library
Input i ϵ m of a gate
Next Input, i++
Supply Voltage
V_{dd}: 0.8V, 1.0V, 1.2V
Next V_{dd} setting
Non-controlling input combination 'j' of other gate input(s), j ϵ k
Next combination, j++
Input gate simulation
Cadence Spectre™
Logic threshold L_{ij} of input 'i' and combination 'j'
Database entry of logic threshold using average over 'k' values

FIGURE A.2: Tool flow for calculating logic threshold of a gate.

Appendix B

Multi-Voltage Test Generation

The Multi-Voltage Test Generator (MVTG) is used for test generation in experiments reported in Chapter 3 and Chapter 4. MVTG was developed by Dr. U. Ingelsson as part of his PhD thesis [Ingelsson, 2009]. It was developed for detecting non-feedback bridge faults in designs operating at more than one V_{dd} setting. The MVTG is capable of generating a test for all detectable logic faults and guarantees 100% fault coverage for resistive bridge faults. The tool flow of the MVTG is shown in Figure B.1. It can be broadly categorized into two parts: 1) Logic fault generator and 2) Test pattern generator.

For a given bridge location and V_{dd} setting, the logic fault generator determines the boolean values at the fault site for a range of resistance value. The boolean values at the fault site includes inputs values to the gates feeding the bridge and the boolean values interpreted by the gates driven by the bridge. This information along with the V_{dd} setting (of operation) and covered resistance range is stored in a data structure called logic state configuration (LSC). The boolean value interpreted by the gates driven by the bridge is calculated by carrying out a DC sweep over a resistance range $\in [0,\infty)$ using Cadence Spectre^{TM} and the logic threshold values of the driven input (see Appendix A for details) is used to determine the cut-off point between faulty and fault-free gate behavior. Since Spectre simulation process for logic fault generation is time consuming and can seriously bog down the test generation process, therefore a database is created as a pre-processing step of test generation (Figure B.1). The database holds the voltage values on the nets affected by the bridge for a range of resistance values at a given V_{dd} setting and gate input(s) feeding the bridge. During test generation the logic fault generator accesses the database, using V_{dd} setting and gate input values as a database key, and gets the voltages on the bridged nets for a range of resistance values. It then uses the logic threshold values (stored in another database as discussed in Appendix A) of the driven gate inputs to determine the exact logic fault behavior at the fault-site and returns one or more LSCs to the test pattern generator.

The test generator uses the list of LSCs generated by the logic fault generator for a given bridge location, and generates a test pattern for each LSC that distinguishes the faulty behavior of a design from fault-free behavior. The MVTG uses ZChaff [zChaff, 2007] (SAT solver) for test pattern generation that propagate the fault effect to the primary output(s) and justifies the logic values of the fault site at the primary inputs of the design. The test pattern generator targets each LSC and returns a test pattern for each detectable logic fault and therefore results in test patterns with overlapping detectable resistance ranges and some of the test patterns are un-necessary that can be removed without affecting the fault coverage of the test. Such test patterns are removed from the test set during the final step shown as “Test pattern selection” in Figure B.1. Test size is reduced by using linear programming based minimum set cover technique [LP, 2009] that ensures that resistance range coverage remains the same by using minimal number of test patterns. The program finally terminates by generating a multi-voltage test set that ensures 100% bridge fault coverage.

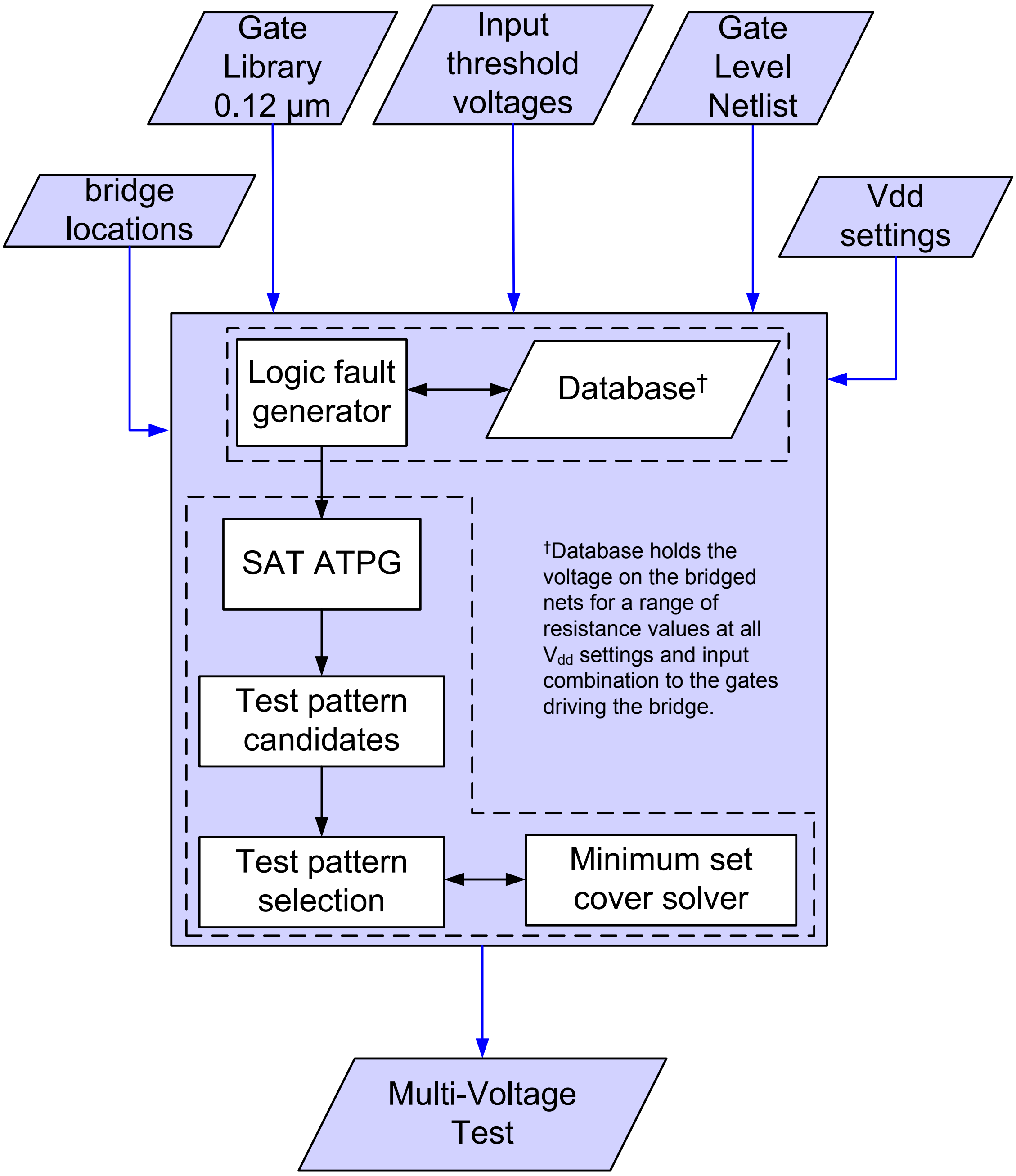


FIGURE B.1: Multi-Voltage Test Generation (MVTG) flow [Ingelsson, 2009].

Appendix C

ISCAS and ITC Benchmarks

Experiments reported in this thesis use ISCAS-85, ISCAS-89 and ITC-99 benchmark designs. Table C.1 and Table C.2 show all circuits used, along with the total number of gates, primary inputs, primary outputs and total number of flip-flops in each design. As can be seen, all ISCAS-85 benchmarks used in this work are combinational designs, while benchmarks in ISCAS-89 and ITC-99 are sequential designs. Sequential designs are converted to full-scan design and therefore all benchmarks used in this thesis are full-scan design. The benchmarks are synthesized using ST Microelectronics 0.12μm cell library, and the two tables show the post-synthesis gate count of each design. The default options of Synopsys Design Compiler are used for synthesis.

Table C.1: ISCAS-85 and ISCAS-89 benchmark designs

		No. of	**No. of**	**No. of**	**No. of**
S. No.	**Design**	**Gates**	**Primary Inputs**	**Primary Outputs**	**Flip Flops**
ISCAS-85 benchmarks [Hansen et al., 1999]					
1.	c432	93	36	7	0
2.	c880	161	60	26	0
3.	c499	187	41	32	0
4.	c1908	205	33	25	0
5.	c1355	226	41	32	0
6.	c2670	269	233	140	0
7.	c3540	439	50	22	0
8.	c7552	731	207	108	0
ISCAS-89 benchmarks [Brglez et al., 1989]					
9.	s344	62	9	11	15
10.	s386	63	7	7	6
11.	s382	74	3	6	21
12.	s1488	281	8	19	6
13.	s9234	434	19	22	228
14.	s5378	578	35	49	179
15.	s13207	1064	31	121	669
16.	s15850	1578	14	87	597
17.	s35932	3689	35	320	1728
18.	s38584	5133	12	278	1452

TABLE C.2: ITC-99 benchmark designs

		No. of	**No. of**	**No. of**	**No. of**
S. No.	**Design**	**Gates**	**Primary Inputs**	**Primary Outputs**	**Flip Flops**
ITC-99 benchmarks [Corno et al., 2000]					
19.	b01	26	2	2	5
20.	b02	15	1	1	4
21.	b03	63	4	4	30
22.	b04	208	8	11	66
23.	b05	315	1	36	34
24.	b06	33	2	6	9
25.	b07	170	1	8	49
26.	b08	86	9	4	21
27.	b09	75	1	1	28
28.	b10	88	11	6	17

Appendix D

SPICE Models

All experiments reported in this thesis utilize ST Microelectronics 0.12μm gate library. In the following, SPICE description of three gates (Inverter, NAND, NOR) is presented to provide library specific information.

```
//  Inverter
subckt ivhsx05 ( a z gnd vdd )

xmn0 ( z a gnd gnd ) enhsgp_bs3ju  w=0.260u l=0.130u ad=0.152p
+ as=0.145p pd=1.470u ps=1.340u

xmp0 ( z a vdd vdd ) ephsgp_bs3ju  w=0.470u l=0.130u ad=0.260p
+ as=0.290p pd=2.550u ps=2.780u

c1 ( vdd gnd ) capacitor c=0.119f
c2 ( gnd gnd ) capacitor c=0.809f
c3 ( z gnd ) capacitor c=0.075f
c4 ( a gnd ) capacitor c=0.263f
c12 ( vdd gnd ) capacitor c=0.105f
c14 ( vdd a ) capacitor c=0.004f
c23 ( gnd z ) capacitor c=0.125f
c24 ( gnd a ) capacitor c=0.114f
c34 ( z a ) capacitor c=0.118f

ends ivhsx05
```

```
// Two Input NAND Gate

subckt nd2hs ( a b z gnd vdd )

xmn0 ( net15 a gnd gnd ) enhsgp_bs3ju  w=0.640u l=0.130u ad=0.061p
+ as=0.305p pd=0.190u ps=2.870u

xmn1 ( z b net15 gnd ) enhsgp_bs3ju  w=0.640u l=0.130u ad=0.218p
+ as=0.061p pd=1.320u ps=0.190u

xmp0 ( z a vdd vdd ) ephsgp_bs3ju  w=0.770u l=0.130u ad=0.192p
+ as=0.471p pd=1.130u ps=4.130u

xmp1 ( z b vdd vdd ) ephsgp_bs3ju  w=0.770u l=0.130u ad=0.192p
+ as=0.471p pd=1.130u ps=4.130u

c1 ( vdd gnd ) capacitor c=0.145f
c2 ( gnd gnd ) capacitor c=0.902f
c4 ( z gnd ) capacitor c=0.109f
c5 ( a gnd ) capacitor c=0.135f
c6 ( b gnd ) capacitor c=0.136f
c12 ( vdd gnd ) capacitor c=0.139f
c14 ( vdd z ) capacitor c=0.022f
c15 ( vdd a ) capacitor c=0.002f
c16 ( vdd b ) capacitor c=0.002f
c24 ( gnd z ) capacitor c=0.188f
c25 ( gnd a ) capacitor c=0.070f
c26 ( gnd b ) capacitor c=0.066f
c45 ( z a ) capacitor c=0.106f
c46 ( z b ) capacitor c=0.142f
c56 ( a b ) capacitor c=0.133f

ends nd2hs
```

```
// Two Input NOR Gate


subckt nr2hs ( a b z gnd vdd )

xmn0 ( z a gnd gnd ) enhsgp_bs3ju  w=0.340u l=0.130u ad=0.116p
+ as=0.323p pd=0.995u ps=3.445u

xmn1 ( z b gnd gnd ) enhsgp_bs3ju  w=0.340u l=0.130u as=0.323p
+ ad=0.116p ps=3.445u pd=0.995u

xmp0 ( net028 a vdd vdd ) ephsgp_bs3ju  w=1.050u l=0.130u ad=0.131p
+ as=0.689p pd=0.250u ps=4.990u

xmp1 ( z b net028 vdd ) ephsgp_bs3ju  w=1.050u l=0.130u ad=0.423p
+ as=0.131p pd=3.180u ps=0.250u

c1 ( vdd gnd ) capacitor c=0.139f
c2 ( gnd gnd ) capacitor c=0.902f
c3 ( z gnd ) capacitor c=0.137f
c5 ( a gnd ) capacitor c=0.121f
c6 ( b gnd ) capacitor c=0.121f
c12 ( vdd gnd ) capacitor c=0.133f
c13 ( vdd z ) capacitor c=0.051f
c15 ( vdd a ) capacitor c=0.002f
c23 ( gnd z ) capacitor c=0.238f
c25 ( gnd a ) capacitor c=0.061f
c26 ( gnd b ) capacitor c=0.060f
c35 ( z a ) capacitor c=0.113f
c36 ( z b ) capacitor c=0.117f
c56 ( a b ) capacitor c=0.105f

ends nr2hs
```

Bibliography

International Technology Roadmap for Semiconductors, 2005, 2005. URL http://www/itrs.net/links/2005itrs/home2005.htm.

TransMeta Crusoe TM5800, October 2009. http://www.cpu-world.com/CPUs/TM5800/.

TPI: Experimental data, November 2007. http://users.ecs.soton.ac.uk/ssk06r/data/TPI.tgz.

M. Abramovici and M.A. Breuer. Fault diagnosis based on effect-cause analysis: An introduction. In *Proceedings ACM/IEEE Design Automation Conference (DAC)*, pages 69–76, June 1980.

M. Abramovici, M.A. Breuer, and A.D. Friedman. *Digital Systems Testing and Testable Designs*. IEEE Press, 1998.

V.D. Agrawal, C.R. Kime, and K.K. Saluja. A tutorial on built-in self-test. I. Principles. *IEEE Design & Test of Computers*, 10(1):73–82, Mar 1993a.

V.D. Agrawal, C.R. Kime, and K.K. Saluja. A tutorial on built-in self-test. II. Applications. *IEEE Design & Test of Computers*, 10(2):69–77, Jun 1993b.

V.D. Agrawal, Dong Hyun Baik, Yong Chang Kim, and K.K. Saluja. Exclusive test and its applications to fault diagnosis. In *International Conference on VLSI Design*, pages 143–148, Jan. 2003.

R.C. Aitken and P.C. Maxwell. Better Models or Better Algorithms? Techniques to Improve Fault Diagnosis. In *HP Journal*, February 1995.

B. Arslan and A. Orailoglu. Extracting precise diagnosis of bridging faults from stuck-at fault information. In *Proceedings IEEE Asian Test Symposium (ATS)*, pages 230–235, November 2003.

D. Arumi, R. Rodriguez-Montanes, J. Figueras, S. Eichenberger, C. Hora, B. Kruseman, and A.K. Lousberg, M.; Maihi. Diagnosis of bridging defects based on current signatures at low power supply voltages. In *Proceedings IEEE VLSI Test Symposium (VTS)*, pages 145–150, May 2007.

D. Arumi, R. Rodriguez-Montaes, J. Figueras, S. Eichenberger, C. Hora, and B. Kruseman. Full open defects in nanometric CMOS. *VLSI Test Symposium, 2008. VTS 2008. 26th IEEE*, pages 119–124, May 2008a.

D. Arumi, R. Rodriguez-Montanes, and J. Figueras. Experimental characterization of CMOS interconnect open defects. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 27(1):123–136, Jan. 2008b.

M. Baby and V. Sarathi. Advanced DFT implementation, 2008. http://www.synopsys.com/news/pubs/insight/2008/art2_dftimplem_v3s4.html.

T. Bartenstein, D. Heaberlin, L. Huisman, and D. Sliwinski. Diagnosing combinational logic designs using the single location at-a-time (slat) paradigm. In *International Test Conference, ITC*, pages 287–296, 2001.

R. Baumann. Soft errors in advanced computer systems. *Design & Test of Computers, IEEE*, 22 (3):258–266, May-June 2005.

J. Bedsole, R. Raina, A. Crouch, and M.S. Abadir. Very low cost testers: Opportunities and challenges. *Design & Test of Computers, IEEE*, 18(5):60–69, September 2001.

Luca Benini and Giovanni de Micheli. System-level power optimization: techniques and tools. *ACM Trans. Des. Autom. Electron. Syst.*, 5(2):115–192, 2000.

K. Bernstein, D. J. Frank, A. E. Gattiker, W. Haensch, B. L. Ji, S. R. Nassif, E. J. Nowak, D. J. Pearson, and N. J. Rohrer. High-performance CMOS variability in the 65-nm regime and beyond. *IBM Journal of Research and Development*, 50(4/5), 2006.

Naresh K. Bhatti and R.D. Blanton. Diagnostic test generation for arbitrary faults. In *IEEE International Test Conference, ITC*, pages 1–9, Oct. 2006.

S. Bhunia, H. Mahmoodi, D. Ghosh, S. Mukhopadhyay, and K. Roy. Low-power scan design using first-level supply gating. *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, 13(3):384–395, March 2005.

S. Bhunia, S. Mukhopadhyay, and K. Roy. Process variations and process-tolerant design. *VLSI Design, International Conference on*, pages 699–704, 2007a.

Swarup Bhunia, Saibal Mukhopadhyay, and Kaushik Roy. Process Variations and Process-Tolerant Design. pages 699–704, January 2007b.

F. Brglez, D. Bryan, and K. Kozminski. Combinational profiles of sequential benchmark circuits. In *IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1929–1934, May 1989.

M. Bushnell and V.D. Agrawal. *Essentials of Electronic Testing for Digital, Memory and Mixed-Signal Vlsi Circuits*. Kluwer Academic, Boston, 2000. ISBN 0792379918.

P. Camurati, D. Medina, P. Prinetto, and M. Sonza Reorda. A diagnostic test pattern generation algorithm. In *International Test Conference, ITC*, pages 52–58, Sep 1990.

K. Chakravadhanula, V. Chickermane, B. Keller, P. Gallagher, and S. Gregor. Test generation for state retention logic. *Asian Test Symposium, 2008. ATS '08. 17th*, pages 237–242, Nov. 2008.

S.J. Chandra and J.H. Patel. Experimental evaluation of testability measures for test generation (logic circuits). *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 8(1):93–97, Jan 1989.

V. Chandra and R. Aitken. Impact of voltage scaling on nanoscale sram reliability. *Proceedings of the conference on Design, automation and test in Europe (DATE)*, Apr. 2009.

J.T.-Y. Chang and E.J. McCluskey. Detecting delay flaws by very-low-voltage testing. In *Proceedings of the International Test Conference (ITC)*, pages 367–376, October 1996a.

J.T.-Y. Chang and E.J. McCluskey. Quantitative analysis of very-low-voltage testing. In *Proceedings of the VLSI Test Symposium (VTS)*, pages 332–337, May 1996b.

Jui-Ming Chang and M. Pedram. Energy minimization using multiple supply voltages. *IEEE Trans. VLSI Syst.*, 5(4):436–443, 1997. ISSN 1063-8210. doi: 10.1109/92.645070.

Mitrajit Chatterjee and D.K. Pradhan. A BIST pattern generator design for near-perfect fault coverage. *IEEE Transactions on Computers*, 52(12):1543–1558, Dec. 2003.

G. Chen, S. Reddy, I. Pomeranz, J. Rajski, P. Engelke, and B. Becker. An unified fault model and test generation procedure for interconnect open and bridges. In *Proceedings IEEE European Test Symposium (ETS)*, pages 22–27, May 2005.

Jian-Jia Chen, Chuan-Yue Yang, Tei-Wei Kuo, and Chi-Sheng Shih. Energy-efficient real-time task scheduling in multiprocessor dvs systems. In *ASP-DAC '07: Proceedings of the 2007*

Asia and South Pacific Design Automation Conference, pages 342–349, Washington, DC, USA, 2007. IEEE Computer Society.

Kwang-Ting Cheng. Gate-level test generation for sequential circuits. *ACM Trans. Des. Autom. Electron. Syst.*, 1(4):405–442, 1996.

V. Chickermane, P. Gallagher, J. Sage, P. Yuan, and K. Chakravadhanula. A power-aware test methodology for multi-supply multi-voltage designs. *Test Conference, 2008. ITC 2008. IEEE International*, pages 1–10, Oct. 2008.

U. Choudhury and A. Sangiovanni-Vincentelli. Automatic generation of analytical models for interconnect capacitances. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 14(4):470–480, Apr 1995.

A.D. Colle, S. Ramnath, M. Hirech, and S. Chebiyam. Power and design for test: A design automation perspective. *Journal of Low Power Electronics*, 1(1):73–84, 2005.

F. Corno, M.S. Reorda, and G. Squillero. RT-level ITC'99 benchmarks and first ATPG results. *IEEE Design & Test of Computers*, 17(3):44–53, Jul/Sep 2000.

A Delgado. Enhancement of defect diagnosis based on the analysis of CMOS DUT behaviour. *Ph.D Thesis,*, July 2008.

Tony Dennis. Uk officially nation of mobile phone users, August 2008. http://www.theinquirer.net/inquirer/news/1002655/uk-officially-nation-mobile-phone-users.

V.R. Devanathan, C.P. Ravikumar, and V. Kamakoti. Variation-tolerant, power-safe pattern generation. *Design & Test of Computers, IEEE*, 24(4):374–384, July-Aug. 2007a.

V.R. Devanathan, C.P. Ravikumar, R. Mehrotra, and V. Kamakoti. Pmscan: A power-managed scan for simultaneous reduction of dynamic and leakage power during scan test. *Test Conference, 2007. ITC 2007. IEEE International*, pages 1–9, Oct. 2007b.

A. Ejlali, B.M. Al-Hashimi, M.T. Schmitz, P. Rosinger, and S.G. Miremadi. Combined time and information redundancy for seu-tolerance in energy-efficient real-time systems. *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, 14(4):323–335, April 2006.

A. El-Maleh and S. Khursheed. Efficient test compaction for combinational circuits based on fault detection count-directed clustering. *Computers & Digital Techniques, IET*, 1(4):364–368, July 2007.

A.H. El-Maleh, S.S. Khursheed, and S.M. Sait. Efficient static compaction techniques for sequential circuits based on reverse-order restoration and test relaxation. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 25(11):2556–2564, Nov. 2006.

P. Engelke. Resistive bridging fault defect-oriented modeling and efficient testing. In *Ph.D Thesis, Freiburg University*, Jan 2009.

P. Engelke, I. Polian, M. Renovell, B. Seshadri, and B. Becker. The pros and cons of very-low-voltage testing: an analysis based on resistive bridging faults. In *Proceedings of the VLSI Test Symposium (VTS)*, pages 171–178, April 2004.

P. Engelke, I. Polian, M. Renovell, and B. Becker. Automatic test pattern generation for resistive bridging faults. *Journal of Electronic Testing: Theory and Applications*, 22:61–69, February 2006a.

P. Engelke, I. Polian, M. Renovell, and B. Becker. Simulating resistive bridging and stuck-at faults. *IEEE Transactions on Computer-Aided Design*, 25:2181–2192, October 2006b.

Piet Engelke, Bernd Becker, Michel Renovell, Juergen Schloeffel, Bettina Braitling, and Ilia Polian. Superb: Simulator utilizing parallel evaluation of resistive bridges. *ACM Trans. Des. Autom. Electron. Syst.*, 14(4):1–21, 2009.

F. Fallah and M. Pedram. Standby and active leakage current control and minimization in CMOS VLSI circuits. *IEICE Trans Electron*, E88-C(4):509–519, 2005.

F.J. Ferguson and J.P. Shen. Extraction and simulation of realistic CMOS faults using inductive fault analysis. In *Proceedings IEEE International Test Conference (ITC)*, pages 475–484, Sep 1988.

P. Franco and E.J. McCluskey. Delay testing of digital circuits by output waveform analysis. In *International Test Conference, ITC*, Oct 1991.

H. Fujiwara and S. Toida. The complexity of fault detection problems for combinational logic circuits. *IEEE Transactions on Computers*, 31(6):555–560, June 1982.

J. Galiay, Y. Crouzet, and M. Vergniault. Physical versus logical fault models mos lsi circuits: Impact on their testability. *Computers, IEEE Transactions on*, C-29(6):527–531, June 1980.

A. Gattiker. Using test data to improve IC quality and yield. In *Proc. IEEE/ACM International Conference on Computer-Aided Design ICCAD 2008*, pages 771–777, November 10–13, 2008. doi: 10.1109/ICCAD.2008.4681663.

M. Geuzebroek, J. Th. Van der Linden, and A. J. Van de Goor. Test point insertion for compact test sets. *Proceedings IEEE International Test Conference (ITC)*, pages 292–301, October 2000.

P. Girard. Survey of low-power testing of vlsi circuits. *Design & Test of Computers, IEEE*, 19 (3):80–90, May/Jun 2002.

Patrick Girard, Nicola Nicolici, and Xiaoqing Wen, editors. *Power-Aware Testing and Test Strategies for Low Power Devices*. Springer, November 2009. ISBN 9781441909275.

S.K. Goel, N. Devta-Prasanna, and R.P. Turakhia. Effective and efficient test pattern generation for small delay defect. In *27th IEEE VLSI Test Symposium, VTS*, pages 111–116, May 2009.

M. A. Gomaa and T. N. Vijaykumar. Opportunistic transient-fault detection. *IEEE Micro*, 26 (1):92–99, 2006.

R. Gomez, A. Giron, and V. Champac. Test of interconnection opens considering coupling signals. *Defect and Fault Tolerance in VLSI Systems, 2005. DFT 2005. 20th IEEE International Symposium on*, pages 247–255, Oct. 2005.

P.T. Gonciari, B.M. Al-Hashimi, and N. Nicolici. Variable-length input huffman coding for system-on-a-chip test. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 22(6):783–796, June 2003.

Y. Gong and S. Chakravarty. On adaptive diagnostic test generation. In *IEEE/ACM International Conference on Computer-Aided Design, ICCAD-95*, pages 181–184, Nov 1995.

T. Gruning, U. Mahlstedt, and H. Koopmeiners. DIATEST: a fast diagnostic test pattern generator for combinational circuits. In *IEEE International Conference on Computer-Aided Design, ICCAD*, pages 194–197, Nov 1991.

R.S. Guindi and F.N. Najm. Design techniques for gate-leakage reduction in CMOS circuits. *Quality Electronic Design, 2003. Proceedings. Fourth International Symposium on*, pages 61–65, March 2003.

M. Hamada, M. Takahashi, H. Arakida, A. Chiba, T. Terazawa, T. Ishikawa, M. Kanazawa, M. Igarashi, K. Usami, and T. Kuroda. A top-down low power design technique using clustered voltage scaling with variable supply-voltage scheme. *Custom Integrated Circuits Conference, 1998. Proceedings of the IEEE 1998*, pages 495–498, May 1998.

M.C. Hansen, H. Yalcin, and J.P. Hayes. Unveiling the ISCAS-85 benchmarks: A case study in reverse engineering. *IEEE Design & Test of Computers*, 16(3):72–80, 1999.

H. Hao and E.J. McCluskey. Very-low-voltage testing for weak CMOS logic ICs. In *Proceedings of the International Test Conference (ITC)*, pages 275–284, October 1993.

C.F. Hawkins, J.M. Soden, A.W. Righter, and F.J. Ferguson. Defect classes-an overdue paradigm for CMOS IC testing. In *Proceedings IEEE International Test Conference (ITC)*, pages 413–425, 2-6 1994.

J.P. Hayes. On modifying logic networks to improve their diagnosability. *IEEE Transactions on Computers*, C-23(1):56–62, Jan. 1974.

J.P. Hayes and A.D. Friedman. Test point placement to simplify fault detection. *IEEE Transactions on Computers*, C-23(7):727–735, July 1974.

C.L. Henderson and J.M. Soden. Signature analysis for ic diagnosis and failure analysis. In *International Test Conference, ITC*, pages 310–318, Nov 1997.

C.L. Henderson, J.M. Soden, and C.F. Hawkins. The behavior and testing implications of CMOS IC logic gate open circuits. *Test Conference, 1991, Proceedings., International*, pages 302–310, Oct 1991.

S. Holst and H. Wunderlich. Adaptive debug and diagnosis without fault dictionaries. In *Proceedings IEEE European Test Symposium (ETS)*, pages 7–12, May 2007.

C. Hora, R. Segers, S. Eichenberger, and M. Lousberg. An effective diagnosis method to support yield improvement. In *International Test Conference, ITC*, pages 260–269, 2002.

L.M. Huisman. Diagnosing arbitrary defects in logic designs using single location at a time (slat). *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 23 (1):91–101, Jan. 2004.

U. Ingelsson. Investigation into voltage and process variation-aware manufacturing test. *Ph.D Thesis, University of Southampton*, July 2009. URL `http://eprints.soton.ac.uk/67440/`.

U. Ingelsson, P. Rosinger, S. S. Khursheed, B. M. Al-Hashimi, and P. Harrod. Resistive bridging faults DFT with adaptive power management awareness. *Proceedings IEEE Asian Test Symposium (ATS)*, pages 101–106, October 2007.

U. Ingelsson, B. M. Al-Hashimi, and P. Harrod. Variation aware analysis of bridging fault testing. *Asian Test Symposium, 2008. ATS '08. 17th*, pages 206–211, Nov. 2008.

U. Ingelsson, B.M. Al-Hashimi, S. Khursheed, S.M. Reddy, and P. Harrod. Process variation-aware test for resistive bridges. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 28(8):1269–1274, Aug. 2009.

Datasheet Intel. Intel: Pxa270 processor datasheet, November 2007. URL http://www.phytec.com/pdf/datasheets/PXA270_DS.pdf.

V. Iyengar, Jinjun Xiong, S. Venkatesan, V. Zolotov, D. Lackey, P. Habitz, and C. Visweswariah. Variation-aware performance verification using at-speed structural test and statistical timing. pages 405–412, Nov. 2007.

N. K. Jha and S. K. Gupta. *Testing of Digital Systems*. Cambridge University Press, Cambridge, 2003. ISBN 0521773563.

S. Johnson. Residual charge on the faulty floating gate CMOS transistor. *Test Conference, 1994. Proceedings., International*, pages 555–561, 1994.

R. Kapur, S. Mitra, and T.W. Williams. Historical perspective on scan compression. *IEEE Design & Test of Computers*, 25(2):114–120, March-April 2008.

T. Karnik and P. Hazucha. Characterization of soft errors caused by single event upsets in CMOS processes. *IEEE Transactions on Dependable and Secure Computing*, 1(2):128–143, April-June 2004.

M. Keating, D. Flynn, R. Aitken, A. Gibbons, and K. Shi. *Low Power Methodology Manual: For System-on-Chip Design*. Springer, 2007.

S. Khursheed, U. Ingelsson, P. Rosinger, B. M. Al-Hashimi, and P. Harrod. Bridging fault test method with adaptive power management awareness. *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, 27(6), June 2008.

Kee Sup Kim, S. Mitra, and P.G. Ryan. Delay defect characteristics and testing strategies. *IEEE Design & Test of Computers*, 20(5):8–16, Sept.-Oct. 2003a.

N.S. Kim, T. Austin, D. Baauw, T. Mudge, K. Flautner, J.S. Hu, M.J. Irwin, M. Kandemir, and V. Narayanan. Leakage current: Moore's law meets static power. *Computer*, 36(12):68–75, Dec. 2003b.

R.K. Krishnarnurthy, A. Alvandpour, V. De, and S. Borkar. High-performance and low-power challenges for sub-70 nm microprocessor circuits. *Custom Integrated Circuits Conference, 2002. Proceedings of the IEEE 2002*, pages 125–128, 2002.

B. Kruseman and M. Heiligers. On test conditions for the detection of open defects. In *Proceedings of the conference on Design, automation and test in Europe (DATE)*, pages 896–901, Munich, Germany, March 2006.

B. Kruseman, A.K. Majhi, G. Gronthoud, and S. Eichenberger. On hazard-free patterns for fine-delay fault testing. *Proceedings IEEE International Test Conference (ITC)*, pages 213–222, Oct. 2004.

S. Kundu, S.T. Zachariah, S. Sengupta, and R. Galivanche. Test challenges in nanometer technologies. *Journal of Electronic Testing: Theory and Applications*, pages 209–218, 2001.

David E. Lackey, Paul S. Zuchowski, Thomas R. Bednar, Douglas W. Stout, Scott W. Gould, and John M. Cohn. Managing power and performance for system-on-chip designs using voltage islands. In *ICCAD '02: Proceedings of the 2002 IEEE/ACM international conference on Computer-aided design*, pages 195–202, New York, NY, USA, 2002.

Soongsoo Lee and T. Sakurai. Run-time voltage hopping for low-power real-time systems. *Design Automation Conference*, pages 806–809, 2000.

Wan-Ping Lee, Hung-Yi Liu, and Yao-Wen Chang. Voltage island aware floorplanning for power and timing optimization. In *ICCAD '06: Proceedings of the 2006 IEEE/ACM international conference on Computer-aided design*, pages 389–394, New York, NY, USA, 2006. ACM.

Chin Jen Lin and S.M. Reddy. On delay fault testing in logic circuits. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 6(5):694–703, September 1987.

S.-H. Lo, D.A. Buchanan, Y. Taur, and W. Wang. Quantum-mechanical modeling of electron tunneling current from the inversion layer of ultra-thin-oxide nmosfet's. *Electron Device Letters, IEEE*, 18(5):209–211, May 1997.

LP. LP Solver, November 2009. URL http://lpsolve.sourceforge.net/.

Xiang Lu, Zhuo Li, Wangqi Qiu, D. M. H. Walker, and Weiping Shi. Longest path selection for delay test under process variation. *IEEE Transactions on Computer-Aided Design*, 24(12): 1924–1929, December 2005.

Tao Luo, David Newmark, and David Z. Pan. Total power optimization combining placement, sizing and multi-vt through slack distribution management. In *Proceedings of the Asia and South Pacific Design Automation Conference, ASP-DAC*, pages 352–357, 2008.

T. Maeda and K. Kinoshita. Precise test generation for resistive bridging faults of CMOS combinational circuits. In *Proceedings IEEE International Test Conference (ITC)*, pages 510–519, October 2000.

A.K. Majhi and V.D. Agrawal. Delay fault models and coverage. In *Eleventh International Conference on VLSI Design*, pages 364–369, Jan 1998.

T. M. Mak. and S. Nassif. Process variation and stochastic design and test. 23(6):436–437, June 2006.

S. Mandava, S. Chakravarty, and S. Kundu. On detecting bridges causing timing failures. In *International Conference on Computer Design, (ICCD '99)*, pages 400–406, 1999.

T.E. Marchok, A. El-Maleh, W. Maly, and J. Rajski. A complexity analysis of sequential ATPG. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 15(11): 1409–1423, Nov 1996.

S. M. Martin, K. Flautner, T. Mudge, and D. Blaauw. Combined dynamic voltage scaling and adaptive body biasing for lower power microprocessors under dynamic workloads. In *Proceedings of the International Conference on Computer-Aided Design (ICCAD)*, San Jose, CA, USA, November 2002.

E. J. McCluskey and F. Buelow. IC quality and test transparency. In *Proc. 'New Frontiers in Testing'. International Test Conference*, pages 295–301, September 12–14, 1988.

E.J. McCluskey and F. Buelow. IC quality and test transparency. *IEEE Transactions on Industrial Electronics*, 36(2):197–202, May 1989.

P. J. Meaney, S. B. Swaney, P. N. Sanda, and L. Spainhower. Ibm z990 soft error detection and recovery. *IEEE Trans. Device Mater. Rel.*, 5(3):419–427, 2005.

S.D. Millman, E.J. McCluskey, and J.M. Acken. Diagnosing CMOS bridging faults with stuck-at fault dictionaries. In *Proceedings IEEE International Test Conference (ITC)*, pages 860–870, September 1990.

R.R. Montanes, J.P. de Gyvez, and P. Volf. Resistance characterization for weak open defects. *IEEE Design & Test of Computers*, 19(5):18–26, 2002.

S. Narayanan, R. Srinivasan, R.P. Kunda, M.E. Levitt, and S. Bozorgui-Nesbat. A fault diagnosis methodology for the UltraSPARC^{TM}-I microprocessor. *European Design and Test Conference, 1997. ED&TC 97. Proceedings*, pages 494–500, Mar 1997.

S. Nassif. Delay variability: sources, impacts and trends. In *IEEE International Solid-State Circuits Conference, ISSCC*, pages 368–369, 2000.

Cellular News. Mobile phone subscribers pass 4 billion mark, December 2008. URL `http://www.cellular-news.com/story/35298.php`.

N. Nicolici and B. M. Al-Hashimi. *Power-Constrained Testing of VLSI Circuits*. Kluwer Academic Publishers, 2003.

T. Niermann and J.H. Patel. Hitec: a test generation package for sequential circuits. In *Proceedings of the European Conference on Design Automation, (EDAC)*, pages 214–218, Feb 1991.

J. Park, M. Naivar, R. Kapur, M.R. Mercer, and T.W. Williams. Limitations in predicting defect level based on stuck-at fault coverage. In *12th IEEE VLSI Test Symposium, VTS*, pages 186–191, Apr 1994.

J.H. Patel. Stuck-at fault: a fault model for the next millennium. *Proceedings IEEE International Test Conference (ITC)*, page 1166, Oct 1998.

Massoud Pedram. Power minimization in IC design: principles and applications. *ACM Trans. Des. Autom. Electron. Syst.*, 1(1):3–56, 1996.

K. Pokhrel. Physical and silicon measures of low power clock gating success: An apple to apple case study. In *SNUG Papers and Presentations*, 2007. http://www.snug-universal.org/cgi-bin/search/search.cgi?San+Jose+2007.

I. Polian, P. Engelke, B. Becker, S. Kundu, J.-M. Galliere, and M. Renovell. Resistive bridge fault model evolution from conventional to ultra deep submicron. In *IEEE VLSI Test Symposium, VTS*, pages 343–348, May 2005.

I. Pomeranz and S.M. Reddy. A same/different fault dictionary: An extended pass/fail fault dictionary with improved diagnostic resolution. In *Proceedings Design, Automation, and Test in Europe (DATE)*, March 2008.

I. Pomeranz and S.M. Reddy. Design-for-testability for path delay faults in large combinational circuits using test points. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 17(4):333–343, April 1998.

I. Pomeranz and S.M. Reddy. On the generation of small dictionaries for fault location. In *Proceedings International Conference on Computer-Aided Design (ICCAD)*, pages 272–279, November 1992.

A.K. Pramanick and S.M. Reddy. On the fault coverage of gate delay fault detecting tests. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 16(1):78–94, Jan 1997.

S. Rafiq, A. Ivanov, S. Tabatabaei, and M. Renovell. Testing for floating gates defects in CMOS circuits. *Test Symposium, 1998. ATS '98. Proceedings. Seventh Asian*, pages 228–236, Dec 1998.

M. Renovell, P. Huc, and Y. Bertrand. The concept of resistance interval: a new parametric model for realistic resistive bridging fault. In *Proceedings of the VLSI Test Symposium (VTS)*, pages 184–189, April 1995.

M. Renovell, P. Huc, and Y. Bertrand. Bridging fault coverage improvement by power supply control. In *Proceedings of the VLSI Test Symposium (VTS)*, pages 338–343, April 1996.

M. Renovell, F. Azais, and Y. Bertrand. Detection of defects using fault model oriented sequences. *Journal of Electronic Testing: Theory and Applications*, 14:13–22, February 1999.

R. Rodriguez-Montanes, E.M.J.G. Bruis, and J. Figueras. Bridging defects resistance measurements in a CMOS process. *Proceedings of the International Test Conference (ITC)*, pages 892–899, Sep 1992.

R. Rodriguez-Montanes, D. Arumi, and J. Figueras. Effectiveness of very low voltage testing of bridging defects. *Electronics Letters*, 42(19):1083–1084, Sept. 2006.

R. Rodriguez-Montanes, D. Arumi, J. Figueras, S. Einchenberger, C. Hora, B. Kruseman, M. Lousberg, and A.K. Majhi. Diagnosis of full open defects in interconnecting lines. *VLSI Test Symposium, 2007. 25th IEEE*, pages 158–166, May 2007.

A. Rousset, A. Bosio, P. Girard, C. Landrault, S. Pravossoudovitch, and A. Virazel. DERRIC: A Tool for Unified Logic Diagnosis. In *Proceedings IEEE European Test Symposium (ETS)*, pages 13–20, May 2007.

Gareth Roy, Andrew R. Brown, Fikru Adamu-Lema, Scott Roy, and Asen Asenov. Simulation study of individual and combined sources of intrinsic parameter fluctuations in conventional nano-mosfets. *IEEE Transactions on Electron Devices*, 53(12):3063–3070, Dec. 2006a.

K. Roy, S. Mukhopadhyay, and H. Mahmoodi-Meimand. Leakage current mechanisms and leakage reduction techniques in deep-submicrometer cmos circuits. *Proceedings of the IEEE*, 91(2):305–327, Feb 2003.

Kaushik Roy, T.M. Mak, and K.-T. Cheng. Test consideration for nanometer-scale CMOS circuits. 2(3):128–136, March-April 2006b.

E.M. Rudnick and J.H. Patel. Efficient techniques for dynamic test sequence compaction. *IEEE Transactions on Computers*, 48(3):323–330, Mar 1999.

V. R. Sar-Dessai and D. M. H. Walker. Resistive bridge fault modeling, simulation and test generation. In *Proceedings of the International Test Conference (ITC)*, pages 596–605, Atlantic City, NJ, USA, September 1999.

M. T. Schmitz, B. M. Al-Hashimi, and P. Eles. *System-level design techniques for energy-efficient embedded systems*. Kluwer Academic Publishers, 2004.

J. Segura, J.L. Rossello, J. Morra, and H. Sigg. A variable threshold voltage inverter for CMOS programmable logic circuits. *Solid-State Circuits, IEEE Journal of*, 33(8):1262–1265, Aug 1998.

S. Sengupta, S. Kundu, S. Chakravarty, P. Parvathala, R. Galivanche, G. Kosonocky, M. Rodgers, and T. M. Mak. Defect-based test: A key enabler for successful migration to structural test. In *Intel Technology Journal*, pages 1–14, Q1 1999.

Jaewon Seo, Taewhan Kim, and N. D. Dutt. Optimal integration of inter-task and intra-task dynamic voltage scaling techniques for hard real-time applications. In *ICCAD '05: Proceedings of the 2005 IEEE/ACM International conference on Computer-aided design*, pages 450–455, Washington, DC, USA, 2005. IEEE Computer Society.

C. Shi and R. Kapur. How power-aware test improves reliability and yield, September 2004. URL http://www.eedesign.com/showArticlejhtml?articleID=47208594.

T. Shinogi, T. Kanbayashi, T. Yoshikawa, S. Tsuruoka, and T Hayashi. Faulty resistance sectioning technique for resistive bridging fault atpg systems. In *Proceedings of the Asian Test Symposium (ATS)*, pages 76–81, Kyoto, Japan, November 2001.

J. Smith. Rapid implementation of low power microprocessors. *Proceedings ACM/IEEE Design Automation Conference (DAC)*, 4(2), October 2009. URL http://www.dac.com/newsletter/shownewsletter.aspx?newsid=54.

J.M. Soden and C.F. Hawkins. Test considerations for gate oxide shorts in CMOS ICs. *IEEE Design & Test of Computers*, 3(4):56–64, Aug. 1986.

J.M. Soden, R.K. Treece, M.R. Taylor, and C.F. Hawkins. CMOS IC stuck-open-fault electrical effects and design considerations. In *International Test Conference, ITC*, pages 423–430, Aug 1989.

J.J.T. Sousa, F.M. Goncalves, and J.P. Teixeira. IC defects-based testability analysis. In *Proceedings IEEE International Test Conference (ITC)*, pages 500–509, Oct 1991.

S. Spinner, I. Polian, P. Engelke, B. Becker, M. Keim, and Wu-Tung Cheng. Automatic test pattern generation for interconnect open defects. *VLSI Test Symposium, 2008. VTS 2008. 26th IEEE*, pages 181–186, 27 2008-May 1 2008.

A. Sreedhar, A. Sanyal, and S. Kundu. On modeling and testing of lithography related open faults in Nano-CMOS circuits. *Design, Automation and Test in Europe, 2008. DATE '08*, pages 616–621, March 2008.

Ashish Srivastava and Dennis Sylvester. Minimizing total power by simultaneous vdd/vth assignment. In *ASP-DAC '03: Proceedings of the 2003 Asia and South Pacific Design Automation Conference*, pages 400–403, New York, NY, USA, 2003.

S. Tani, M. Teramoto, T. Fukazawa, and K. Matsuhiro. Efficient path selection for delay testing based on partial path evaluation. *VLSI Test Symposium, VTS 98.*, pages 188–193, April 1998.

Vivek Tiwari, Deo Singh, Suresh Rajgopal, Gaurav Mehta, Rakesh Patel, and Franklin Baez. Reducing power in high-performance microprocessors. In *Proceedings of the 35th annual Design Automation Conference, DAC*, pages 732–737, 1998.

N.A. Touba. Survey of test vector compression techniques. *IEEE Design & Test of Computers*, 23(4):294–303, April 2006.

Nur A. Touba and Edward J. McCluskey. Pseudo-random pattern testing of bridging faults. In *Proceedings IEEE ICCD*, pages 54–60, October 1997.

Nur A. Touba and Edward J. McCluskey. Test point insertion based on path tracing. In *Proceedings IEEE VLSI Test Symposium (VTS)*, pages 2–8, April 1996.

A. Veneris, R. Chang, M.S. Abadir, and M. Amiri. Fault equivalence and diagnostic test generation using atpg. In *Proceedings of the 2004 International Symposium on Circuits and Systems, ISCAS*, pages 221–224, May 2004.

H. Vranken, F.S. Sapei, and H.-J. Wunderlich. Impact of test point insertion on silicon area and timing during layout. In *Design, Automation and Test in Europe Conference and Exhibition,*, Feb. 2004.

R. L. Wadsack. Fault modeling and logic simulation of CMOS and MOS integrated circuits. *The Bell System Technical Journal*, 57:1449–1474, May 1978. URL http://adsabs.harvard.edu/abs/1978ATTTJ..57.1449W.

J. Waicukauski and E. Lindbloom. Failure Diagnosis of Structured VLSI. *IEEE Design & Test of Computers*, 6(4):49–60, August 1989.

L.T. Wang, C.W. Wu, and X. Wen. *Vlsi Test Principles and Architectures: Design for Testability*. Elsevier Morgan Kaufmann Publishers, San Francisco, 2006. ISBN 9780123705976.

Seongmoon Wang and S.K. Gupta. Atpg for heat dissipation minimization during test application. *IEEE Transactions on Computers*, 47(2):256–262, Feb 1998.

Neil H. E. Weste and Kamran Eshraghian. *Principles of CMOS VLSI design: a systems perspective*. Addison-Wesley Publishing Co., 1994.

T.W. Williams and N.C. Brown. Defect level as a function of fault coverage. *IEEE Transactions on Computers*, C-30(12):987–988, Dec. 1981.

T.W. Williams and K.P. Parker. Design for testability–A survey. *Proceedings of the IEEE*, 71 (1):98–112, Jan. 1983.

R Wray. Half world's population "will have mobile phone by end of year", September 2008. http://www.guardian.co.uk/technology/2008/sep/26/mobilephones.unitednations.

Jue Wu and E.M. Rudnick. A diagnostic fault simulator for fast diagnosis of bridge faults. In *Proceedings IEEE International Conference on VLSI Design (ICVD)*, pages 498–505, January 1999.

Li Yan and J. R. English. Economic cost modeling of environmental-stress-screening and burn-in. *IEEE Trans. Rel.*, 46(2):275–282, 1997.

M. Yilmaz, K. Chakrabarty, and M. Tehranipoor. Test-pattern grading and pattern selection for small-delay defects. In *26th IEEE VLSI Test Symposium, VTS*, pages 233–239, May 2008.

N.B. Zain Ali and M. Zwolinski. Dynamic voltage scaling aware delay fault testing. In *Proceedings of the IEEE European Test Symposium (ETS)*, pages 15–20, Southampton, UK, May 2006.

Noohul Basheer Zain Ali. An investigation of delay fault testing for multi voltage design. *Ph.D Thesis, University of Southampton, UK*, 2009. URL `http://eprints.soton.ac.uk/66389/`.

R. Zarr and J. Taylor. Powerwise adaptive voltage scaling (avs) technology webinar, February 2009. http://www.national.com/onlineseminar/2009/avs_overview/adaptive-voltage-scaling-overview_customer.pdf.

zChaff. zChaff Boolean Satisfiability problem solver, March 2007. URL `http://www.princeton.edu/~chaff/zchaff.html`.

Xiaodeng Zhang and K. Roy. Peak power reduction in low power bist. In *First International Symposium on Quality Electronic Design, ISQED*, pages 425–432, 2000.

Quming Zhou and K. Mohanram. Gate Sizing to Radiation harden Combinational logic. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 25(1): 155–166, Jan. 2006.

Wei Zou, Wu-Tung Cheng, and S.M. Reddy. Bridge Defect Diagnosis with Physical Information. In *Proceedings IEEE Asian Test Symposium (ATS)*, pages 248–253, December 2005.

Wei Zou, Wu-Tung Cheng, and S.M. Reddy. Interconnect open defect diagnosis with physical information. *Test Symposium, 2006. ATS '06. 15th Asian*, pages 203–209, November 2006.

Wei Zou, Wu-Tung Cheng, S.M. Reddy, and Huaxing Tang. Speeding up effect-cause defect diagnosis using a small dictionary. In *IEEE VLSI Test Symposium*, pages 225–230, May 2007.