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

Investigation into Voltage and Process Variation-Aware Manufacturing Test

by

Bo Urban Ingelsson

A thesis submitted for the
degree of Doctor of Philosophy

July 2009

UNIVERSITY OF SOUTHAMPTON

ABSTRACT

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

Doctor of Philosophy

Investigation into Voltage and Process Variation-Aware Manufacturing Test

by Bo Urban Ingelsson

Increasing integration and complexity in IC design provides challenges for manufacturing testing. This thesis studies how process and supply voltage variation influence defect behaviour to determine the impact on manufacturing test cost and quality. The focus is on logic testing of static CMOS designs with respect to two important defect types in deep submicron CMOS: resistive bridges and full opens.

The first part of the thesis addresses testing for resistive bridge defects in designs with multiple supply voltage settings. To enable analysis, a fault simulator is developed using a supply voltage-aware model for bridge defect behaviour. The analysis shows that for high defect coverage it is necessary to perform test for more than one supply voltage setting, due to supply voltage-dependent behaviour. A low-cost and effective test method is presented consisting of multi-voltage test generation that achieves high defect coverage and test set size reduction without compromise to defect coverage. Experiments on synthesised benchmarks with realistic bridge locations validate the proposed method.

The second part focuses on the behaviour of full open defects under supply voltage variation. The aim is to determine the appropriate value of supply voltage to use when testing. Two models are considered for the behaviour of full open defects with and without gate tunnelling leakage influence. Analysis of the supply voltage-dependent behaviour of full open defects is performed to determine if it is required to test using more than one supply voltage to detect all full open defects. Experiments on synthesised benchmarks using an extended version of the fault simulator tool mentioned above, measure the quantitative impact of supply voltage variation on defect coverage.

The final part studies the impact of process variation on the behaviour of bridge defects. Detailed analysis using synthesised ISCAS benchmarks and realistic bridge model shows that process variation leads to additional faults. If process variation is not considered in test generation, the test will fail to detect some of these faults, which leads to test escapes. A novel metric to quantify the impact of process variation on test quality is employed in the development of a new test generation tool, which achieves high bridge defect coverage. The method achieves a user-specified test quality with test sets which are smaller than test sets generated without consideration of process variation.

Contents

List of Figures

List of Tables

DECLARATION OF AUTHORSHIP

I, Bo Urban Ingelsson, declare that the thesis entitled *Investigation into Voltage and Process Variation-Aware Manufacturing Test* and the work presented in the thesis are both my own, and have been generated by me as the result of my own original research. I confirm that:

- this work was done wholly or mainly while in candidature for a research degree at this University;
- where any part of the thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated;
- where I have consulted the published work of others, this is always clearly attributed;
- 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;
- I have acknowledged all main sources of help;
- 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;
- parts of this work have been published as listed in Section 1.6

Signed:

Date:

Acknowledgements

Throughout this Ph.D. project, I have benefited from the support and guidance of my supervisor Prof. Bashir M. Al-Hashimi, to whom I am deeply thankful. I am grateful for the funding and research facilities provided by the School of Electronics and Computer Science of the University of Southampton.

A big "thank you" to every-one who has been helpful to me in preparation of the conference and journal papers, especially my co-authors Dr. Paul Rosinger, Saqib Khursheed, Dr. Peter Harrod and Prof. Sudhakar Reddy.

This Ph.D. project has benefited from discussions with several very helpful people, and I want to thank them. Thank you Prof. Sandip Kundu, Dr. Harry Oldham, Dr. Piet Engelke, Dr. Ilia Polian, Prof. Krishnendu Chakrabarty, Prof. David Flynn, Paul Hughes, Matthew Collins, Ross Torkington, Jonathan Barker, Rishad A. Shafik, Dr. Noohul Basher Zain Ali, Dr. Biswajit Mishra, Dr. Matthew Swabey, Dr. Ashish Darbari, Karthik Baddam, Mustafa Imran Ali and Dr. Simon Ogg.

I am thankful to Prof. Erik Larsson, Erik Jan Marinissen and Dr. Sandeep Goel for introducing me to test research and for encouraging me to pursue this Ph.D.

Finally, I would like to thank my parents, Bo and Ann-Sofie Ingelsson, and my siblings Emma Ingelsson Alkbring, Ida and Lisa Ingelsson for their love, support and understanding over the past three years.

S.D.G.

Chapter 1

Introduction

Since the technology for manufacturing integrated circuits (ICs) was invented, the level of integration has continued to increase [1], leading to increased functionality, increased performance and ever smaller transistors. This development has lead to challenges in manufacturing testing of ICs. The research community has addressed many of these challenges, but among the ones that remain to be addressed is the problem of testing ICs in the presence of different types of variation. Two types of variation are addressed in this thesis with regard to their influence on manufacturing testing: process and supply voltage variation.

To see why these types of variation should be studied, consider that recent designs have transistors with a length of 45nm or below. The manufacturing of such small transistors is prone to process variation, a fact that until recently has been ignored in development on manufacturing test methods. Similarly, supply voltage variation, as is employed in several modern low-power designs, has not been considered in terms of its impact on manufacturing test. State-of-the-art test methods tend to use abstract models of how defects occur and behave to simplify test generation and several techniques rely on the ability of such tests to detect also defects that are not explicitly modelled. While these abstract models do not consider process variation and supply voltage variation it is important to study the influence of such variation on manufacturing test.

This thesis studies the impact of supply voltage variation and process variation on manufacturing testing with particular regard to logic testing of static CMOS circuits. Two important defect types are investigated, namely resistive bridge defects and full open defects, with the aim of developing test methods to cope with any negative influence of the considered variation on test cost and test quality.

The chapter at hand provides background on relevant concepts used in the thesis. In particular, the chapter includes background on how the supply voltage is varied in low power IC design (Section 1.1 and Section 1.2) and background on process variation

(Section 1.3). Furthermore, there is a review of many concepts in manufacturing testing that are used in the thesis (Section 1.4). Subsequently, the outline of the thesis is given along with a list of publications that have resulted from the presented research.

1.1 Dynamic Voltage and Frequency Scaling

The complexity of integrated circuits in battery-driven applications has been steadily increasing. The usability of such applications depends on the battery life time [2] and there are two ways of increasing the battery life time. Either improve the battery or reduce the power consumption of the device. Development of new battery technology has been lagging behind [3]. Therefore, the trend is to reduce the power consumption of integrated circuits by low power circuit design. Two main low-power design techniques, namely Multi-Voltage design and Dynamic Voltage and Frequency Scaling (DVFS), adjust the supply voltage to limit power consumption. This thesis studies the impact of such supply voltage variation on the behaviour of bridge and open defects (Chapter 3 and Chapter 4) and investigate how tests can be generated to detect defects under supply voltage variation. Therefore, this section and the next discuss DVFS and multi-voltage design respectively.

DVFS is a technique used in low-power IC design, which uses a range of operational supply voltages to implement power modes. DVFS utilises the fact that the power that is due to switching of logic states in a digital CMOS circuit is proportional to the square of the supply voltage. This relation is shown in Equation 1.1 where P_{Dyn} is the dynamic power consumption due to switching activity, f is the clock frequency and Vdd is the supply voltage.

$$P_{Dyn} \propto f \cdot Vdd^2 \tag{1.1}$$

This shows that using a lower supply voltage leads to less dynamic power consumption. However, a lower supply voltage also leads to slower operation of logic gates and therefore lower performance. To see how the dynamic power consumption P_{Dyn} relates to the total power consumption P_{Tot}, consider Equation 1.2. The total power consumption P_{Tot} is the sum of the dynamic power consumption P_{dyn} and the static power consumption from leakage P_{Stat}. This means that techniques such as DVFS save on dynamic power consumption but there is also the static power consumption to consider.

$$P_{Tot} = P_{Dyn} + P_{Stat} \tag{1.2}$$

DVFS is an Adaptive Power Management technique (APM) [4], which means that power management is integrated on-chip. Other APM implementations include decoupling

of clock and supply voltage for circuitry that are not utilised and by doing so power consumption is reduced. Decoupling the clock from a circuit is called clock gating and is used to reduce dynamic power consumption. The other method, decoupling of supply voltage, is called power gating and is used to reduce leakage power consumption. DVFS reduces the dynamic power consumption by scaling down the operational clock frequency and the circuit supply voltage (Vdd) [5]. The scaling performed in DVFS is dynamic with regard to the workload of the system, so that power can be saved using low-performance mode when the IC is used but under-utilised and to provide high-performance in high-power mode when the IC is heavily used [5, 2]. Typically, a DVFS design has a set of discrete supply voltage and clock frequency settings to implement power modes.

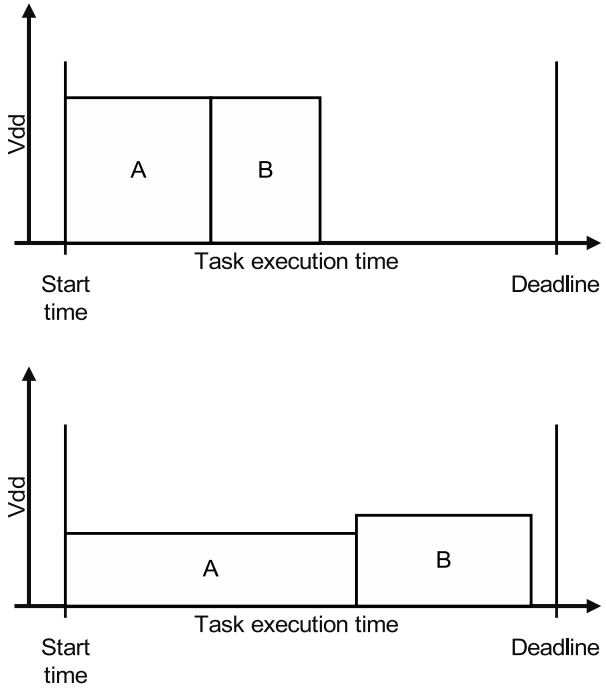


FIGURE 1.1: Task schedules with (bottom) and without (top) DVFS

Figure 1.1 shows an example of a scenario in which DVFS can save power. Two tasks, A and B, are to be performed in sequence and finish before a deadline. The deadline is a point in time when the output data of the tasks are required. The top graph shows both tasks operating at the same voltage and clock frequency, as is the case in a design that operates using a single high-performance supply voltage setting. There is idle time between the finish time of the second task (B) and the deadline. This means that the two tasks can be performed slower and still meet the deadline. In a DVFS design, this slack time is traded for lower clock frequency, as shown by the bottom graph of Figure 1.1 where the tasks take longer time. With a lowered clock frequency, it is possible to lower the supply voltage accordingly, as shown in Figure 1.1 where the height of the rectangles that represent the tasks have been adjusted to reflect different supply voltage settings for the two tasks. Lowering the supply voltage increases the transition delay of logic gates, therefore lowering the supply voltage is only possible when the clock frequency is

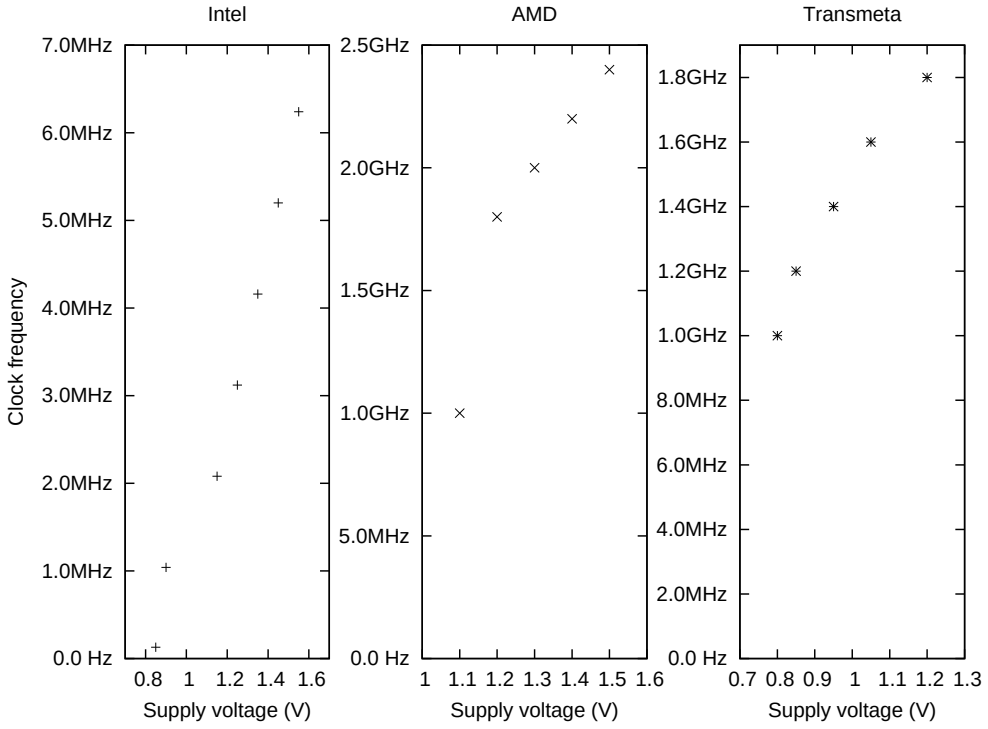


FIGURE 1.2: Supply voltage and clock frequency settings for three DVFS designs

low enough to accommodate the increased transition delay.

Examples of designs with DVFS implemented include ARM processors with IEM (Intelligent Energy Management) [6], XScale Intel processors [7], AMD Athlon 64 processors [8], and LongRun2 Transmeta processors [9]. Figure 1.2 shows three examples with the recommended supply voltage and clock frequency settings.

The switching of power modes in designs with DVFS can be controlled either by a predetermined application-specific power schedule [5], by a system that attempts to predict future process requirements [10] or by a system that is scheduling tasks depending on the battery current profile [2]. In order to implement DVFS capability in a design, extra hardware is included, such as a power management controller [6], level-shifters [11] and for the capability of continuous voltage scaling, particular pipeline latches [12]. Further savings in terms of power consumption can be gained by scaling the body bias voltage in conjunction with the supply voltage [13]. This is a technique that limits the static power consumption (leakage).

When discussing systems with multiple supply voltage settings, it is useful to have one particular setting for reference, the nominal supply voltage. In this thesis, the nominal supply voltage refers to the supply voltage that is recommended for a particular gate library. Often a gate library that is designed for a particular VLSI technology, has been characterised for a particular supply voltage. For example, the two gate libraries that are used for experimentation in this thesis, one for 0.12μm technology [14] and the other

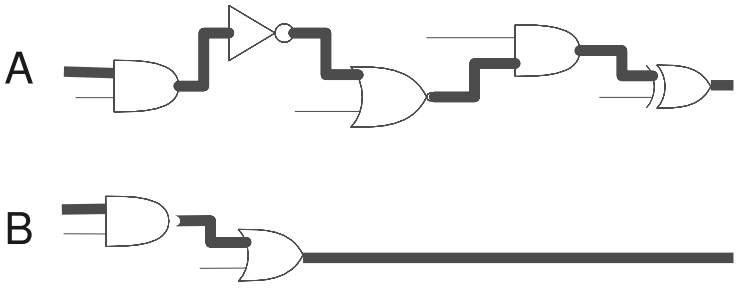
for 45nm technology [15], are associated with the nominal supply voltages 1.2V and 0.9V respectively. The nominal supply voltages are often set for a high performance, which also means relatively high power consumption. To save power, DVFS will typically introduce additional supply voltage settings below the nominal supply voltage.

1.2 Multi-Voltage Design

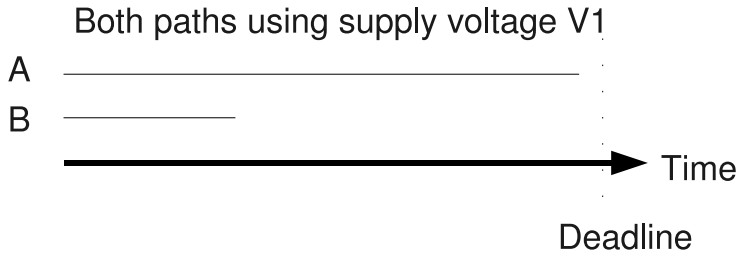
Multi-voltage design is another low-power design technique that adjusts the supply voltage. Multi-voltage design partitions the internal logic of the chip into multiple voltage regions, each with its own supply voltage and corresponding supply voltage rail infrastructure [16]. It is based on the observation that in a modern design, different blocks of circuitry have different performance objectives and constraints. A processor, for instance, determines the system performance and needs to run as fast as the semiconductor technology will allow. In this case, a relatively high supply voltage is required. On the other hand, circuitry that conducts inter-system communication using a given protocol may run at a fixed, relatively low frequency dictated more by the protocol than the technology [16]. The low frequency allows the considered circuitry to operate on a relatively low supply voltage. By operating blocks that are not critical to system performance using a lower supply voltage, power consumption can be reduced in accordance to Equation 1.1.

Multi-voltage design can also reduce power consumption within blocks if the path delay distribution allows [17]. In a typical design with a required clock frequency, all paths will have a total delay less than the clock period. Multi-voltage design exploits the observation that some of the paths will have a significantly lower total delay, which means that the circuitry on such paths can be operated at a lower supply voltage without impact on the overall performance. The basic idea in this type of multi-voltage design is to identify the non-critical paths and to power the gates in those paths with a lower voltage, which leads to reduction of power consumption. The example in Figure 1.3(a) shows two paths, A and B. There are fewer logic gates on path B than on path A and path B will reach the final state before path A. Therefore, path B is a candidate for being operated with lower voltage. Both paths, A and B, have the same timing constraint which is set by the clock period, as illustrated by Figure 1.3(b) where both paths are using the same supply voltage V1. As mentioned above, path B is a candidate for being operated with a lower voltage V2 (V2<V1). Figure 1.3(c) shows that with supply voltage V2, path B has a total delay that is better adjusted to the clock period. By using the lower supply voltage V2 on path B, less dynamic power is consumed on path B.

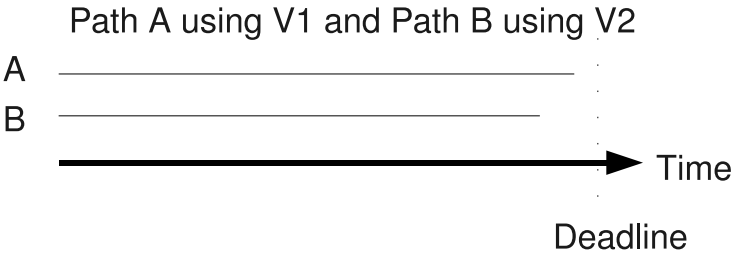
To implement a design with multiple voltage regions, as in the case of multi-voltage, level shifters are necessary to interface between different blocks. Signals crossing from



(a) Two paths with different total delay



(b) Timing diagram with both paths on the same supply voltage



(c) Timing diagram with different supply voltage settings for the two paths

FIGURE 1.3: Example of multi-voltage design principle

one voltage region to another have to be interfaced through level shifters which shift the signals to the appropriate logic levels [16].

1.3 Process Variation

Besides supply voltage variation, this thesis is concerned with process variation and how it influences defect behaviour, which has implications for manufacturing test. This thesis studies the impact of process variation in the context of resistive bridging faults, which is an important defect type in deep submicron designs (see below). In the study, test methods are developed to achieve high test quality in the presence of such variation (Chapter 5). The following provides background information about process variation.

Recently, IC technology has reached such levels of integration, which involves miniaturisation, that the length of transistors are only tens of nanometres. Designs with transistors that have features which are shorter than 0.5μm are referred to as Deep Sub-

Micron designs (DSM) [18]. In DSM, the gate oxide (or dielectric) of the transistors can be very thin, down to ten atom layers [19]. Even small variations in the parameters of such small transistors has impact on the performance of the manufactured circuit. As the dimensions scale down to allow more integration on-chip, it is getting increasingly difficult to control all the IC parameters to the values that are specified in the design of the circuit [20]. These IC parameters include, among others, the concentration of doping atoms in the N-well and P-well substrates [21], the thickness of the gate oxide/dielectric TOX [22] and the length L of a given transistor [23]. It is often found that the actual values of some IC parameters vary across a wafer (the silicon substrate on which several ICs of the same design are manufactured next to each other in the same process), and also within each die, so that manufactured ICs from the same manufacture are slightly different [24]. This is called process variation. Process variation is considered either across different dies or within each die [22, 24] depending on the distance between circuitry that has correlated IC parameters. Within-die variation has no correlation with parameter values in other dies. The magnitude of variation on different parameters has been studied in [22]. Most of the IC parameter variations are tolerable and the chips will function normally, due to a sufficient noise margin, but variation may impact performance in terms of maximum clock frequency or power consumption [20]. However, these process variations become larger relative to the intended values of the IC parameters as the dimensions of transistors are scaled down in the pursuit of higher integration, higher performance and lower power consumption. Therefore, process variation is a challenge particularly for designs implemented in DSM technology [20].

It should be noted that some of the causes of process variation are unpredictable and unavoidable. To review some of these variations in terms of their mechanisms and characteristics, consider random dopant fluctuations, sub-wavelength lithography effects and line edge roughness [24]. The concept of random dopant fluctuations express that there are so few doping atoms in the substrate of a transistor, that the location and distribution of these atoms influence the value of the transistor threshold voltage VT [21], which is the gate-to-source voltage that makes the transistor start to conduct. The study in [21] noted that the atomistic effect of the location of doping atoms leads to a bell-shaped probability distribution for the transistor threshold voltage VT. Because of the very local aspects of the distribution of doping atoms, random dopant fluctuations will affect adjacent transistors differently, i.e. there is no correlation in the variation. Up to 10% variation in the VT value has been observed [20, 25, 26]. The transistor length L is impacted both by sub-wavelength photo-lithography effects and line edge roughness [23, 24]. Photo-lithography is the process used in IC manufacturing to draw structures onto the wafer using light shining through masks, that contain the designed pattern, onto light-sensitive material, which will then be etched away depending on if it was illuminated or not. In pursuit of the ability to manufacture even smaller transistors, the resolution of the photo-lithography has traditionally been improved by moving deeper into ultraviolet spectrum of light. However, the wavelength of the optical source used for lithography has

not improved for nearly a decade. To manufacture ever smaller transistor that measure shorter than the wavelength of the light source, sub-wavelength lithography has been developed. Sub-wavelength lithography is associated with diffraction effects. Despite efforts of offsetting these effects, they still lead to variation in the manufacturing process, so that dies are affected depending on their relative position on the wafer to the light source. These effects have been observed in the forbidden pitch phenomenon [27]. The other effect that impacts the transistor length L is line edge roughness which reflects the difficulty of making the sides of the transistor channel completely smooth. This affects primarily the edge on the vertical side of wires because many nets are designed to be higher than they are wide, as a compromise between high integration (many transistors on a small area) and electrical resistance (which depends on the cut area of the net). The effect of line edge roughness is only correlated within a distance of less than 90nm [28] which means that it is unlikely that two adjacent transistors will be affected in the same way by line edge roughness. The study in [23] observed that the magnitude of line edge roughness variation is in the order of 5nm, and this magnitude remains independent of the scale of the manufactured transistors. There is a lack of published material on the mechanisms and correlations of TOX variation, but it is understandable that something that is so thin, around ten atom-layers [19], is difficult to manufacture. Due to the very thin gate oxides in DSM designs, the discussions in this thesis regard the effect of TOX variation to be uncorrelated between adjacent transistors. This is based on the observation that it is unlikely for variations in terms of an atom-layer more or less, to be correlated. It should be noted that TOX affects the value of VT and the gate capacitance. Furthermore, a very thin gate oxide will lead to gate tunnelling leakage, with leakage currents from the source, channel and drain nodes of the transistor flowing to and from the gate node [29]. Gate tunnelling leakage is further discussed in Chapter 4.

Design methods to cope with process variation include [12,30,31,32]. In [12], the authors suggest to add an extra latch to flip-flops to detect timing failures. When timing failures are detected, the circuit would use this information to adjust the supply voltage and clock frequency. Gate sizing is proposed in [31] to ensure confidence in the speed of a circuit under process variation. Another approach to cope with process variation is to use adaptive body bias and supply voltage scaling to adjust for delay and leakage caused by process variation [30].

1.4 Manufacturing Testing of Integrated Circuits

This thesis studies the impact of variation, in terms of supply voltage and process variation, on manufacturing testing. This section provides an introduction to manufacturing testing and the relevant concepts that are used later in the thesis.

The manufacturing process of integrated circuits is highly complex. Due to the com-

plexity of the manufacturing process, defects will be present and not all die on a wafer will operate correctly. Foreign particles and imperfections in the silicon wafer or the processing steps may result in bridged connections or missing features [33]. Such fabrication defects result from an imperfect manufacturing process [34]. Manufacturing tests are used after the IC is manufactured to verify that every gate and register in the IC are operational and have not been compromised by manufacturing defects. It is the aim of manufacturing testing to determine which die are good and should be used in end systems. There are certain types of defects that often occur in modern ICs. These include bridges, unintended connections between two or more circuit nodes, and opens, breaks between circuit nodes that were intended to be connected. More defect types are listed in Section 2.3. The main focus of this thesis is on bridge defects and open defects.

As circuit design become more densely integrated and therefore more complicated, more efforts in terms of testing are needed in order to maintain a high test quality. The invention of the integrated circuit (IC) in 1958 made it possible to integrate several logic functions onto the same piece of silicon. The integrated circuit meant that the basic components of a circuit could no longer be tested independently before the circuit was constructed, because all components are manufactured in the same process as the circuit itself. From the above it can be seen that integration leads to a requirement to test a large circuit rather than many small circuits.

Testing an IC involves applying stimulus to the inputs of the IC and observing the test response at the outputs. For digital circuits, which are the focus of this thesis, the stimulus consists of a vector of Logic-1's and Logic-0's such that each element of the vector corresponds to an IC input. The input assignment vector, together with the corresponding vector of expected test response for the IC outputs, is called a test pattern. A set of test patterns is called a test set.

The following sections give an overview of a range of concepts in manufacturing testing, including how defects are modelled using fault models (Section 1.4.2) which aid the evaluation (Section 1.4.3) and generation (Section 1.4.4) of test patterns. Furthermore, hardware added to a design to aid testing (Section 1.4.5) is discussed along with test cost (Section 1.4.6) and different types of tests and test methods (Section 1.4.7).

1.4.1 Defect Coverage by Fault Detection

Manufacturing defects result from an imperfect manufacturing process [34]. Typical defects include bridges (unintended connections between two or more circuit nodes) and opens (breaks between circuit nodes that were intended to be connected) [33]. The defects that are of interest to manufacturing testing are those that cause malfunctions. Defects can change the logic behaviour, increase delay (i.e. reduce circuit performance) or increase the power consumption of the circuit. In this thesis the focus is on the defects

that change the logic behaviour. A defect is covered by a test if the test detects the faulty logic behaviour which is caused by the defect. Faulty logic behaviour is detected by a test pattern if the test response seen at the output of the circuit that contains the defect is different from test response of a fault-free circuit. The test pattern to detect a defect can be designed if the faulty behaviour for the defect is known. Therefore, the behaviour of defects is described in fault models.

1.4.2 Fault Modelling

To design tests that identify defective ICs, fault models are developed and employed to predict how faults occur and their impact on circuits. A fault model is a formal description of how a defect alters the behaviour of a design. Using the fault models, test patterns can be designed to detect the impact of the defect, i.e. the faulty behaviour, by controlling nets so that the defect is activated and the faulty behaviour occurs while observing the behaviour on other nets. In this context, it is useful to define what is meant by controlling and observing. A net is controllable if an assignment to the IC inputs exists such that the net is set to the desired logic value. A net is observable if an assignment to the IC inputs exists such that a propagation path from the net to a primary output is created, so that the logic value on at least one primary output depends on the logic value on the net-under-observation. It should be noted that a test pattern needs to both control and observe nets in order to detect faults, which can lead to contradictions in terms of the assignment to IC inputs. If no test pattern exists which detects a logic fault, that logic fault is called undetectable.

A fault model typically specifies the faulty behaviour that can occur and where such behaviour can occur. That means that the fault model identifies the possible fault locations and therefore also the number of possible faults in a given circuit. For example, open defects can occur on any net. Therefore all nets are possible fault locations. When the possible fault locations are known, it is possible to evaluate the quality of a given test through a fault model-specific metric called fault coverage. The typical definition of fault coverage is the ratio of the number of faults detected to the total number of considered faults, usually given as a percentage. Full fault coverage, 100%, means that all possible fault locations that are specified by the fault model are tested.

The Stuck-At model is an example of a fault model [35, 36, 34]. With this model, a net can have two faulty behaviours, either stuck-at-0 or stuck-at-1. Therefore, each net is a possible fault location. To achieve full fault coverage according to the Stuck-At fault model, the test should detect both behaviours on each net. To detect a stuck-at-0 on net n, the test must control net n to Logic-1 and simultaneously observe net n. Detailed fault models will be further discussed regarding bridging faults and opens in Section 1.4.8 and Section 1.4.9 respectively.

There are many types of possible defects in modern ICs and some of them are reviewed in Section 2.3. Different defect types impact ICs to produce different faulty behaviours. To detect many defects and to simplify test generation, many fault models are abstract, as in the case of the stuck-at model. This means that tests generated with such fault models cover defects with modelled behaviour but fail to cover defects that cause unmodelled behaviour. Because of the many types of possible defects and the abstract nature of fault models, some defects will not be covered by, for example, a test with full stuck-at fault coverage. Defects that the test fails to cover are called test escapes [37, 38, 39, 40]. To reduce test escape, different approaches exist, including the employment of a combination of more detailed fault models, application of a large (near exhaustive) test set, and N-detection. N-detection testing applies a number N of different test patterns to each net-under-test or gate-under-test, to increase the probability of detecting the faulty behaviour of unmodelled defects [41, 42, 43].

1.4.3 Fault Simulation

The fault coverage of a test set T is measured by means of fault simulation. Each test pattern $V \in T$ is simulated for each considered fault f to determine if f is detected by V. The number of detected logic faults is then compared with the total number of considered faults as described in Section 1.4.2. The fault simulation procedure simulates two circuits, D and Df, using the stimulus of test pattern V. Here, D is the intended design and Df is the same design modified by the fault f. The simulation of the two circuits produces two test responses $D(V)$ and $Df(V)$. If there is a discrepancy in the test responses so that $D(V) \neq Df(V)$, the fault f is detected by the test pattern V. A detailed discussion on how fault simulation can be performed in a software tool is given in Section 3.2.3, where such a tool is used for resistive bridging faults. The fault simulation software is adapted for use on full open defects in Section 4.4.3 and for considering process variation in Section 5.3.3.

1.4.4 Test Generation

So far in this chapter, the discussion has introduced the concepts of test patterns, how test patterns are applied, how they detect faults and how they can be evaluated using fault simulation. Next, consider how test patterns are generated.

Test generation is the process of generating test patterns that when applied detect faulty circuit behaviour caused by defects [34]. Test generation is a hard problem considering that the numbers of test patterns should be kept low while achieving high fault coverage. Another aspect to the test generation problem is that some gates of a design are only accessible through other gates and a fair amount of computation is required to determine how such gates can be controlled and observed. In fact, the problem of proving that a

fault is undetectable, or providing a test pattern for it, is an NP-complete problem. This means that for worst-case problems the computation time is exponential in the number of inputs to the design. For large designs it can take a prohibitively long time to perform test generation for all faults and the test generator may have to give up on some faults. There are efficient algorithms for automatic test pattern generation (ATPG) [34]. These algorithms often focus on particular fault models such as the stuck-at fault model.

In this thesis, the concept of an ATPG-engine is used to refer to an algorithm that together with a fault model forms an ATPG tool. The main task for such an algorithm is to determine the assignments to the primary inputs of the circuit, that sensitises a path that contains the targeted defect location. The path is used to propagate the faulty signal from the defect location to a primary output. A path is called sensitised if it is composed of lines that for a test t change value in the presence of a fault f. Such lines are sensitised to the fault f by the test t [34]. An example of an ATPG-engine for combinatorial designs is the D-algorithm, which uses a particular algebra called D-notation for the generation of test patterns [34]. The computation time of the D-algorithm is exponential in the number of circuit nodes. Therefore, several algorithms improve on on the basic D-algorithm. Two such improved ATPG algorithms include PODEM and FAN [34]. These algorithms employ observations about the circuit structure to reduce the number of test generation problems that lead to exponential computation time and to make the computation time dependent on the number of primary inputs rather than on the number of circuit nodes. Another type of ATPG-engine, which is used in this thesis, is based on the Boolean Satisfiability problem, which is the problem of determining if there exists an assignment to the variables of a Boolean formula that makes the formula evaluate to **true**. There are solvers available, such as [44], that solve the Boolean Satisfiability problem by finding such an assignment to the variables. An ATPG-engine can be built using such a solver, by generating the Boolean formula that is equivalent of comparing the outputs of two circuits D and Df, where D corresponds to the fault-free circuit and Df corresponds to the same circuit modified by a fault f. The Boolean function should be defined so that it results in **true** if D and Df produce different results for the same input assignment. The input assignment generated by the solver is the stimulus vector of a test pattern for the considered fault. The solver in [44] is complete in the sense that it will find a test pattern if one exists. Therefore, if the solver fails to find a test pattern, the problem is unsatisfiable and the considered fault is undetectable. For more details on how to implement an ATPG-engine using this type of solver, see Appendix B. This type of ATPG-engine is used for resistive bridging faults in Section 3.2.4 and Section 5.5. Fault simulation plays an important role in test generation. Many test generation methods use a fault simulator to evaluate a proposed test. Based on the evaluation, additional test patterns are generated until the fault simulation shows that satisfactory fault coverage is obtained.

1.4.5 Scan Testing

The testing that is considered in this thesis employs structural information about the considered design to reason about the location of faults and how faults should be detected. To provide controllability and observability structural tests employ additional hardware that connect the flip-flops (registers) of sequential circuits using multiplexers, making the collection of flip-flops that are connected in this way into shift registers called scan-chains [34]. The flip-flops operate as scan-chains during test scan mode and operate as normal flip-flops otherwise. This modifies the test application procedure so that the stimuli are not applied just at the input pins of the IC but also at the outputs of the flip-flops. Similarly, test responses are not captured just at the output pins of the IC but also at the inputs of the flip-flops. The test application procedure using scan-chains is called scan-based testing and is illustrated in Figure 1.4.

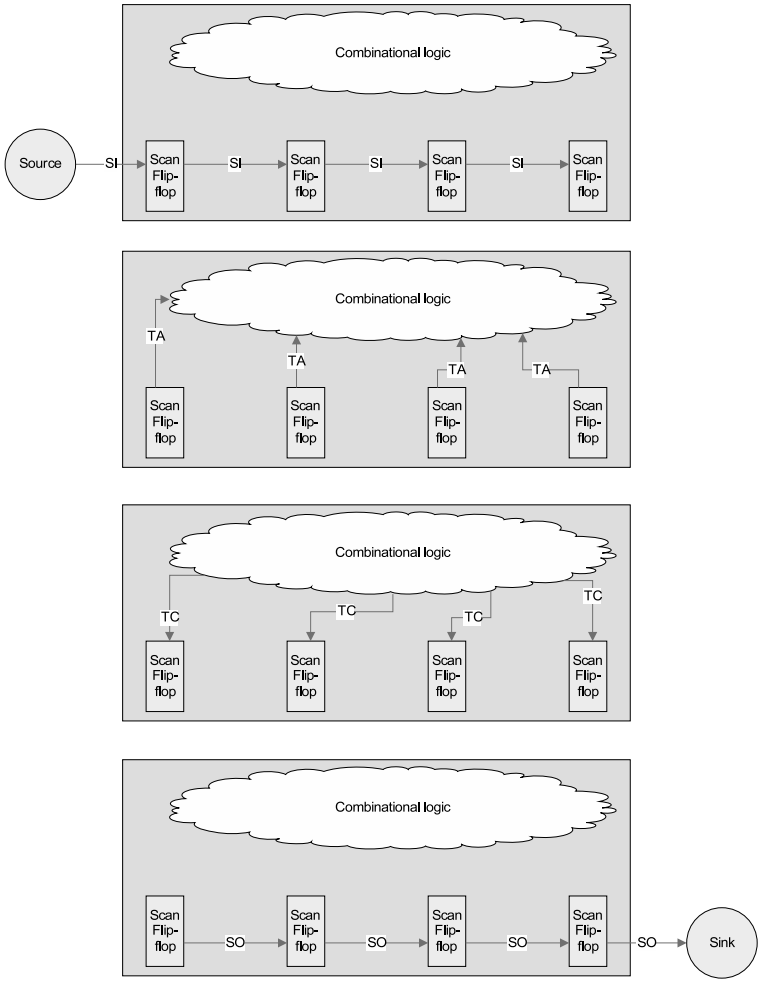


FIGURE 1.4: Test application procedure for scan-based testing

Figure 1.4 shows the procedure for testing using scan chains. The scan chain is used to shift the stimuli from a source, through the IC input pins into the circuit (as marked with SI, scan-in, in the top of Figure 1.4). The scan-in operation is followed by a clock

cycle of test application when the stimuli are applied at the output of the flip-flops (as marked with TA, test application, in the second part of Figure 1.4). During this clock cycle, the stimuli exercise the combinatorial circuitry between the flip-flops and produce test responses. The test responses are captured at the input of the flip-flops (as marked with TC, test response capture, in the third part of Figure 1.4) at the end of the clock cycle. Subsequently, the test response data is shifted out of the circuit by the scan chain to the IC output pins. From there the test response is transported to a sink (as marked with SO, scan-out, in the bottom of Figure 1.4). In this context, a source is where stimuli come from and a sink is where test responses go to be evaluated. The source and the sink can be implemented by automatic test equipment (ATE) or through extra hardware on-chip, called Built-in-Self-Test (BIST). An ATE has fast memories used for shifting test data to and from tested circuits and interfaces with tested circuits using probes and test ports. The BIST solution typically consists of memories and shift-registers that are designed to produce pseudo-random test stimuli sequences. Both ATE and BIST can serve as source for test stimulus and sink for test responses.

With regard to the design of scan-chains, there is a standard called IEEE 1149.1 [45] which specifies the interface used to communicate test data and circuitry to assist in testing integrated circuits in the context of chips assembled on printed circuit boards.

The concept of adding circuitry to a design to enable test to be conducted, as in the case of the scan-chains and BIST mentioned above, is called design-for-test (DfT).

1.4.6 Test Cost

As was mentioned above, the number of test patterns should not be too high, and the test quality should be high. Both these factors can be expressed in terms of test cost, which includes the test application time (a function of the number of test patterns), and the cost of return of defective products (a function of the ability of the test patterns to find defects). Further terms in the test cost are the cost of chip area for DfT, extra circuit delay due to DfT, the cost of designing DfT and test patterns, and the cost of renting or buying automatic test equipment (ATE). In high volume manufacturing testing, any slight reduction of the test time of an IC can lead to substantial savings.

1.4.7 Logic Testing, Delay Fault Testing and Current-Based Testing

Manufacturing testing of ICs can be divided into logic testing, delay fault testing and current-based testing. Logic testing is the type of testing considered in this thesis, but some of the relevant related research discussed in Chapter 2 addresses delay fault testing and current-based testing so all three types are reviewed in this section.

- Logic testing targets defects that cause time-independent logic malfunction. Operating scan-chains is the normal way of applying logic tests and tests generated according to the stuck-at fault model belong to this category.

- Delay fault testing targets defects that cause additional circuit delay so that the circuit does not meet its performance requirements (e.g. a specified clock period). The delay is the time it takes for the circuit to go from one state to another. A change of input data, for example the application of a test pattern, will cause a change on output data but not momentarily. The transition is associated with a delay. The purpose of delay fault testing is to detect defects that causes the transition delay to surpass the required clock period for the desired performance. Delay fault testing requires two test vectors, one that initialises the circuit and one that causes a transition in logic state. The time for the transition to pass through the circuit is compared to the clock period by capturing the logic value at the outputs into the scan flip-flops after a time corresponding to the clock period has passed. Due to the fact that the scan-chains form simple shift registers, there are two techniques available for applying delay fault testing. The first technique is called launch-on-capture [46], which means that the second test pattern is the result of applying the first test pattern for one clock cycle. The clock cycle of the first test pattern ends with the capture of the second test pattern in the flip-flops. The next clock cycle applies the second test pattern, effectively causing a transition of logic state. The test response is captured at the end of this clock cycle. The second technique for application of delay fault test patterns is called launch-on-shift [47], which means that the second test pattern is the result of shifting the first test pattern one step in the scan-chain. Similar to the first technique, a clock cycle is applied with the first test pattern, which ends with a shift of the scan-chain which applies the second test pattern for the next clock cycle. At the end of this clock cycle, the test responses are captured in the scan-chain flip-flops. Both techniques (launch-on-capture and launch-on-shift) are associated with constraints regarding what test patterns that can be applied and consequently limited in terms of the circuitry and the defects that can be tested.

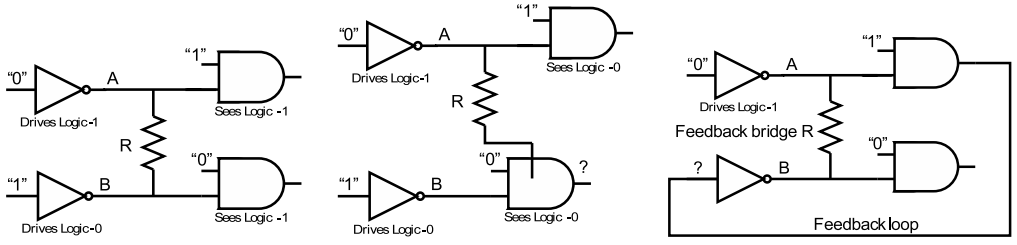
- Current-based testing, targets defects that cause the supply current profile of the tested circuit to change. For example, IDDQ testing [48, 49, 50, 51] measures the current when the circuit is in a stable state (i.e. no activity, quiescent mode). The two key observations behind this test technique is that (1) CMOS circuits ideally do not conduct any current while in a stable state and (2) many defect types produce such a current that can be measured to detect the presence of such defects. However, physical CMOS circuits experience small leakage currents during the stable state and for designs with very small transistors (as in the case in deep submicron designs), very thin gate oxides/dielectrics (approaching 10 atom layers) and low supply voltage (<1V) lead to an increase of these leakage currents [38]. This makes IDDQ testing less effective in detecting defects. Methods

like Delta-IDDQ testing still make current-based testing possible in many DSM designs. Some design methods have been suggested for making IDDQ testing more effective for DSM designs, but these methods typically result either in increased area or reduced performance [49]. Applying test patterns for IDDQ measurements is typically slower than voltage-based testing (logic testing and delay fault testing) because the circuit needs time to settle to a stable state (no activity) before the current measurement can take place. Another example of current based testing is the energy consumption ratio test (ECR) [52, 53]. It measures two dynamic currents and compares them. If one of the currents is elevated due to a defect, that will be reflected in the ratio between the two currents. This ratio is compared with that of other tested ICs from the same design to evaluate the test result. If the ECR value for an IC is significantly different from that of the other ICs, the IC is called an outlier and is regarded as defective. Tests that rely on comparison with other tested ICs, as in the case of ECR, cannot be evaluated until the test has been performed on many other ICs, which delays the pass-or-fail decision for all the ICs compared to voltage-based testing, where the pass-or-fail decision can be taken as soon as a fault is detected. The ECR test method has been shown to tolerate process variations, which would otherwise limit the usability of current-based tests [52].

1.4.8 Testing for Bridge Defects

This thesis considers two important defect types, resistive bridges and full opens. This section gives some background on the behaviour of bridge defects. Shorts and bridges are defects that connect two or more nodes that are not designed to be connected. In a short defect, at least one of the nodes is a power rail, either supply voltage or ground. In a bridge, none of the nodes connected by the defect are power rails. Bridges can form between two signal nets and are then called inter-gate bridges (Figure 1.5(a)), or they can form between internal nodes of gates and other nets and are then called intra-gate bridges [54, 55] (Figure 1.5(b)). Inter-gate bridges behave primarily as static defects. Intra-gate bridges on the other hand have primarily dynamic behaviour, but also pattern dependence, i.e. the faulty behaviour in a given gate does not depend only on the logic assignment to the inputs of that gate, but also on other signals in the circuit [54] and subsequently the choice of test pattern is very important. Inter-gate bridges that connect a net A to another net B so that a feedback loop is created, are called feedback bridges [56, 57, 58, 59, 60, 55, 61] (Figure 1.5(c)). This thesis focuses on non-feedback and inter-gate bridges, i.e. the defects that connect two signal nets without feedback. It should be noted that 30-40% of all bridge defects are feedback bridges [62] and that only up to 11% of all bridges are intra-gate bridges (based on the results presented in [63]). Even though feedback and intra-gate bridges are excluded from the study in the thesis, non-feedback and inter-gate bridges, which are studied, correspond

to a significant number of defects. In this thesis, if not stated otherwise, a bridge refers to an inter-gate, non-feedback bridge. If the defect has a resistance higher than 0Ω, it is called a resistive bridge. Measurements on bridge defects found in IC manufacturing testing [64] have showed that most bridges have a resistance $< 500\Omega$, while some bridges have been found with up to $20k\Omega$. The defect leads to altered circuit behaviour when the two nets are driven towards opposite logic values. Figure 1.5(a) shows an example circuit with a bridge defect between net A and net B. In the example, net A is driven to Logic-1 and net B is driven to Logic-0. Because of the defect, the voltages on net A and net B will depend on the relative strength of the gates that drive net A and net B in terms of driving current. These gate drive strengths depend on the input assignments to the gates. If the defect is a resistive bridge, there will be different voltage values on the bridged nets, which will depend on the bridge resistance R. The voltage expected on a net that is driven to Logic-0 is 0V and the voltage expected on a net that is driven to Logic-1 is the supply voltage, but the voltages on the bridged nets will be in-between 0V and supply voltage. The bridged nets each drive one or more gate inputs. The logic behaviour of a bridge defect depends on how the voltage on the bridged nets is translated into logic values by the inputs of the driven gates. The voltage on a gate input translate into a voltage on the output of the driven gate, which depends on the analog input-to-output function of the gate. Even if the input voltage to a gate is not a clear logic value because of a defect, the output of the gate is typically restored to a clear logic value. Therefore, it can be seen that the gate input translates the input voltage to a logic value. Often the analog input-to-output function is simplified by assuming a fixed logic threshold voltage, such that input voltage above the threshold is seen as Logic-1 and otherwise as Logic-0. In the example of Figure 1.5(a), net B is influenced by net A through the bridge defect so that the input that is driven by net B sees a Logic-1 instead of a Logic-0. This is a faulty behaviour due to the bridge defect.



(a) Inter-gate bridge defect location (b) Intra-gate bridge defect location (c) Feedback bridge location

FIGURE 1.5: Bridge defect types

The likely bridge defect locations are pairs of nets that are next to each other in the circuit layout. Such locations can be identified using Weighted Critical Area calculation, which considers the geometry data of the physical layout of the design. Weighted Critical Area calculations determine the nets that are close to each other [65]. Another method for identifying likely bridge defect locations is through extraction of coupling capacitance

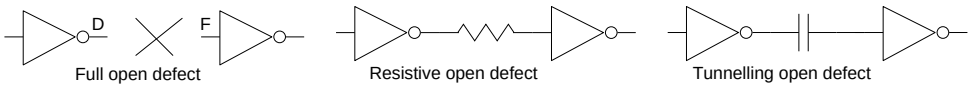
between two nets from physical layout. In the case of coupling capacitance extraction, a capacitance value above a given value (say $0.1fF$) would mean that the nets are close to each other and therefore it is possible that a bridge could occur between the two nets.

Bridge defects can also cause additional delay. These bridges typically have a bridge resistance that is outside the range that can be detected by logic testing. An analysis of delay fault testing targeting resistive bridges was performed in [66,67]. Other studies have recommended IDDQ testing for bridging faults [48,68], because the active bridge allows a current to flow from the supply voltage rail, through the gate that is driving high, through the defect and through the gate that is driving low to the ground rail. This current could be detected by IDDQ testing to determine the presence of a bridge defect. The research presented in this thesis focus on detecting the static behaviour caused by bridge defects using logic testing, because logic testing is typically part of most test solutions. Furthermore, logic testing is effective also when the effectiveness of IDDQ testing in detecting defects is reduced because of large leakage currents (Section 1.4.7). A full discussion on the behaviour of resistive bridges is given in Chapter 3 where a multi-voltage test generation method is presented for resistive bridges.

1.4.9 Testing for Open Defects

Open defects can be categorised into full opens (Figure 1.6(a)) and resistive opens (Figure 1.6(b)). A full open is a complete break that separates two wires that should have been connected (Figure 1.6), while a resistive open adds a resistance between two wires that should be connected but without that extra resistance. To see evidence that full open defects occur in practise, consider the study in [69], where the resistance distribution for opens on metal wires, vias and contacts was determined. It was shown that over 40% of all opens have >1GΩ resistance, even 60% for metal wires. Such high resistances must be considered as full opens. Furthermore, complete breaks have been found in diagnosis of faulty chips [37,70,71,72]. Full opens are as important to consider as resistive opens according to [37]. Full opens and resistive opens can take place anywhere in the circuit, on interconnect (inter-gate) or within gates (intra-gate). Full open defects on interconnect should be discussed apart from resistive opens and intra-gate full opens, because a full open on interconnect have a static behaviour [70] whereas resistive opens have dynamic behaviour (RC delay) [73,74] and intra-gate full opens also cause dynamic behaviour. Intra-gate opens have been studied in [75,76,77,78,71,79], and it has been reported that such defects can cause delay behaviour, can increase IDDQ and can cause static faulty logic behaviour [77]. The focus in Chapter 4 is on full open defects on interconnect, since most open defects occur on interconnect [80]. The considered defect type in Chapter 4 completely separates a net from its driver (Figure 1.6(a)) where net F is separated from the driver D. This is unlike tunnelling opens [81] (Figure 1.6(c)) which have a capacitive coupling between the separated net and the driver. Furthermore,

tunnelling currents can cross the break in tunnelling opens.



(a) Full open (b) Resistive open (c) Tunnelling open

FIGURE 1.6: Open defect

There are two varieties of full open defects on interconnect to consider, with and without the influence of gate tunnelling leakage, and both of them manifest static behaviour. If a full open defect is not influenced by gate tunnelling leakage, the voltage on the net that is separated from its driver (the victim net F in Figure 1.6(a)) will depend on trapped charge and capacitive coupling to nearby circuitry, namely neighbouring nets and nodes of driven gates [82]. In the complementary case, if a full open is influenced by gate tunnelling leakage, transistor gates connected to the victim net will leak charge to or from the victim net and find an equilibrium when the same amount of charge enter and leave the victim net, which determines the victim net voltage [83, 84, 29]. The logic behaviour of full open defects on interconnect are given by how the gate inputs that are driven by the victim net interpret the victim net voltage. If the victim net is below the logic threshold voltage (a simplification of the analog input-to-output function of a CMOS gate), the input sees a Logic-0 and otherwise a Logic-1. The behaviour of full opens is complicated because of the fact that gate inputs that are driven by the victim net can interpret the victim net voltage to different logic values depending on the logic threshold voltage of the input. This phenomenon occurs when the victim net voltage is about half of the supply voltage (half-swing) and is known as the "Byzantine Generals problem" because of its similarities with a classical computer-science problem [85]. A full discussion on testing for full opens is given in Chapter 4, where the supply voltage dependent detectability of full opens is investigated.

1.4.10 Testing Low-Power ICs

As this thesis addresses testing for designs that use multiple supply voltages and most such designs are low-power designs, it is relevant to discuss other concerns in terms of testing low-power designs. Testing low-power designs has attracted a fair amount of research, but the problem of testing designs with multiple supply voltages has not been explicitly addressed.

The main problem associated with testing low-power designs arise from the fact that the power supply rails of a low power design are often scaled with regard to the demands of the functional mode of the design without regard to the power supply demands during test application. It may be found that power supply rails designed in this way are insufficient to supply enough current during test, as testing tends to cause a higher

switching activity [86, 87] than during normal operation. One of the reasons for high switching activity is that a lot of switching occurs in scan-based testing during the scan operation. The extra power consumption can cause the circuit to be less reliable or, in some cases, can provoke instant circuit damage. The risk of damage is related to over-heating. The problem of testing low-power designs to avoid over-heating is called power constrained testing and methods that have been employed and suggested [88] include:

- Sizing the power supply, packaging and cooling according to test requirements rather than to the requirements of the IC operation in functional mode.
- Testing at a reduced speed so that the heat that is generated by testing has time to dissipate and so avoid over-heating.
- Partitioning the system-under-test and schedule the tests for the partitions with power constraints.
- In test generation, fill don't-care positions in test patterns such that the switching activity during testing is reduced. Don't care positions in test patterns are arbitrary assignments to those IC inputs that will have no impact on detecting the faults that the test pattern was generated for.
- Test pattern re-ordering to reduce switching activity.
- Modifying the scan-chains to form shift-and-update registers so that the circuit-under-test will have constant inputs during the scan operation. This involves adding extra latches to each flip-flop, which may prove expensive in terms of silicon area.
- Test scheduling on module-based SoCs so that heat accumulation local to a set of modules is considered and over-heating avoided [89].
- Perform part of the testing using built-in-self-test (BIST) at a time separate from other testing so that the heat that is generated by testing has time to dissipate and so avoid over-heating.
- Reordering the flip-flops in the scan-chain to reduce switching activity [90].
- Performing the test at a supply voltage and clock frequency setting that minimises power consumption [91]

In this thesis, the problem of power constrained testing (which is primarily concerned with reducing switching activity during test) is considered orthogonal to the problem of testing in the presence of variation, i.e. the problems can be solved independently and the solutions can be applied together. Therefore, power constrained testing is not further discussed in this thesis.

1.5 Thesis Outline and Scope

This thesis is focused on logic testing of static CMOS circuits of resistive bridge defects and full open defects. The study is relevant because the considered defect types are among the most important ones in modern ICs. Furthermore, static CMOS is the dominant technology for modern IC designs and logic testing is typically included in every test solution. In this context, the purpose of the research in this thesis is to gain better understanding of how process and supply voltage variations impact the detectability of defects in deep submicron designs. Furthermore, this thesis will investigate how increased knowledge on the impact of such variation can be used to develop low-cost and effective test solutions for designs that use more than one supply voltage and designs that are sensitive to process variation. To conduct this type of study, the work presented in this thesis also involves developing software tools to enable experimentation and analysis.

The thesis is structured as follows:

- Chapter 2 - Literature Review and Motivation
  The state-of-the-art in research relevant to the problems addressed in this thesis is reviewed. This involves fault modelling for bridge defects and open defects, testing in the presence of supply voltage variation and testing in the presence of process variation. The literature on supply voltage-dependent defect behaviour is summarised and relevant research in testing using other-than-nominal supply voltage settings are reviewed.

- Chapter 3 - Analysis of Testing for Resistive Bridges under Supply Voltage Variation
  The supply voltage-dependent behaviour of resistive bridge defects is studied using three supply voltage settings and a 0.12μm CMOS technology and it is observed that more bridge resistance is exposed for lower supply voltage levels. Bridge defects that require specific supply voltage settings for detection are analysed using a bridge fault simulator that is developed for the specific purpose of the study. The findings are employed in a test generation algorithm which produces supply voltage specific test sets that together achieves full defect coverage. This means that testing should be performed using more than one supply voltage to detect all defects. Experimental results on synthesised benchmark circuits and realistic bridge locations show that full defect coverage can be achieved for resistive bridging faults over the entire set of considered supply voltage settings. The results validate the proposed method and the concept of generating supply voltage-specific test sets.

- Chapter 4 - Analysis of Testing for Full Open Defects under Supply Voltage Variation

The mechanisms behind supply voltage-dependent behaviour of full opens are investigated to find how such behaviour can lead to supply voltage-dependent detectability of full opens. Two complementary mechanisms are found, one is related to gate tunnelling leakage and the other depends on capacitive coupling. A simulation tool is developed to aid the study by providing quantitative results. The analysis shows that full open defects cause supply voltage-dependent behaviour but also that full open defects are detectable independent of the supply voltage with few exceptions. The analysis is supported by extensive simulation results on synthesised benchmark circuits.

- Chapter 5 - Process Variation-Aware Testing for Resistive Bridge Defects
  The impact of within-die process variation on test quality is studied in the context of resistive bridging faults. The study shows that the logic behaviour of bridge defects does not only depend on the defect resistance and the supply voltage, as discussed in Chapter 3, but also on two parameters that are influenced by process variation. These parameters are logic threshold voltage and gate drive strength. It is observed that a test generated with a state-of-the-art method, that achieves full defect coverage on a circuit which has all parameters on nominal values, can fail to cover some defects in the presence of process variation. A metric called test robustness is presented to quantify the impact of process variation on the quality of a test. The metric is used to guide a novel process variation aware test generation method, which can achieve a user-specified test robustness target using a small number of test patterns. Experimental results on synthesised benchmark circuits and realistic bridge locations show the impact of process variation and the benefits of the proposed test generation method.

- Chapter 6 - Conclusion
  A concluding discussion summarises the contributions achieved by the research described in this thesis and future work is described.

1.6 List of Publications from this Ph.D. Project

The following peer-reviewed papers have come out of the Ph.D. project.

- Resistive bridging faults DFT with adaptive power management awareness
  Ingelsson, U., Rosinger, P., Khursheed, S. S., Al-Hashimi, B. M., and Harrod, P.
  in Proceedings of the IEEE Asian Test Symposium, October 2007, pages 101-106
  This paper contains the supply voltage-aware test generation method for resistive bridging faults as is represented in Chapter 3. The paper also contains work by S. S. Khursheed (another Ph.D. student in the same research group as the author) on test point insertion to reduce the number of supply voltage used in testing, which is not part of this Ph.D. project and subsequently not discussed in this thesis.

- Bridging fault test method with adaptive power management awareness
  in IEEE Transactions on Computer-Aided Design of Integrated Circuits, 2008
  This paper extends the research presented in the paper mentioned above with more results and a post-processing step to the method to reduce the total test set size. With the presented method it is found that more than one supply voltage is required to achieve full bridge defect coverage. The paper also contains work by S. S. Khursheed on test point insertion to reduce the number of supply voltage settings used in testing, which is not part of this Ph.D. project and subsequently not discussed in this thesis.

- Variation aware analysis of bridging fault testing
  Ingelsson, U., Al-Hashimi, B. M. and Harrod, P.
  in Proceedings of the IEEE Asian Test Symposium, November 2008, pages 206-211
  This paper contains an analysis of the impact of process variation on testing for resistive bridging faults as is represented in Chapter 5 including the test robustness metric that quantifies the impact of process variation on test quality.

- Process variation-aware test for resistive bridges
  Ingelsson, U., Al-Hashimi, B. M., Khursheed, S., Reddy, S. M. and Harrod, P.
  in IEEE Transactions on Computer-Aided Design of Integrated Circuits, 2009 (accepted for publication)
  This paper presents the process variation-aware test generation method that is discussed in Chapter 5.

Software tools developed by the author of this thesis during the course of this Ph.D. project have proved useful in other research projects that were not conducted by the author of this thesis. This software implements the supply voltage-aware test generation method and the associated fault simulator for resistive bridging faults. The research projects that have benefited from this software [92, 93, 94, 95] are not further discussed in this thesis, but can be seen as continued work on problems related to those addressed in this thesis.

Chapter 2

Literature Review and Motivation

This chapter provides an overview of state-of-the-art research that is related to this thesis. The overview includes a discussion on fault models and test generation methods for the considered defects, namely resistive bridge defects (Section 2.1) and full open defects (Section 2.2). With regard to the focus of this thesis on studying the impact of supply voltage on manufacturing testing, Section 2.3 reviews the supply voltage-dependent behaviour of defects and test methods that use other-than-nominal supply voltage. Subsequently, the state-of-the-art is reviewed in testing ICs that are influenced by process variation (Section 2.4). Finally, this chapter includes motivation for the research in subsequent chapters in light of the reviewed research (Section 2.5).

2.1 Bridging Faults

Fault modelling (Section 1.4.2) for bridge defects has developed from simple and abstract to more complicated and detailed [96]. The bridging fault models can be categorised into three types, stuck-at related models, models that abstract from the bridge resistance and models that take the bridge resistance into account. The following sections discuss the three categories.

2.1.1 Stuck-at-Related Models

There are several different bridging fault models with varying level of abstraction. Some studies have targeted bridging faults without an actual bridging fault model. Such studies include modelling the bridging fault with multi-line stuck-at faults [97], N-detect testing using stuck-at fault test patterns [98, 42], and considering signal probabilities in test generation to increase the probability of activating bridging faults [42].

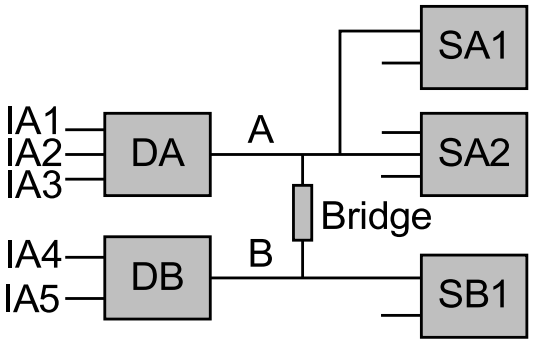


FIGURE 2.1: Circuit where nets A and B are bridged

2.1.2 Models that Abstract From the Bridge Resistance

All actual bridging fault models have the activation criterion in common, that to activate a bridge defect, it is required that the bridged nets are driven to opposite logic values. Figure 2.1 shows a circuit with two bridged nets, A and B, which are driven by the gates DA and DB. The bridged nets drive the successor gates SA1, SA2 and SB1. In a bridging fault model that takes process variation into account [99], there is no prediction of the defect behaviour, but instead the fault model considers all possible logic behaviours. All possible input assignments and logic behaviours at the bridge fault site (Appendix A) are considered, to cover all eventualities that arise from process variation. This means that all possible logic configurations are considered at the inputs that are driven by the bridged nets. If the bridged nets have a combined fanout of k inputs, the number of possible logic configurations is 2^k. The result is a large set of logic faults to process, which takes a long time. To compensate, the approach in [99] is abstract from IC parameters and bridge resistance, motivated by the need for fast fault simulation of the large set of logic faults. Most abstract bridging fault models describe the logic behaviour using simple Boolean functions, as in the wired model [100] which was developed for TTL technology, where the logic behaviour is determined by wired-OR or wired-AND. The wired model is not accurate for CMOS [100]. In the 4-way model [98, 42], the logic behaviour is determined by the gates that drive the bridged nets, such that one gate dominates the other gate and drives both the bridged nets to the same logic value. A net that is driven to the dominating value in this way is called the aggressor net and the other net is called the victim net, because the victim net changes its logic value due to the defect. Either of the two gates may be the one driving the aggressor net and while there are two logic values in digital circuits, four faults may occur for the same bridge location [98, 42]. If the gates that drive the bridged nets are called DA and DB, as in Figure 2.1, the four faults are: DA dominates DB with Logic-1, DA dominates DB with Logic-0, DB dominates DA with Logic-1 and DB dominates DA with Logic-0.

Instead of the 4-way model, further research [101,57] defined a primitive bridge function of the input assignment to the gates that drive the bridged nets. The generic primitive bridge function f for the circuit in Figure 2.1 is shown in Equation 2.1, where $L(A)$ is

the logic value on net A and $L(B)$ is the logic value for net B.

$$L(A) = L(B) = f(IA1, IA2, IA3, IA4, IA5) \tag{2.1}$$

The primitive bridge function calculates the logic value on the bridged nets and is defined by Spice-type simulations. The simulations are performed once for each pair of gates in the gate library. A similar model, called the voting model [102] uses information about the drive strengths of different gates to estimate the voltage on the bridged nets and determine the aggressor net. The voting is performed by considering the bridge location as a resistive divider with regard to the on-resistance of the gate that is driving high and the on-resistance over the gate that is driving low. The gate with the least on-resistance drives the aggressor net. In the voting model, it is taken into account that the drive strength of a gate depends on its input assignment. An improvement to this model, called the biased voting model [102], calculated the actual voltages on the bridged nets. Furthermore, the biased voting model improved on the above mentioned bridging fault models by considering the fact that the inputs each have a different logic threshold voltage, i.e. the voltage level on a gate input for which the corresponding gate output would change logic value. When the logic threshold for each gate input is slightly different, the interpretation of the voltages on the bridged nets into logic values varies between gate inputs. The previous models (the wired-AND/OR model, the 4-way model, the model with a primitive bridge function, the voting model) had considered that all inputs that are driven by the same net would see the same logic value, i.e. that all inputs would have the same logic threshold voltage.

2.1.3 Models that Consider the Bridge Resistance

The bridging fault models mentioned so far do not explicitly consider the bridge resistance. In a sense, these bridging fault models assume that the logic behaviour is the same for all bridge resistance values, or that the bridge has 0Ω resistance. Actually, each bridge location corresponds to a range of possible defects, each with different bridge resistance. The resistance of a bridge defect is fixed but of unknown value. A circuit where two nets, A and B, are bridged with a resistance R_{sh} is shown in Figure 2.2. The two nets A and B are driven by the gates DA and DB respectively and the nets drive the successor gates SA and SB.

Several studies have observed that the voltage on the bridged nets depends on the resistance [103, 104] as shown in Figure 2.3. For the example in Figure 2.2, the graphs in Figure 2.3 show how the voltages VA and VB on the bridged nets depend on the bridge resistance R_{sh}. In this example, net A is driven high and net B is driven low. For $R_{sh} = 0\Omega$, VA=VB and for higher values of R_{sh} the values for VA and VB diverge

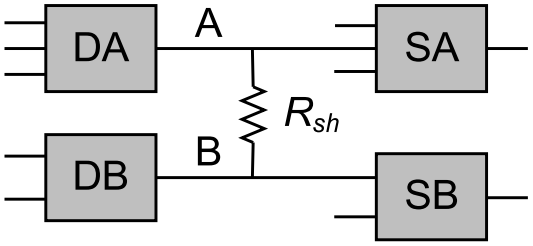


FIGURE 2.2: Example resistive bridge

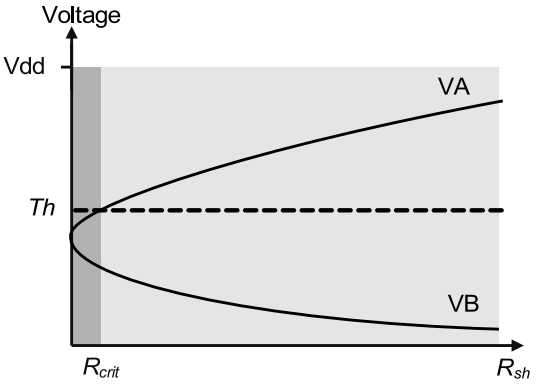


FIGURE 2.3: The voltage on the bridged nets as it depends on the bridge resistance

until for a sufficiently high R_{sh} value, VA reaches supply voltage level and VB reaches 0V.

A bridging fault model that takes the bridge resistance into account (i.e. a model for resistive bridging faults) associates a range of bridge resistances to each logic fault, as was considered in [103]. The fault model in [103] is called the parametric bridging fault model because it uses resistance as a parameter. The study in [103] was based on the assumption that all involved gates have the same logic threshold voltage Th, equal to half the voltage of Logic-1. With this assumption, there is only one resistance interval of faulty behaviour to consider, which is from $R_{sh} = 0\Omega$ to $R_{sh} = R_{crit}$, from the lowest to the highest resistance value that can cause malfunction. R_{crit} is called the critical resistance. This is illustrated in Figure 2.3 by the horizontal line marked Th. If the voltage on a gate input is below the logic threshold Th, this voltage is seen as Logic-0, otherwise as Logic-1.

A further development of the parametric bridging fault model was made in [104], where the logic threshold voltage was explicitly considered for each input that is driven by a bridged net. That means that there are more resistance intervals to consider, because there is a highest resistance value R_{crit} for each input, such that resistances above R_{crit} are not seen as malfunction for that input. Consider the two inputs that are driven by net A and B in Figure 2.2. The logic threshold voltages for these two inputs are shown in Figure 2.4, marked Th_{SA} and Th_{SB}. There are two critical resistances $R_{crit,A}$ and $R_{crit,B}$. In this scenario, there are three disjunct resistance intervals with different logic

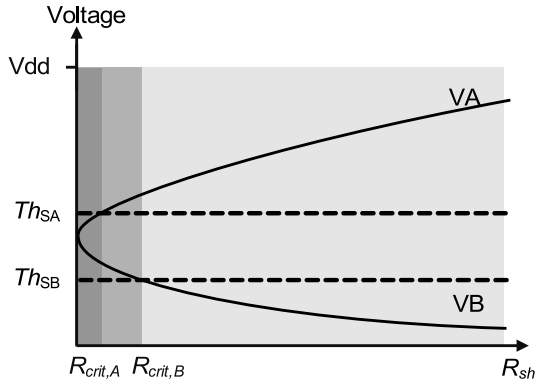


FIGURE 2.4: The resistance ranges corresponding to the logic threshold voltages Th_{SA} and Th_{SB}

behaviours. These intervals are $[0, R_{crit,A}]$, $[R_{crit,A}, R_{crit,B}]$ and $[R_{crit,B}, \infty]$. The first two intervals correspond to logic faults. The first logic fault, for the range $[0, R_{crit,A}]$, is visible as a faulty Logic-0 on input A and a faulty Logic-1 on input B. The second logic fault, for $[R_{crit,A}, R_{crit,B}]$ is visible only as a faulty Logic-1 on input B. The third interval corresponds to bridges with a resistance that cannot be detected by logic testing. A higher fanout on the bridged nets lead to a larger set of logic faults with corresponding bridge resistance intervals. The fact that each bridge location has a range of bridge resistance values has implications on the concepts of fault detection and fault coverage. From the above it can be seen that, each parametric bridging fault is associated with a resistance range that is covered if the fault is detected. This resistance range associated with a logic fault is called the Analog Detectability Interval, ADI [105], of the logic fault. A test that detects the logic fault covers the corresponding analog detectability interval. Therefore, the ADI becomes a part of the Covered Analog Detectability Interval of the test, the CADI [105]. A test with full parametric fault coverage covers the complete set of bridge resistances that can cause detectable malfunction. This set is called the Global Analog Detectability Interval, GADI. The parametric bridging fault coverage FC is defined in Equation 2.2 where b represents the bridge location and T represents the test.

$$FC(b,T) = \frac{\|CADI(b,T)\|}{\|GADI(b)\|} \tag{2.2}$$

$$FC(B,T) = \frac{\sum_{b \in B} FC(b,T)}{\|B\|} \tag{2.3}$$

For a circuit with a set of bridge locations B, the fault coverage is given in Equation 2.3. Further discussion of refinement of the parametric bridging fault model can be found in [38, 39, 105, 106, 96]. For example, the study in [39] discusses defects that the tests fail to cover (test escapes) by using the parametric bridging fault model and [96] discusses methods of estimating the current through the bridge resistance and the highest bridge

resistance value R_{crit} that can cause malfunction. The assumption of well defined logic threshold voltages for gate inputs often used in conjunction with the parametric model has been criticised in [38] because that assumption disregards process variation and noise. Instead, [38] suggests to consider voltage in $[0.9 \cdot Vdd, Vdd]$ and in $[0, 0.1 \cdot Vdd]$ reliably detected as Logic-1 and Logic-0 respectively. This thesis will use the simplification of a well defined logic threshold voltage in Chapter 3. Process variation and its influence on the behaviour of resistive bridge defects is considered in Chapter 5 where the influence of process variation on the logic threshold voltage is investigated using Monte-Carlo simulation (Section 5.2.1). The parametric bridging fault model has been used for bridging fault simulation [107,108], diagnosis [92], ATPG [109,106,110,111] and studies that involve testing using other-than-nominal supply voltage [112,113,109,114,115,116].

2.1.4 Test Generation and Fault Simulation for Resistive Bridging Faults

Recently, fault simulation and ATPG tools for the parametric bridging fault model have been proposed [109, 117, 106, 110, 111, 108]. These tools make use of interval algebra to represent the intervals of bridge defect resistance that cause malfunction. The study in [109] identified the logic fault that corresponds to the largest resistance interval for a given bridge location and determines the corresponding test pattern. In contrast to [109], the sectioning approach from [110] considers each input i that is driven by the bridged nets to have a critical resistance $R_{crit,i}$, which is the highest resistance that cause malfunction on input i, and these critical resistances form sections of disjunct resistance intervals. For example, input j has the critical resistance that is the next beyond that of input i, which is represented by the section $[R_{crit,i}, R_{crit,j}]$. For each section with faulty behaviour, the corresponding logic fault is identified. This transforms the problem of testing all defect resistance to targeting a set of logic faults and improves the test quality compared with [109], but the number of considered faults grows. In [111], the authors combine the advantages of the interval based [109] and the sectioning approach [110] into a more efficient test generation procedure by targeting the logic fault of the section with the highest resistance values first. Fault simulation is then used to identify all other sections covered by the test pattern and only not-yet-covered resistance intervals are considered in the subsequent processing. Targeting the sections with the highest resistances first increases the probability of finding a test pattern that covers the whole resistance range early, which would be revealed by the fault simulation. To enable fast fault simulation of resistive bridges, two studies have proposed methods with parallel evaluation of test patterns [117, 108]. The method in [108] is also capable of parallel evaluation of bridging faults, so that parallel simulation can be conducted either with regard to the test patterns or with regard to the faults. Furthermore, the fault simulation method [108] employs propagation of detectable resistance intervals, which is useful for the interval algebra mentioned above.

2.2 Full Open Faults

In terms of fault modelling for full open defects, it is important to consider the location of the defect and the factors that influence the defect behaviour. The following sections consider the defect location and two complementary mechanisms that determine the behaviour of full open defects. The first mechanism is gate tunnelling leakage, which is the phenomenon that charges can tunnel through a thin gate oxide, so that the gate of a transistor is not electrically isolated from the other nodes of the transistor. The other mechanism, that influences the behaviour of full open defects in absence of gate tunnelling leakage, is capacitive coupling to neighbouring nodes.

2.2.1 Full Open Defect Location

A vital component of the fault model for full opens is the location of the defect, which has been considered in [85,118,72]. A full open defect can occur anywhere along the nets of the design, within a logic gate or on the interconnect. The study in [85] addresses the fact that the behaviour of a full open defect on interconnect depends on the location on the net. For a net that has a fanout of two or more, this leads to a number of different possible configurations, which is illustrated by Figure 2.5. If the defect is at the driver (segment A), all fanouts (Input 1 and Input 2) should be affected by the defect. If the defect is on a branch, for example segment B, only fanouts on that sub-net will be affected, i.e. Input 1 would be affected. The behaviour of full open defects also depends on influence of neighbouring nets. A defect on segment C in Figure 2.5 can be influenced by the neighbouring net N, but a defect on segment D cannot be influenced in the same way. Thorough studies have shown that the probability of open defects is particularly high in vias and contacts [37, 69]. Therefore all possible vias and contacts should be tested for the possibility of a full open defect. Analysed from a different perspective, it can be seen that a fault model for full open defects should consider all vias and contacts as likely defect locations. Considering the possible locations for full open defects have proved useful in several diagnosis and fault analysis studies [85,119,118,72]. The defect coverage metric is typically defined as the ratio of detected faulty behaviours and the number of considered faulty behaviours. The set of considered faulty behaviours will depend on the assumed defect locations and logic configurations as seen by the inputs that are driven by the victim net.

2.2.2 Gate Tunnelling Leakage Model

The main difficulty in predicting the behaviour of full open defects is to determine the voltage on the victim net. When the victim net voltage is known, the logic behaviour is found by comparing the victim net voltage to the logic threshold voltages of the

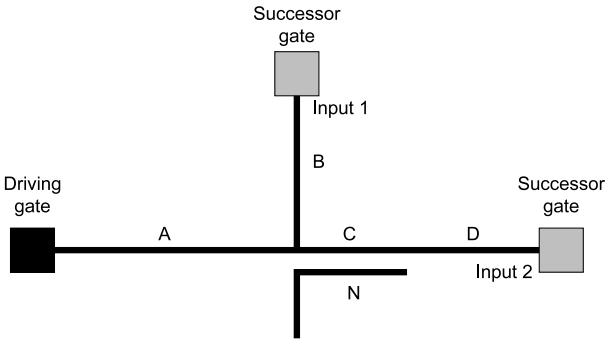


FIGURE 2.5: A net with several possible full open locations

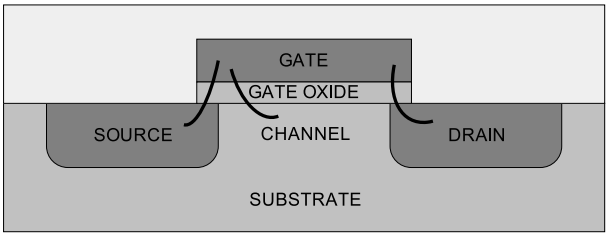
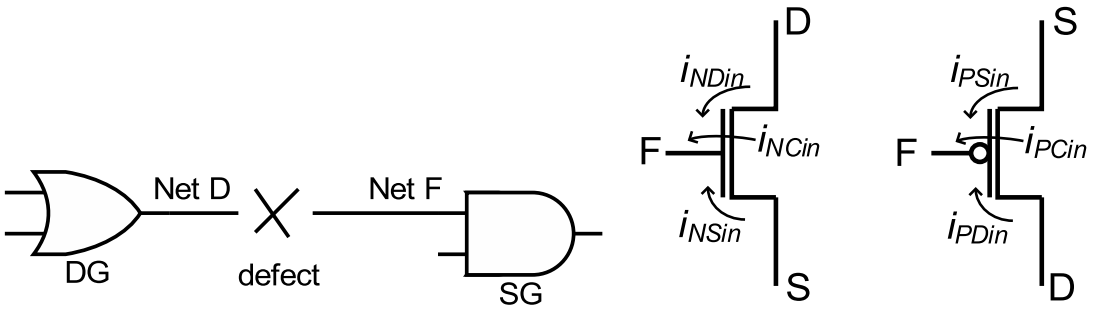


FIGURE 2.6: Gate tunnelling leakage current components

inputs that are driven by the victim net. There are two complementary mechanisms that control the voltage on the victim net. These are: gate tunnelling leakage [83,29,84] and in the absence of gate tunnelling leakage, capacitive coupling to neighbouring nets and nodes. This section discusses the mechanism of gate tunnelling leakage and the next section addresses how full open defects are modelled in terms of capacitive coupling.

In deep submicron designs, the gate oxide tends to be very thin (down to ten atom-layers [19]) for performance reasons, which means that it is possible for small currents to tunnel through the gate oxide, so that the transistor gate is not isolated from the other nodes of the transistor. Figure 2.6 shows the nodes of a transistor including source, drain, channel and gate. The marks in the figure shows the leakage current components to or from the source, drain and channel nodes through the gate oxide. Recent technologies reduce the capacity for gate tunnelling leakage by replacing the gate oxide by materials [120] with a high dielectric constant compared to the traditional silicon dioxide. With such materials, it is possible to make the gate dielectric thicker while maintaining the same circuit performance and a thicker dielectric mean less tunnelling leakage current.

Consider the scenario of a victim net that is influenced by gate tunnelling leakage. Figure 2.7 shows a full open defect which is influenced by gate tunnelling leakage. The leakage currents between the net F and the nodes of the driven NMOS and PMOS transistors of input i in gate SG are shown in Figure 2.7(b). The nodes are marked S for Source, D for Drain and C for Channel. The arrows in Figure 2.7(b) do not show the direction of the currents but rather the sign convention. A current that adds charge



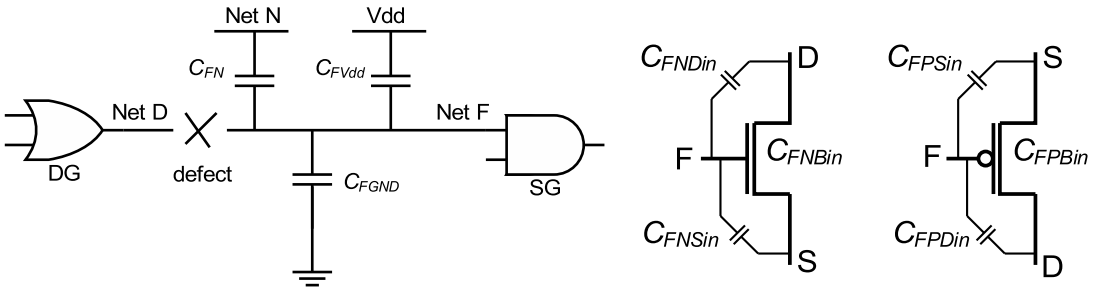
(a) Defect location

(b) Leakage currents to and from nodes of driven transistors

FIGURE 2.7: A full open defect which is influenced by gate tunnelling leakage

to net F is counted as positive and a current that removes charge from net F is counted as negative.

Very little fault modelling has been done in the context of full opens influenced by gate tunnelling leakage. A theoretical study [83] showed that the victim net voltage is determined by the tunnelling leakage currents to and from conducting NMOS and PMOS transistors for which the victim net is the gate. The victim net voltage is subject to a delay from any transition of logic state in the driven gates, because the leakage currents are relatively small and before the victim net voltage becomes stable, gate capacitances and coupling capacitances to other nets must be charged or discharged correspondingly. The study in [84] reported that this delay can be as long as a second, but more probably tens or hundreds of microseconds for future technologies. After the delay, the final static victim net voltage is that which causes the leakage current to the victim net to be equal to the leakage current from the net, which means that the total amount of charge on the net is constant. For an open defect that affects an inverter, it was reported that the voltage on the victim net will be so low that it will be seen as Logic-0 [83,84]. The study in [29] showed that the leakage currents themselves depend on the gate voltage, such that the victim net voltage can be bi-stable, depending on its original voltage before any transition in the circuit. If the victim net voltage was originally high, the final value can be higher than if the victim net voltage was originally low. All the presented studies [83, 84, 29] have employed Spice-type simulation to determine the behaviour of full open defects, which leads to accurate modelling of the behaviour at the cost of long computation times per full open defect. An observation was made in [84] which indicated that the victim net voltage was seen as Logic-0 when the defect affected a single inverter. The observation is useful, but does not lead to the general conclusion that all full open defects that are influenced by gate tunnelling leakage would behave the same way, for reasons that are further discussed in Section 4.3.3. From the above it can be seen that further research efforts are needed in fault modelling for full opens in the presence of gate tunnelling leakage.



(a) Defect location

(b) Coupling capacitances between net F and nodes of driven transistors

FIGURE 2.8: A full open defect influenced by capacitive coupling

2.2.3 Capacitive Coupling Model

In the alternative scenario, if there is no gate tunnelling leakage, the victim net is electrically isolated from the other nodes of the circuit. In this scenario, several studies [121, 82, 72, 122, 118] have modelled the victim net voltage as a function of trapped charge on the victim net, capacitive coupling to neighbouring nets and capacitive coupling to nodes of gates that are driven by the victim net. An example is shown in Figure 2.8, where net F is the victim net, separated from net D (the driver) by the defect. There are coupling capacitances to the supply voltage rail C_{FVdd} and ground C_{FGND}. Further, there is a neighbouring net N with the coupling capacitance C_{FN} between net F and net N. Figure 2.8(b) shows capacitive coupling between net F and nodes of NMOS and PMOS transistors belonging to input i of gate SG. The nodes are marked S for Source, D for Drain and B for Bulk.

The function for the victim net voltage is shown in Equation 2.4, where V_{victim} is the victim net voltage (the voltage on net F), $C_{nodes\ with\ Logic-1}$ is the total capacitance to neighbouring nodes that are at Logic-1 voltage (C_{FVdd}+C_{FPSin}+C_{FPBin} and potentially C_{FPDin}, C_{FNDin} and C_{FN} depending on their voltage), $C_{neighbouring\ nodes}$ is the total capacitance to neighbouring nodes ($C_{FN} + C_{FGND} + C_{FVdd} + C_{FNDin} + C_{FNSin} + C_{FNBin}$+$C_{FPDin}$+$C_{FPSin}$+$C_{FPBin}$), Vdd is the supply voltage, $Q_{trapped}$ is the amount of trapped charge on the victim net and $C_{victim\ to\ GND}$ is the capacitance between the victim net and ground (C_{FGND}).

$$V_{victim} = \frac{C_{nodes\ with\ Logic-1}}{C_{neighbouring\ nodes}} \cdot Vdd + \frac{Q_{trapped}}{C_{victim\ to\ GND}} \tag{2.4}$$

The coupling capacitances work as a capacitive voltage divider that regulates the voltage on the victim net, such that neighbouring nets and nodes of driven gates that are at Logic-1 elevate the victim net voltage. If the nets and nodes are at Logic-0, they reduce the victim net voltage. This model has been verified by measurements on manufactured designs [121, 82] and by its ability to predict open defect locations [72]. The amount of

charge that can be trapped on a victim net due to a full open is not well known, but $Q_{trapped}/C_{victim\ to\ GND}$ has been assumed to correspond to a variation of $[-0.3V, 0.3V]$ in [122] and a variation of $[-1V, 1V]$ in [80] and in [123] the trapped charge has been assumed negligible taking into account the possibility to eliminate this charge during IC fabrication. Due to the uncertainty of the value for the trapped charge it is not possible to fully predict the behaviour of a full open defect [124].

Capacitive coupling to neighbouring nets, as considered in the model, causes the victim net voltage to vary with the signal on a neighbouring net [67]. If the neighbouring node is down-stream from the victim net, i.e. if the logic value on the neighbouring node depends on the voltage on the victim net, it forms a feedback loop, with the possibility of oscillation or memory effects [125, 97, 82, 122].

In this section and the section before (Section 2.2.2), the literature was reviewed regarding modelling for the influence of gate tunnelling leakage and capacitive coupling on the behaviour of full open defects. This thesis will consider both sources of influence in Chapter 4. The leakage unaware model, that considers capacitive coupling, has been well studied in the literature and can be directly applied in the analysis conducted in Chapter 4 (Section 4.2). But the leakage aware model, that considers gate tunnelling leakage, is based on Spice simulations and because of the Spice simulations it is computationally intensive. More modelling is required so that the influence of gate tunnelling leakage can be considered with less computation. Such modelling is discussed in Section 4.3.

2.2.4 Test Generation and Fault Simulation of Full Open Defects

In the scenario of negligible gate tunnelling leakage, studies on test generation for full open defects [122, 126, 127] use the fact that neighbouring nets at Logic-1 elevate the victim net voltage and neighbouring nets at Logic-0 reduce the victim net voltage. Therefore, to test for stuck-at-1 behaviour, as many as possible of the neighbouring nets should have Logic-1, and to test for stuck-at-0 behaviour, as many as possible of the neighbour nets should have Logic-0. Other test generation methods that target full opens on interconnect have considered only the logic behaviour of such defects while abstracting from the capacitance values, the leakage currents and the victim net voltage [128, 97]. Instead, the suggested approach in [97] is to generate test sets that create both logic values on the net-under-test and all possible logic behaviours at the driven inputs. The probability of defect detection using such methods was estimated in [129]. A problem associated with such test generation is the challenge to effectively consider a large number of logic behaviours. This problem was addressed in [128] with a method that makes use of information about the structure of the circuit to reduce the number of logic faults to consider.

Full open defects that cause a victim net voltage that is nearly half the supply voltage

will cause both PMOS transistors and NMOS transistors of driven gate inputs to be active, leading to a current from power supply through the PMOS and the NMOS transistors to ground. If this current shows a significant increase in the IDDQ current, a current-based test will detect the full open defect. This observation was employed in [80, 122]. It should be noted that IDDQ testing does not detect all full open defects, and voltage based testing is useful for the remaining defects.

2.3 Supply Voltage-Dependent Defects and Test Methods

This section reviews previous research on defects that have supply voltage-dependent behaviour and test methods that make use of particular supply voltage settings. A summary produced specifically for this thesis to simplify comparison of the discussed defect types is given in Table 2.1 along with the recommended test method and supply voltage setting as given in the literature. This section addresses defects in general, whereas Section 2.3.1 and Section 2.3.2 address resistive bridges and full opens in more detail.

Several studies have showed that various defects with delay behaviour can be better detectable at a supply voltage lower than the nominal supply voltage [130, 67] because the delay from defects are accentuated by gates that operate using a low supply voltage. The defects that have been shown to have this behaviour are transmission gate opens, threshold voltage shift, diminished drive strength, bridges, shorts, effects of hot-carrier degradation, tunnelling opens, NMOS gate-oxide shorts and particular cases of PMOS gate-oxide shorts [131, 132, 130, 81, 133]. The behaviour of PMOS gate-oxide-shorts depends on transistor parameters and is difficult to predict [132]. The study in [130] considered transmission gate opens, threshold voltage shift and diminished drive strength. These defects would typically not cause malfunction during circuit operation using a single fixed nominal supply voltage and are therefore called delay flaws (not delay faults), but [130] suggests to apply tests at a lowered supply voltage so that flaws can be detected. Detecting delay flaws is useful, because the delay flaws can develop into malfunctions over time. For defects with supply voltage-dependent delay, the study in [133] showed that it is possible to determine the defect type by applying delay tests for a range of supply voltage settings. A test method that exploits the supply voltage-dependent delay behaviour of these defects is Very-Low-Voltage (VLV) testing [131, 130]. In VLV testing, tests are applied at a supply voltage which is typically $2 \cdot VT$, where VT is the NMOS transistor threshold voltage [130], because it is found that this very-low-voltage is effective in revealing delay defects.

Testing circuits with DVFS (Dynamic Voltage and Frequency Scaling, Section 1.1) capability has recently been considered by [134, 135, 136]. In [135] it is suggested that the supply voltage control available on-chip in a circuit with DVFS capability can be utilised

for built-in delay fault testing. The study in [134] observed by Spice simulation that transmission gate opens are best detectable at a low supply voltage using a delay fault test, however it does not have to be as low as $2 \cdot VT_{NMOS}$, which is the case in very-low-voltage testing. Further, it was reported that resistive opens on interconnect are slightly better detectable at an elevated supply voltage using a delay fault test, which is also the conclusion of [70, 79]. The conclusion of [134] is that to test circuits with transmission gates in the context of DVFS, it is necessary to apply test at more than one supply voltage.

Testing with an elevated supply voltage has been suggested in [137] as a method for stressing the circuit to make gate oxides that are too thin (i.e. a defect) more detectable. This method is not based on supply voltage-dependent behaviour of the defect, but rather the fact that thin gate oxides are made worse by the stress of an elevated supply voltage. This procedure should only be performed during a short time, due to the detrimental effect it could have on IC performance.

From the above review of defect types with supply voltage-dependent circuit behaviour, a summary is provided in Table 2.1. The table can be read as an overview of the test effort required for circuits with DVFS capability. The left-most column lists the defect types that have been characterised in the literature. The following two columns show for two supply voltage settings, corresponding to very-low-voltage and elevated voltage respectively, the test types that are effective for each defect type. For example, transmission gate opens are best detectable by a delay test applied at very-low-voltage. The fourth column refers to the papers that studied the defect type.

TABLE 2.1: Test types and test supply voltages that are effective for the reviewed defect types

Defect type	**Very Low Voltage**	**Elevated Voltage**	**Source**
Resistive bridges	Delay test / Logic test		[131, 113, 109, 138, 114, 115]
Resistive shorts	Delay test / Logic test		[131, 138, 114]
NMOS gate oxide shorts	Delay test		[132, 138]
PMOS gate oxide shorts	Delay test	Delay test	[132]
Tunnelling opens	Delay test		[138]
Transmission gate opens	Delay test		[130]
Threshold voltage shift	Delay test		[130, 138]
Diminished drive strength	Delay test		[130, 133]
Resistive opens on interconnect		Delay test	[70, 133, 79, 134]
Gate-internal resistive opens		Delay test	[79]
Full opens on interconnect	Logic test	Logic test	[70, 123]
Gate-internal full opens	Delay test / Logic test		[139]*
Oxide thinning		Logic test	[137]**
Effects of hot-carrier degradation	Delay test		[131]

*= [139] is a study on testing analog ICs, which gives other detection criterions than for digital designs.
**= [137] proposes a stress test.

Table 2.1 shows that most of the reviewed defect types are best detectable with a delay test applied at very-low-voltage. However, there are some defects such as some PMOS

gate oxide shorts and resistive opens that are best detected by testing at an elevated supply voltage. Because of the reports regarding full open defects [70, 123], where [70] has not seen any supply voltage-dependent behaviour and [123] has observed better detection at a lowered supply voltage, these are entered into the table both for very-low-voltage and elevated voltage. Furthermore, even though the entries in Table 2.1 list the tests and supply voltage that are most effective in defect detection, some defects can manifest themselves as faults only at other supply voltage settings, particularly in case of resistive bridging faults and resistive shorts [112, 114]. The fact that defects of different types manifest for different supply voltage settings indicate that testing for designs that operate on multiple supply voltage settings should be performed using more than one supply voltage to achieve high defect coverage. As the purpose of this thesis is to study the impact of supply voltage variation, as in the case of multi-voltage and DVFS designs, the reviewed studies are relevant and encouraging, but research remains to be conducted in terms of analysing the behaviour of some of the considered defects further to gain the knowledge required for developing low-cost and effective test solutions. Further analysis of the supply voltage-dependent behaviour of resistive bridge defects and full open defects, which are important defect types in deep submicron designs, is conducted in Chapter 3 and Chapter 4 respectively. In the context of these two defect types, Section 2.3.1 and Section 2.3.2 review test methods that consider the influence of supply voltage for bridges and opens respectively. In the chapters that consider supply voltage variation (Chapter 3 and Chapter 4) three supply voltage settings are considered for analysis on a gate library with the nominal supply voltage 1.2V. The three supply voltage settings are 1.2V, 1.0V and 0.8V. The lowest supply voltage setting 0.8V can be compared to the voltage used in VLV testing. So the analysis is performed for nominal voltage (1.2V), a very low voltage (0.8V) and a voltage setting in-between (1.0V). In actual multi-voltage designs, the supply voltage settings are defined by the IC designer, but these three supply voltage settings are chosen to conduct relevant analysis.

2.3.1 Supply Voltage-Aware Test Methods for Bridge Defects

This thesis discusses the supply voltage-dependent behaviour of resistive bridging faults in Chapter 3. On this topic, there has been a number of studies, including [131, 113, 112, 109, 138, 114, 134, 115, 116]. The research in [140] tested with varying supply voltage settings to find the lowest voltage level, the MINVDD (the minimum Vdd), for which the circuit was still producing correct test responses. This MINVDD was then compared with that of other circuits of the same design to identify outliers, i.e. circuits that behave differently and therefore are likely to have a defect. This test method was further developed in [138, 141]. Testing with a supply voltage that is lower than the nominal supply voltage has been suggested for detecting resistive shorts and resistive bridging faults that cause static faults, i.e. the malfunction is time-independent [112, 113, 114, 116]. The general trend is that resistive bridges are better detectable at a lowered supply

voltage [113, 109, 116], which was employed for logic testing of digital CMOS circuits in [112]. It should be noted that resistive bridges that cannot be detected by logic testing (the defect resistance is outside the detectable range) may be detectable by delay fault testing. The delay behaviour of resistive bridges also depend on the supply voltage [131, 134] with the same conclusion as for the static behaviour, that resistive bridges are generally better detected at a low supply voltage. Even though testing for bridges at a lowered supply voltage is more effective, it can be slower than testing at the nominal supply voltage, because the circuit operating at a lowered supply voltage has to be clocked using a correspondingly lowered clock frequency [114]. Furthermore, three effects that causes some defects to be better detected using another supply voltage than the lowest have been reported in the literature and loss of defect coverage due to testing only at a lowered supply voltage has been observed in [109, 114]. Firstly, particular bridge defects cannot be detected at the lowest available supply voltage [114], because the lowest supply voltage causes an undetectable logic fault for the bridge defect. As the behaviour depends on the supply voltage, a higher supply voltage can cause the same bridge defect to have detectable logic behaviour. Secondly, it has been reported that mobility saturation can cause some bridge defects to be better detected at an elevated supply voltage [115]. The saturation of the mobility of electrons is achieved at voltages lower than in the case of holes, which affects NMOS and PMOS conductance differently. For particular bridges where the NMOS network has a similar total conductance as the PMOS network, the effect of supply voltage-dependent mobility saturation can affect the supply voltage detectability of bridge defects. Thirdly, [112] has identified a mechanism that causes resistive bridges to be better detectable at other supply voltage settings than the lowest, due to supply voltage-dependent logic threshold voltage for gate inputs. From the above it can be seen that resistive bridging faults are better detectable at a low supply voltage than with nominal or elevated supply voltage. However, not all bridge defects can be tested using the same low supply voltage which means that testing should be performed using more than one supply voltage to detect all bridge defects.

2.3.2 Supply Voltage-Aware Test Methods for Full Open Defects

This thesis discusses the supply voltage-dependent behaviour of full open defects in Chapter 4. On this topic there has only been a little previous research. For the related defect type, resistive opens, the research presented in [79, 134] recommend delay fault testing using a high or elevated supply voltage, but in the context of full open defects, the existing knowledge of supply voltage-dependent behaviour consists of [70, 123]. In [70], tests were applied using several different supply voltage settings and there was no difference in the test results for full open defects. However, a more recent work [123] observed that testing using a low supply voltage improved the detection of some full opens, due to an increased sensitivity to coupling capacitance. Both studies [70, 123] considered full open defects that were not influenced by gate tunnelling leakage. No previous research

has analysed supply voltage-dependent behaviour of full open defects in the presence of gate tunnelling leakage.

2.4 Process Variation-Aware Test Methods

Process variation causes the manufactured chips to deviate from the specification to which they were designed [142,143], including variations in delay and leakage power [24]. The likelihood of cross-talk and soft errors (transient failures due to influence from outside the IC, for example radiation of charged particles) is elevated in designs that are affected by process variation [142]. To measure process variation, the study in [144] and the study in [145] suggested to include sensors on-chip. In that context, process variation is targeted by testing, in contrast to other test methods that aim to detect defects. The study in Chapter 5 of this thesis aims to detect defects inspite of process variation. However, often process variation is not a defect in itself. In the absence of defects, process variation induced behaviour is most often well tolerated because the design has a sufficient noise margin. The following sections review studies that target defects in the presence of process variation, with regard to delay, power consumption and logic behaviour respectively.

2.4.1 Delay Fault Testing under Process Variation

The study in [146] shows that resistive open and resistive short defects that, due to their resistance values, are expected to be detected by logic testing in the absence of process variation, can escape logic testing due to process variation, and concluded therefore that delay fault testing is a vital component of the test effort for such defects. The impact of process variation on delay fault testing of ICs has recently received increased attention [147, 143, 148, 149, 150, 151]. Process variation tends to affect gate-delay and subsequently path-delay and can change the ratio between rise-time and fall-time for a gate. A consequence of such variation is that it is difficult to determine the delay that is associated with the signal paths through the circuit. The problem of determining the longest path in terms of delay under process variation has been addressed in [147, 148, 149, 151]. In the presence of process variation, a path through a net is said to be longest, for that net, if there exists a configuration of IC parameter values for which the path has the maximum delay among all paths through the net [149]. So for each net, there can be multiple paths that each is longest under different configurations of the IC parameters. The approach of [147] considered correlation between different paths that arise from common sub-paths to determine a set of longest paths. The study in [148] provided a method for calculating the delay as a function of IC parameters. The method in [148] was used in [149] to select the longest paths to target with delay fault testing under process variation. This was done while using a path pruning algorithm to reduce

the set of targeted longest paths while keeping high path-delay fault coverage. Another method was employed in [151], where the paths to target with delay fault testing were determined based on statistical timing, considering the probability for each node in the circuit to be on the longest path. Another problem in delay fault testing under process variation is that some process variation induced delay is accentuated by IR-drop (fluctuations in supply voltage) and causes false delay test failures [143]. The study in [150] presented variation-tolerant delay fault test generation, to avoid false delay test failures, by minimising the switching activity due to the transitions that are caused by the delay fault test. The reasoning behind the approach in [150] is that switching activity causes IR-drop. By limiting the switching activity due to the delay fault test, process variation induced delay will lead to less false delay test failures.

2.4.2 Current-Based Testing under Process Variation

Current-based testing in the presence of process variation has been addressed in [152, 153, 53]. The impact of process variation on IDDQ testing for bridges was measured in [152] for a small adder circuit. The fault coverage was significantly reduced for 10% variation on VT and likewise for 3% variation on the W/L ratio (W and L are the width and length dimensions of a transistor gate). In [153] it is demonstrated that IDDQ testing combined with measurements of other parameters is effective in finding defects also in the presence of process variation. The basic technique is to identify outliers in scatter plots, where IDDQ measurements are on one axis and the other parameter on the other axis. Furthermore, controlling the body bias voltage helps in reducing the influence of process variation on the test results [153, 24]. The ECR test method has been shown to be tolerant of process variations [53].

2.4.3 Logic Testing under Process Variation

In testing for static defects, process variation has been considered in testing analog devices [154], and testing for bridges in static CMOS designs [99, 155]. Testing analog devices under process variation means that the stimuli used for nominal values of IC parameters are not necessarily effective for all configurations of IC parameters as they occur due to variation. Therefore [154] presented a method for generating more effective stimuli while taking process variation into account. A recent study [99] on bridging faults has shown that a test that is capable of detecting bridges for nominal values of IC parameter fails to detect some bridge defects in the presence of process variation. In [99] a new bridging fault model was developed considering process variation. The fault model is logic based, i.e. independent of IC parameters and abstract from the bridge defect resistance, motivated by the need for fast fault simulation. In [155], the fault model was improved by a method to reduce the number of considered logic faults and a test generator was presented, however the presented methodology was still independent

of IC parameters. It was shown in [155] that there is an upper limit to the number of test patterns required to achieve full defect coverage for a given bridge. This means that process variation-aware test generation for bridges is feasible. The test generation approach in [155] aims for full bridge defect coverage in the presence of process variation.

2.5 Motivation

Based on the extensive literature review carried out in this chapter, it appears that some progress has been made in examining the impact of supply voltage and process variation on manufacturing test. Research has shown that various defects have supply voltage-dependent behaviour and some are better detected at a low supply voltage whereas others are better detected at an elevated supply voltage (Table 2.1). Recent research on process variation has addressed problems in delay fault testing [147,143,148,149,150,151] and there is interest in logic testing under process variation as well [99]. This progress is encouraging, however more research is needed to gain better understanding, with the aim to develop low-cost and effective test solutions for designs with multiple supply voltages and ICs that are influenced by process variation.

The research presented in this thesis is motivated as follows.

- **Development of supply voltage-aware test generation for resistive bridge defects**
  The development of a supply voltage-aware test generation tool targeting resistive bridge defects in designs that are meant to operate using multiple supply voltage settings will be addressed in Chapter 3. For designs that operate with a single supply voltage, there is a realistic bridging fault model [104] and an effective test generation algorithm [111]. But even though supply voltage-dependent behaviour of resistive bridge defects has been observed [113, 116, 112, 115], no previous work has addressed the problem of test generation for resistive bridge defects in designs that employ multiple supply voltage settings, which is the focus of Chapter 3.

- **Analysis of supply voltage-dependent detectability of full open defects**
  Chapter 4 will analyse the supply voltage-dependent behaviour of full open defects and determine if testing for such defects can be conducted using a single supply voltage setting for designs that are meant to operate using more than one supply voltage setting. In this context, supply voltage-dependent behaviour has been observed for full open defects [123], but no previous work has conducted further analysis of this behaviour, which is the focus of Chapter 4.

- **Analysis of the impact of process variation on test quality**
  Process variation affects the performance of modern ICs but there has been little work investigating the impact of process variation on manufacturing test [99, 150,

151]. Chapter 5 will analyse the impact of process variation on test quality in the context of resistive bridge defects. This involves investigating how such defects behave under process variation and to develop effective and low-cost test generation to achieve high test quality in the presence of process variation.

Chapter 3

Testing for Resistive Bridges under Supply Voltage Variation

Interconnect resistive bridges represent a major class of defects for deep submicron CMOS. An interconnect resistive bridge defect connects two signal nets that are not designed to be connected. The connection caused by the defect has some resistance value, which is fixed but of unknown value. Typically a Resistive Bridging Fault (RBF) is modelled by adding a resistor to the netlist. A resistive bridge defect has a fixed bridge resistance value, but the modelling of a RBF represents a range of possible defect resistance values that causes the same logic behaviour. To cover all bridge defects that can cause malfunction at a given bridge location, it is often necessary to detect several RBFs corresponding to different defect resistance ranges.

It has been shown in [112, 114] that the defect coverage (i.e. how much of the range of possible bridge resistance that is covered by the test) of a test set targeting interconnect resistive bridging faults can vary with the supply voltage used for test application. The fact that the defect coverage for resistive bridging faults can vary with the supply voltage means that, depending on the operating supply voltage setting, a given RBF may affect the correct operation of the design. Consequently, if the operating supply voltage for the design is known, the test should be applied at that supply voltage to provide the required defect coverage. However, if the design is meant to operate using more than one supply voltage setting, as is the case in many low-power designs (Section 1.1 and Section 1.2), it may be necessary to perform testing at more than one supply voltage setting, to detect defects which manifest themselves as faults only at particular supply voltages. No previous work has demonstrated how such multi-voltage tests should be generated.

The aim of this chapter is to propose a new automatic test generation method targeting RBFs as they can occur for a set of supply voltage settings. This involves the development of a suit of software tools including tools for bridge location extraction, gate

library characterisation, supply voltage-aware fault simulation and ATPG for bridging faults, and test-set size reduction. The proposed test generation method improves on previous studies by considering more than one supply voltage setting and this chapter introduces the concept of supply voltage-specific test sets.

The analysis and experimentation is performed on a 0.12μm gate library from ST Microelectronics, with ISCAS85 and -89 benchmark circuits that are synthesised using Synopsys Design Compiler and placed-and-routed using Cadence Encounter to enable identification of realistic bridge locations. The analysis and the methods discussed in this chapter are not restricted to 0.12μm technology. Bridge behaviour can be analysed in the same way also in more recent technologies (deep sub-micron) because the experiments in this chapter are based on analog simulation in Cadence Spectre. The suite of software tools is implemented in C++ using a solver for the Boolean Satisfiability problem as ATPG-engine (Section 1.4.4 and Appendix B).

3.1 Analysis and Statement of Problem

This section presents an example of a bridge defect, to show that a resistive bridge can lead to a logic fault. Subsequently, the modelling of resistive bridging faults is discussed (Section 3.1.1) along with the circuit behaviour in the presence of resistive bridge defects (Section 3.1.2). Furthermore, a definition of defect coverage for resistive bridges in designs that operate with multiple supply voltages is defined (Section 3.1.3) and prior work on test generation is reviewed (Section 3.1.4).

The following example shows the behaviour of a resistive bridging fault. A simulation of an RBF fault site (Figure 3.1) was conducted using Cadence Spectre and gates from a 0.12μm gate library (Appendix A defines the concept of a fault site). Two nets N_{high} and N_{low} are bridged with a defect of resistance R_{sh}. Net N_{high} is driven high when at least one of the inputs to the driving gate, a two input NAND, is at Logic-0, i.e. when net a is at Logic-0. Net N_{low} is driven by an and-or-invert gate (AO2HS), which has an input assignment that causes it to drive N_{low} to Logic-0. The particular input assignment has been selected for illustration purposes. The voltage on the bridged nets depend on the gate output conductance of the gates that drive the bridged nets, which in turn depends on the input assignment. Therefore, this example has an input assignment that has been selected to show the impact of bridge resistance. Net N_{high} has one successor gate, which is a three input NOR gate. A NOR gate will only propagate the signal (in this case the logic value for N_{high} to the successor gate output, net b) if the other inputs are at Logic-0, which is the case in Figure 3.1. Similarly, net N_{low} has one successor gate, a two input AND gate, which has to have Logic-1 on the side input to propagate from N_{low} to net c.

To determine if a typical resistive bridge causes static or dynamic behaviour, transient

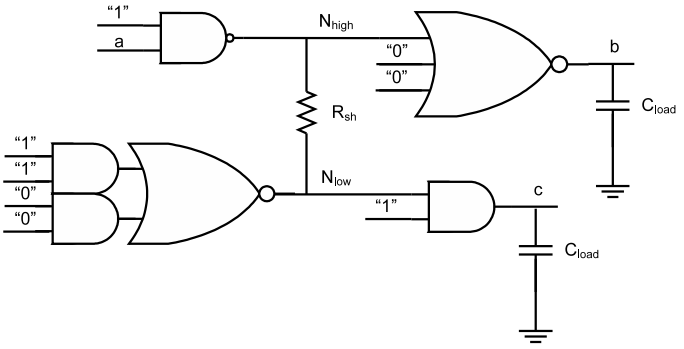
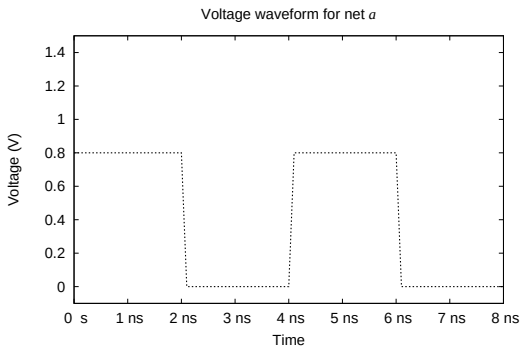


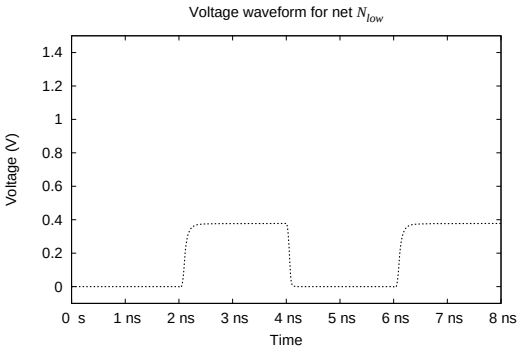
FIGURE 3.1: Example fault site with bridging defect

simulation of the circuit in Figure 3.1 was performed with 0.8V supply voltage, a defect resistance of R_{sh}=500Ω and C_{load} was set to the input capacitance of a small inverter. The bridge resistance R_{sh}=500Ω has been chosen to show the impact of a bridge on circuit behaviour and will be used also in a later example (Section 3.1.5). The logic value on net a was varied to activate and deactivate the bridge defect and the resulting waveforms are shown in Figure 3.2. Figure 3.2(a) is the voltage on net a, which represents the simulation stimulus. Between time 0 and time 2ns the signal is Logic-1. Between 2.1ns and 4ns, the signal is Logic-0, and so on. In the fault-free circuit, there is no path from net a to net N_{low} in the fault-free circuit. Therefore, the voltage on N_{low} should be consistently at ground voltage, independent of net a. However, for a defect with 500Ω resistance, the defect causes 0.4V on net N_{low} at time 3ns when net a has 0V (Figure 3.2(b)). At this time N_{high} is driven high, which causes the bridge defect to be active, as it connects two nets N_{high} and N_{low} which are driven to opposite logic values. The 0.4V on net N_{low} is propagated as a faulty Logic-1 on net c as can be seen in Figure 3.2(c). It should be noted that Logic-1 on net c is faulty behaviour, resulting from the defect which is activated when net a is at Logic-0.

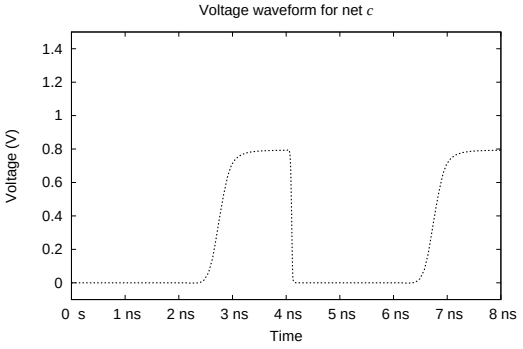
From Figure 3.2 it can be seen that the circuit behaviour due to the specific considered bridging defect is static. The differentiation between static and dynamic behaviour is in the time-dependence of the faulty behaviour. The faulty behaviour of a static defect remains as the time progresses, while the faulty behaviour of a dynamic defect dissipates with time to eventually return to fault-free logic behaviour. The only delay behaviour in the example of Figure 3.2 is a minor delay (<0.5ns) on the successor gate output, net c, when the faulty behaviour starts, but this does not make the defect behaviour dynamic because the faulty behaviour continues until net a goes to Logic-1 at time 4ns which deactivates the bridge defect. From the above example, it can be seen that bridge defects can change the logic behaviour of CMOS circuits and that the changed behaviour of the considered 500Ω defect is static. It is widely accepted that most bridge defects with low resistance have static behaviour [101, 109, 104]. Most resistive bridge defects have a low resistance [64], however defects with up to 20kΩ have been reported. This means that many of the resistive bridges have such a low resistance that they have severe impact on



(a) Stimulus on net a



(b) The behaviour on net N_{low} with a 500Ω bridge



(c) The behaviour on net c with a 500Ω bridge

FIGURE 3.2: Waveforms for simulation on the fault site in Figure 3.1

circuit operation. Indeed, in [66] it was shown that more than 44% (up to 82%) of bridges can be detected by logic testing. The remaining bridge defects are either undetectable or only detectable by delay fault testing and current based testing. The resistance of the bridge defect should be considered because there is a wide range of possible resistance values and corresponding circuit behaviour for physical bridge defects.

3.1.1 Modelling of Resistive Bridging Faults

This section summarises the literature review on bridge fault modelling from Section 2.1 to put the previous work in the context of the study performed in this chapter and to make the chapter self-contained.

Since the bridge resistance value can take any value from 0Ω to practically infinite resistance and since the resistance value is not known in advance, accurate modelling of RBFs is a difficult task [96]. Various methodologies for determining the behaviour for a particular defect resistance have been suggested [102, 96, 66]. As can be seen from the review of fault models for resistive bridge defects in Section 2.1, the most accurate model in the literature is the parametric bridging fault model, which considers both the influence of the bridge resistance and the fact that inputs have different logic

threshold voltage values. The parametric bridging fault model has been used in several studies [109, 106, 110, 111]. Considering the parametric bridging fault model, a number of simulation techniques [110, 117, 107] have been proposed.

It should be noted that some papers consider feedback bridges [55, 61, 59], others do not. In this work the problem of testing devices that have oscillating behaviour due to feedback bridges is considered orthogonal to the problem of testing in the presence of supply voltage variation for resistive bridging faults. This means that test-generation for such designs can be developed independently of test generation for feedback bridges. Throughout the rest of this chapter, only non-feedback bridges are discussed.

3.1.2 Circuit Behaviour in the Presence of Resistive Bridging Faults

The main difficulty in RBF simulation and test generation arises from the fact that the bridging resistance is a continuous parameter which in a given defect is fixed but not known in advance. A recent approach based on interval algebra [114, 107] allowed treating the whole continuum of bridge resistance values R_{sh} from 0Ω to ∞ 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 nets from 0V (Logic-0) or supply voltage (Logic-1) to some intermediate values, which will depend on the bridge resistance, R_{sh}. The logic behaviour due to the R_{sh} value and the resulting intermediate voltages on the bridged nets, can be expressed in terms of the logic values seen by the gate inputs that are driven by the bridged nets, based on how the voltage on each input compare with the logic threshold voltage for the input. The logic threshold voltage is a simplified view of the input-to-output transfer characteristics of a gate with respect to a particular input. A voltage above the logic threshold is seen as Logic-1 and otherwise as Logic-0. The logic threshold voltage is specific to each input and gate type. The logic threshold voltage Th is the voltage on the input that causes the gate output to change logic value and should not be confused with the transistor threshold voltage VT.

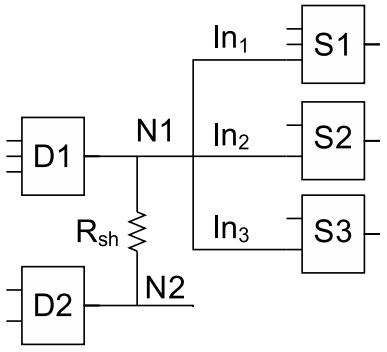


FIGURE 3.3: Bridge defect example

A typical bridging fault scenario is illustrated in Figure 3.3. Two gates, D1 and D2 are driving the bridged nets, while S1, S2 and S3 are successor gates, i.e. gates having

inputs driven by one of the bridged nets. The resistive bridge affects the logic behaviour only when the two bridged nets are driven at opposite logic values. For example, 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_{N1}) on the equivalent resistance of the physical bridge is shown in Figure 3.4 for a supply voltage of 1.2V. The deviation of V_{N1} from the ideal voltage level (supply voltage) is largest for small values of R_{sh} and decreases for larger values of R_{sh}. Figure 3.4 is the behaviour for a given input assignment to the gates D1 and D2. Another input assignment would change the V_{N1} curve in Figure 3.4. For some input assignments net N1 would be driven low. Still the deviation from the ideal voltage level would be largest for small R_{sh} values and decrease for larger values of R_{sh}.

To translate this analog behaviour into the digital domain, the logic threshold voltage levels $Th1$, $Th2$ and $Th3$ of the successor gates S1, S2 and S3 have been added to the V_{N1} plot. For each value of the bridging resistance R_{sh}, the logic values read by inputs In_1, In_2 and In_3 can be determined by comparing V_{N1} with the logic threshold voltage of the corresponding input. These values are shown in the second part of Figure 3.4. Crosses are used to mark the faulty logic values and ticks to mark the correct ones. It can be seen that, for bridges with $R_{sh} > R_3$, the logic behaviour at the fault site is fault-free (all inputs read the correct value), while for bridges with R_{sh} between 0 and R_3, one or more of the successor inputs are reading 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 behaviour. Methods for calculating the critical resistance have been presented in [109, 107, 96].

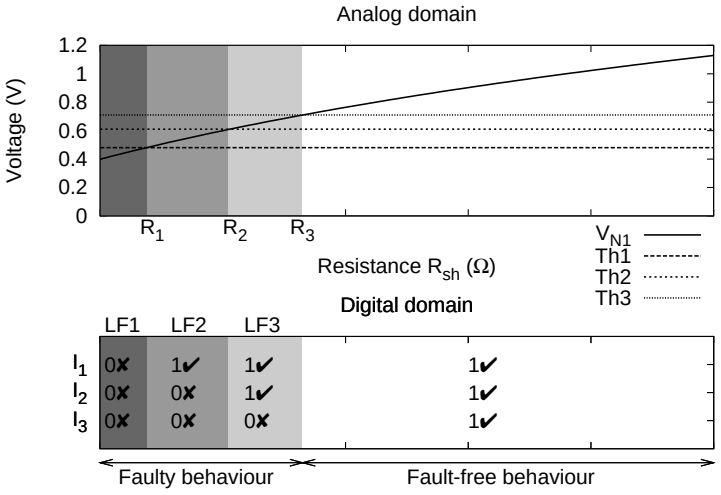


FIGURE 3.4: Bridging fault behaviour

Bridging resistance intervals can be identified based on their corresponding logic behaviour. For example, bridges with $R_{sh} \in [0, R_1]$ exhibit the same faulty behaviour in the digital domain (all successor gate inputs read the faulty logic value). Similarly,

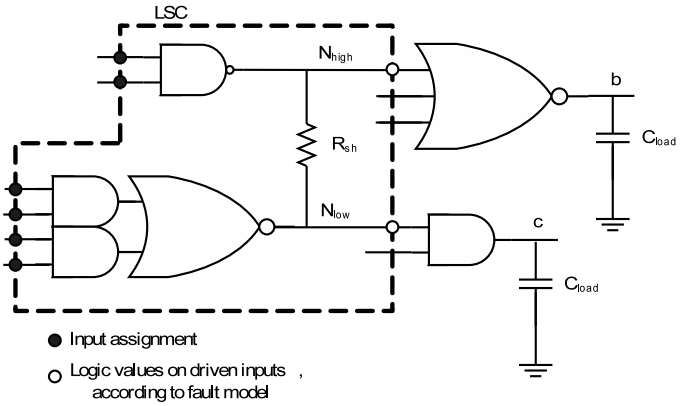


FIGURE 3.5: Logic State Configuration example

for bridges with $R_{sh} \in [R_1, R_2]$, successor gates S2 and S3 read the faulty value, while S1 reads the correct value, and finally, for bridges with $R_{sh} \in [R_2, R_3]$ only S3 reads a faulty value while the other two successor gates read the correct logic value. Consequently, each interval $[R_i, R_{i+1}]$ corresponds to a distinct logic behaviour occurring at the bridging fault site. It should be noted that the logic behaviour and the resistance ranges also depends on the input assignment to the gates D1 and D2. For the example in Figure 3.4, net N1 is driven high, but if net N1 was driven low, because of another input assignment, the logic behaviour and the resistance ranges would be different.

The logic behaviour at the fault site can be captured using a data structure which will be further referred to as logic state configuration (LSC). An LSC consists of the logic values at the inputs of the driving gates and the logic values detected by the inputs of the successor gates, as is shown in the example in Figure 3.5. The logic values at the inputs of the driving gates are marked with black dots and the logic values detected by the inputs of the successor gates are marked with black circles filled with white. The LSC is a purely logic-based construct and does not contain a model for the analog behaviour of the defect. However, a model for the defect can be used to generate a set of LSCs to model the logic behaviour in the presence of a resistive bridge (in the same way as in the example of Figure 3.4). This set contains LSCs for all input assignments that activate the bridge to take into account the fact that the gate output conductance of a gate depends on its input assignment and the gate output conductance influences the voltage on the bridged nets. LSCs that correspond to faulty logic behaviour are directly associated with logic faults (LF). An LSC is said to be detectable, if at least one test pattern exists which can justify the net values specified by the LSC (input assignment and logic values on driven inputs, Figure 3.5) and also make the faulty behaviour observable at the primary outputs. An LSC for which no such test pattern exists is referred to as an undetectable LSC. This means that undetectable LSCs corresponding to faulty behaviour cannot occur during the functional operation of the circuit, and consequently only detectable LSCs have to be targeted during test generation to ensure correct operation of the circuit.

3.1.3 Multi-Voltage-Aware Defect Coverage Metric

The union of the resistance intervals corresponding to detectable LSCs forms the Global Analog Detectability Interval (GADI) [105, 107] (for background on the terminology of Analog Detectability Interval, see Section 2.1.3). This means that GADI represents the entire range of detectable physical defects. Given a test set T, the Covered Analog Detectability Interval (CADI) represents the range of defect resistances covered by T through the detection of detectable LSCs. The CADI for a bridging defect is the union of one or more resistance intervals, the union of intervals corresponding to detected LSCs [112, 105, 114, 107, 111].

As can be seen from Section 2.1.3, the fault coverage FC according to the parametric bridging fault model is calculated by comparing how much of the GADI has been covered by the CADI. In this chapter, the concept called fault coverage in [107] is called defect coverage[1]. When the CADI of test set T is identical to the GADI of fault f, T is said to achieve full defect coverage for f. The defect coverage metric can be seen in Equation 2.2 for a single supply voltage scenario. In Equation 2.2, B is the considered set of bridges.

Previous work on test generation for bridging faults [111] uses the concept of GADI assuming a fixed supply voltage scenario. In this work, the concept of GADI is extended to capture the dependence of the bridging fault behaviour on the supply voltage by defining the Multi-Voltage GADI, called MVGADI, Equation 3.1, as the union of supply voltage specific GADIs for a given design. CADI is extended in the same way to MVCADI, Equation 3.2, which represents the union of the supply voltage specific CADI that result from test sets for the different supply voltages. Test set T_{Vddi} corresponds to supply voltage $Vddi$. The new defect coverage metric that capture the influence of the supply voltage is called Multi-Voltage Defect Coverage (MVDC) and is shown in Equation 3.3.

$$\text{MVGADI}(b) = \bigcup \text{GADI}(Vddi, b) \tag{3.1}$$

$$\text{MVCADI}(T, b) = \bigcup \text{CADI}(T_{Vddi}, b, Vddi) \tag{3.2}$$

$$\text{MVDC}(T, B) = \frac{\sum_B \frac{resistance\ detected}{detectable\ resistance}}{\|B\|} = \frac{\sum_{b \in B} \frac{\|\text{MVCADI}(T,b)\|}{\|\text{MVGADI}(b)\|}}{\|B\|} \tag{3.3}$$

As was seen in the literature review of Section 2.1, alternative models and various ways of determining the logic behaviour of resistive bridges have been suggested [102, 104, 96], but to be able to reason about the influence of varying the supply voltage on the

[1]Defect coverage is a more correct term for the same concept in the context of resistive bridges, because testing for resistive bridges is aimed at covering the full range of bridge resistance (GADI) rather than the full set of faults that the bridge defects can cause.

detectability of physical defects it is essential to include the resistance value of the defect [112,114]. Some studies have tested for resistive bridges at a supply voltage other than the nominal, like Very-Low-Voltage testing [114] or other approaches [113,109,115, 116]. In the literature review concerning these methods, Section 2.3.1, it can be seen that even though in general resistive bridge defects are better detected at a low supply voltage than a high supply voltage, there are some resistive bridges that manifest at other supply voltage settings than the lowest [112,114,115]. The observations in [112,114,115] show that testing for resistive bridging faults using only a single supply voltage setting will not detect all bridging defects. Section 3.1.5 gives an example that motivates why testing using multiple supply voltages is required.

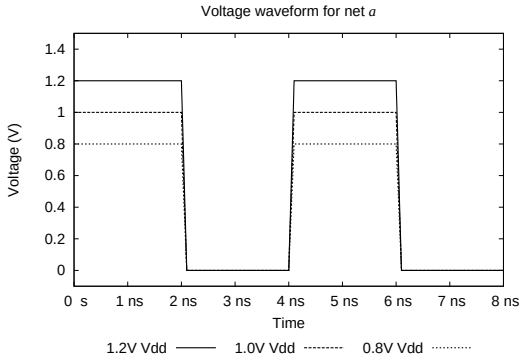
3.1.4 Test Generation

Several test generation methods for resistive bridging faults have been proposed [109, 106,110] and more recently [97,111]. The method presented in [106] is to guarantee the application of all possible logical configurations at the bridge fault site without detailed electrical analysis, which means that all bridge resistance values are covered without being explicitly considered. However, considering all possible logic configurations as in [106] leads to a large number of logic faults to process. In [97], the effect of a bridge on a node with fan-out is modelled as a multiple line stuck-at fault. The model in [97] does not consider the bridge resistance. Such methods as [106,97] are not useful for conducting the analysis described in Section 3.1.5, which determines the impact of supply voltage variation on testing for resistive bridging faults, since it is required to consider the bridge resistance. The literature review in Section 2.3.1 found that the most efficient test pattern generation approach that is known [111] combines the advantages of [109] and [110]. The method in [109] targets the logic fault that corresponds to the largest range of bridge resistance and the method in [110] targets the whole range of bridge resistance as a set of logic faults corresponding to disjunct resistance intervals. Furthermore, the method in [111] employs interval-based fault simulation to keep a log of covered and not-yet-covered bridge defects. It should be noted that all test generation methods described above [109, 106, 97, 111, 110] are intended for a fixed supply voltage setting, i.e. all tests are applied at the same supply voltage. In the next section it is explained why it is sometimes necessary to use more than one supply voltage setting during test to ensure full bridging defect coverage for designs with multiple supply voltage settings.

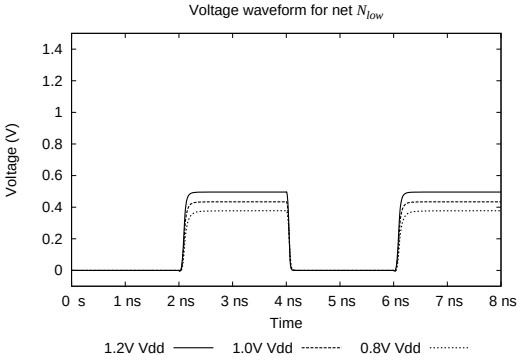
3.1.5 Motivation of Testing Using Multiple Supply Voltages

This section provides an analysis of the effect of varying supply voltage on bridging fault behaviour, which provides the starting point for the work presented in this chapter.

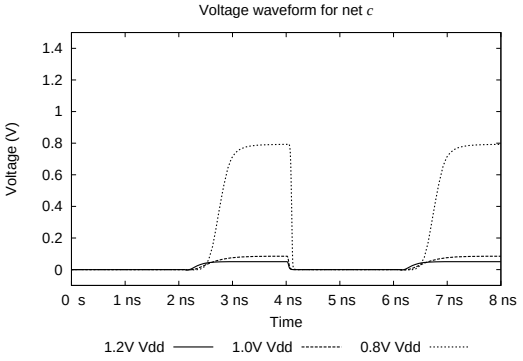
First, the example in Figure 3.1 and Figure 3.2 is revisited to demonstrate supply voltage dependent behaviour of the example bridge of 500Ω. Repeating the same experiment as in Figure 3.2 for three supply voltage settings, 0.8V, 1.0V and 1.2V, produces the waveforms in Figure 3.6. The stimulus on net a, shown in Figure 3.6(a) are alternating between Logic-1 and Logic-0, according to the supply voltage setting, which determines the voltage for Logic-1. The stimulus on net a affects whether N_{high} is driven high or low. When N_{high} is driven to Logic-1, the bridge is active and both N_{high} and N_{low} get deteriorated voltage, i.e. no longer supply voltage and ground voltage respectively. The resulting voltage on N_{low} can be seen in Figure 3.6(b). For all three supply voltage settings, the voltage on N_{low} is close to 0.4V. The signal on N_{low} is propagated through the AND gate to produce a signal on net c as shown in Figure 3.6(c). It can be seen in Figure 3.6(c), that net c is at Logic-0 for supply voltage 1.2V and 1.0V, but with Logic-1 for 0.8V. This example shows a bridge defect that has supply voltage dependent behaviour and follows the general trend of better defect detection at a lowered supply voltage.



(a) Stimulus on net a



(b) The behaviour on net N_{low} with a 500Ω bridge

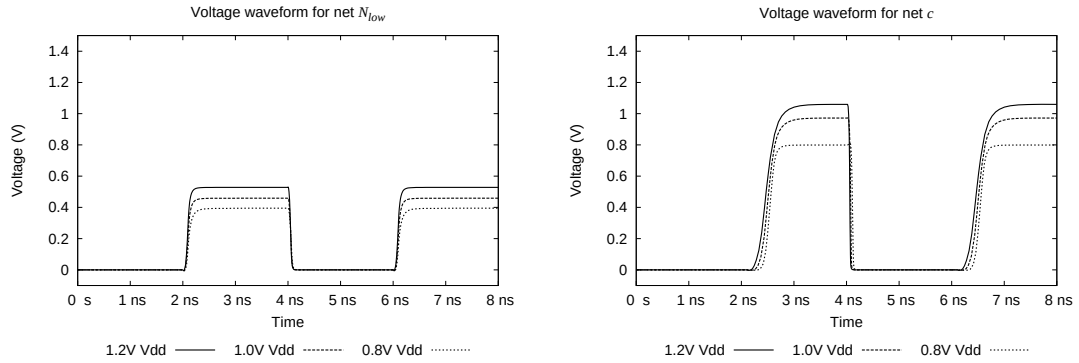


(c) The behaviour on net c with a 500Ω bridge

FIGURE 3.6: Waveforms for simulation on the fault site in Figure 3.1 for supply voltage settings 0.8V, 1.0V and 1.2V and a 500Ω bridge

It should be noted that not all bridge resistances and input assignments to the gates that drive the bridged nets lead to supply voltage dependent behaviour. The example in Figure 3.1 and Figure 3.6 is based on a configuration selected for demonstration

purposes.

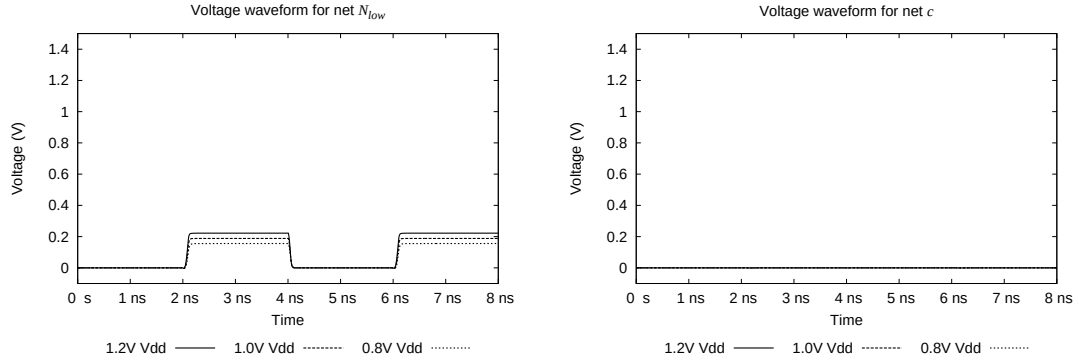


(a) The behaviour on net N_{low} with a 100Ω bridge

(b) The behaviour on net c with a 100Ω bridge

FIGURE 3.7: Waveforms for simulation on the fault site in Figure 3.1 for supply voltage settings 0.8V, 1.0V and 1.2V and a 500Ω bridge

The supply voltage dependent behaviour shown in Figure 3.6 is due to the bridge resistance and the input assignment to the gates that drive the bridged nets. For other bridge resistances or input assignments, there may not be supply voltage dependent behaviour. Figure 3.7 shows the waveforms for bridge resistance $R_{sh} = 100\Omega$. The stimulus on net a is the same as in the experiment of Figure 3.6. It should be noted that the voltage on net N_{low} is slightly higher in Figure 3.7(a) than in Figure 3.6(b). In fact, the voltage on N_{low} is high enough to be seen as Logic-1 for all supply voltage settings as can be seen by the high voltage values around time 3.5ns in Figure 3.7(b). As Logic-1 is the faulty behaviour on net N_{low} the bridge defect with $R_{sh} = 100\Omega$ can be detected using any supply voltage and there is no supply voltage dependent behaviour.



(a) The behaviour on net N_{low} with a 500Ω bridge

(b) The behaviour on net c with a 500Ω bridge

FIGURE 3.8: Waveforms for simulation on the fault site in Figure 3.1 for supply voltage settings 0.8V, 1.0V and 1.2V and a 500Ω bridge and the altered input assignment $\{1, 1, 1, 1\}$ to the gate that drives N_{low}

Figure 3.8 shows the waveforms for simulating the circuit in Figure 3.1 with $R_{sh} = 500\Omega$ but with another input assignment to the gate that drives net N_{low}. Instead of the input assignment shown in Figure 3.1, $\{1, 1, 0, 0\}$, the assignment $\{1, 1, 1, 1\}$ is used,

which makes the gate stronger in driving N_{low} towards ground (increased gate output conductance). This reduces the voltage on N_{low} compared to with the original input assignment, as shown in Figure 3.8(a) (compare with Figure 3.6(b)). In fact, the voltage on N_{low} is low enough to be seen as Logic-0 for all supply voltages as can be seen by the consistently low voltage on net c as shown in Figure 3.8(b). Logic-0 is the fault-free behaviour on net N_{nlow}, which shows that the input assignment $\{1, 1, 1, 1\}$ on the gate that drives N_{low} cannot detect the 500Ω defect for any supply voltage and there is no supply voltage dependent behaviour.

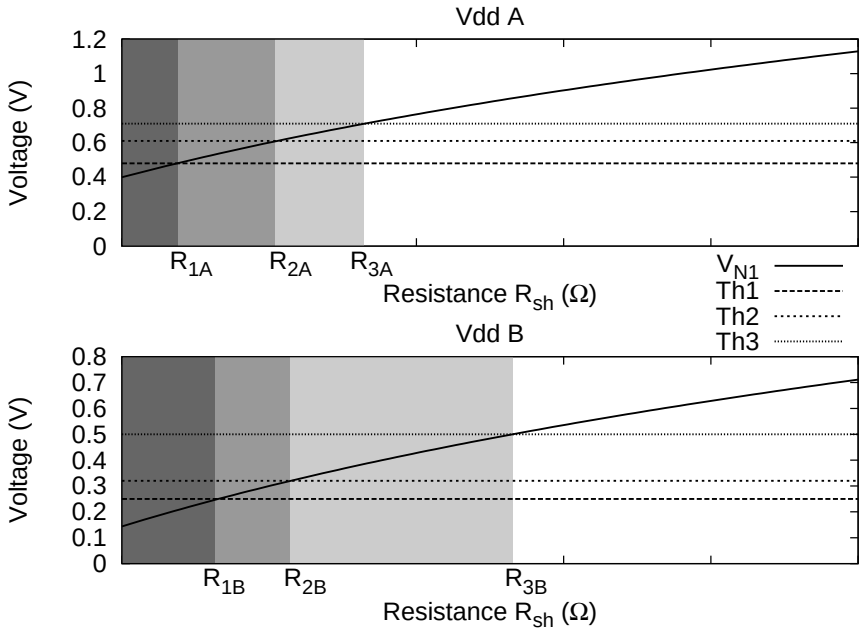


FIGURE 3.9: Effect of supply voltage on bridging fault behaviour: Analog domain

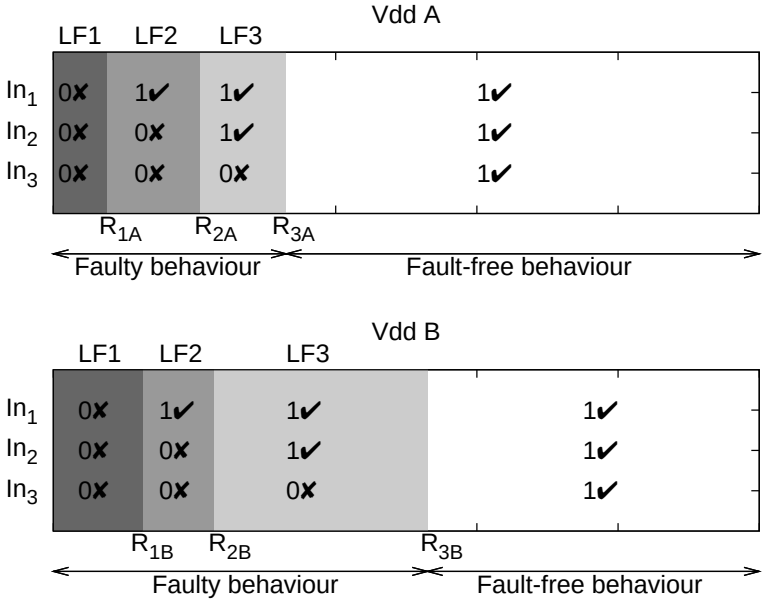


FIGURE 3.10: Effect of supply voltage on bridging fault behaviour: Digital domain

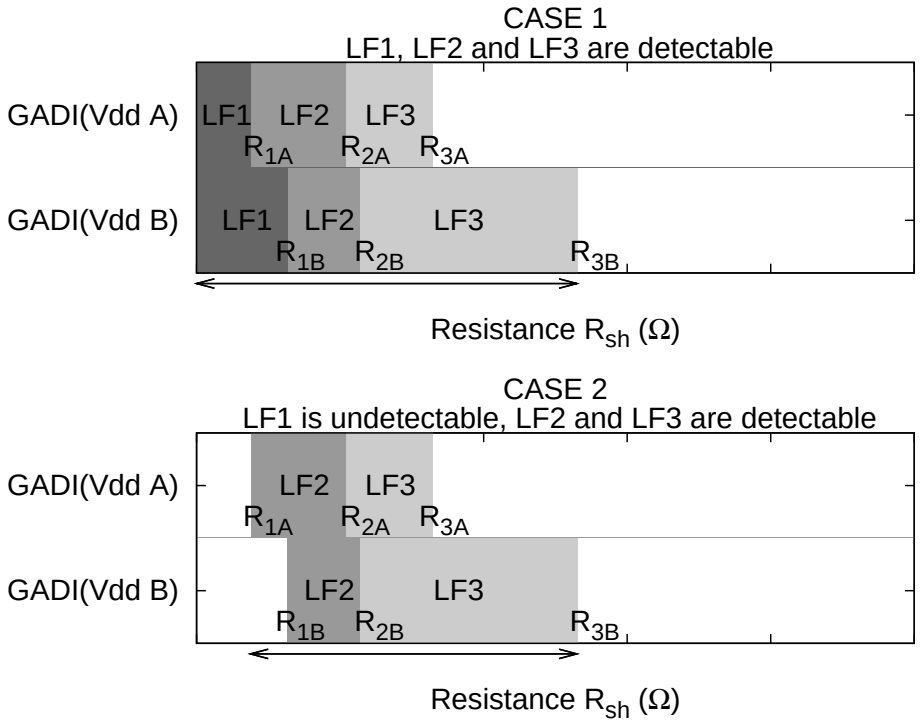


FIGURE 3.11: Effect of supply voltage on bridging fault behaviour: Observable bridging resistance ranges

Next, to see why supply voltage dependent behaviour require tests to be applied using more than one supply voltage, and to build an understanding of how such tests could be identified, consider the example circuit in Figure 3.3. Figure 3.9 shows the relation between the voltage on the output of gate D1 (Figure 3.3) and the bridging resistance for two different supply voltages $\mathrm{Vdd_A}$ and $\mathrm{Vdd_B}$. The diagrams in Figure 3.10 show how the analog behaviour at the fault site (Appendix A) translates into the digital domain. In this example, three distinct logic faults LF1, LF2 and LF3 could be identified for each supply voltage setting. However, because the voltage level on net N1 does not scale at the same rate as with the logic threshold voltages of S1, S2 and S3 when changing the supply voltage, the resistance intervals corresponding to LF1, LF2 and 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 $\mathrm{Vdd_A}$, a test pattern targeting LF3 will detect bridges with $R_{sh} \in [R_{2A}, R_{3A}]$, while at $\mathrm{Vdd_B}$ it will detect a much wider range of physical bridges ($R_{sh} \in [R_{2B}, R_{3B}]$). Analysing this from a different perspective, a bridge with $R_{sh} = R_{3B}$ will cause a logic fault at $\mathrm{Vdd_B}$ but not at $\mathrm{Vdd_A}$. To see the need for using multiple supply voltage settings during test, consider the following two scenarios. In Case 1 (Figure 3.11) all three logic faults LF1, LF2 and LF3 are detectable. Figure 3.11 shows the ranges of bridging resistance corresponding to faulty logic behaviour for the two supply voltage settings which are the GADI sets corresponding to the two supply voltage settings. It can be seen that GADI($\mathrm{Vdd_A}$) represents about 45% of the overall

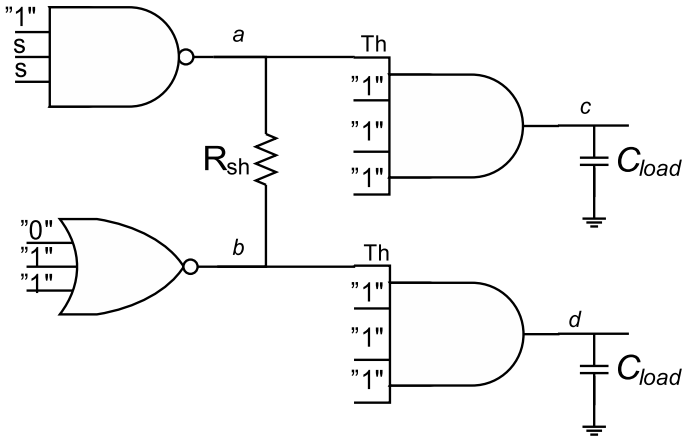
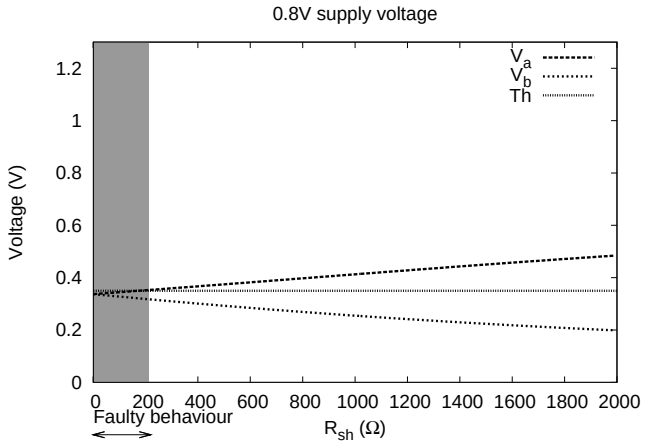


FIGURE 3.12: Example bridge defect that is best detected using the highest supply voltage setting

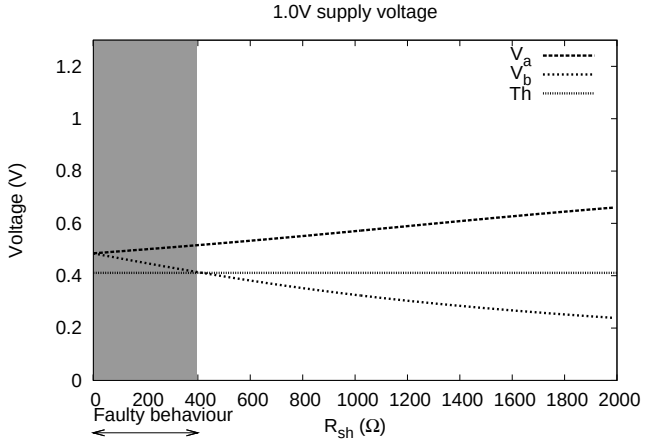
MVGADI while GADI($\mathrm{Vdd_B}$) fully covers the overall MVGADI. This means that a test set detecting LF1, LF2 and LF3 will achieve full bridging defect coverage when applied at $\mathrm{Vdd_B}$. In Case 2 from Figure 3.11, only LF2 and LF3 are detectable, which means that there is no test pattern which can detect LF1. In this case, GADI($\mathrm{Vdd_A}$) represents about 30% of the overall MVGADI while GADI($\mathrm{Vdd_B}$) represents about 90% of the overall MVGADI. This means that full bridging defect coverage cannot be achieved using a single supply voltage setting.

The example in Figure 3.2 shows the waveforms for a bridge defect that is best detected at the lowest supply voltage. In the following, an example bridge defect is given for which testing using the highest supply voltage is necessary. Figure 3.12 shows the considered bridge fault site. Net a is driven by a 3-input NAND gate, with the first input set to Logic-1 and the other two inputs are both controlled by the stimulus s. When the stimulus signal s is Logic-0, net a is driven high and when s is Logic-1, net a is driven low. Net a is bridged to net b through the bridge defect with the resistance R_{sh}. Net b is driven by a 3-input NOR gate with the input assignment $\{0, 1, 1\}$. Both net a and net b drive the A input of a 4-input AND gate, leading to net c and net d respectively. The capacitance C_{load} seen by net c and net d is the typical input capacitance of an inverter. The gates and the input assignments in this example have been selected for demonstration purposes, to show a bridge defect that is best detectable using the highest supply voltage setting (1.2V from the set of supply voltage settings 1.2V, 1.0V and 0.8V).

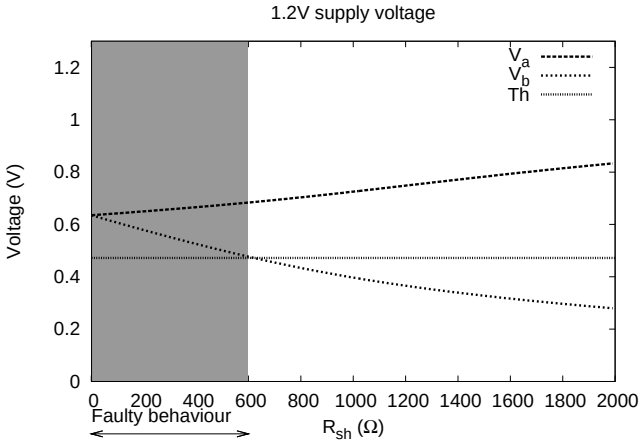
Figure 3.13 shows the voltage on the bridged nets, net a and net b, as a function of the bridge resistance R_{sh} along with the logic threshold voltage Th for the A input of the 4-input AND gates in Figure 3.12. As can be seen from Figure 3.13(a), Figure 3.13(b) and Figure 3.13(c), the range of detectable bridge resistance increases with supply voltage. It should be noted that bridge resistances in the range $[400\Omega, 600\Omega]$ can only be detected using the 1.2V supply voltage setting. This is an example of a bridge defect that can only be detected using the highest supply voltage setting.



(a)



(b)



(c)

FIGURE 3.13: The voltage on net a and net b in Figure 3.12 as it depends on the bridge resistance and the supply voltage

From the analysis above it can be concluded that to achieve full MVGADI coverage in a system with variable supply voltage, situations can arise where it is necessary to apply tests at several supply voltage settings. It can also be concluded that a test pattern targeting a particular defect would be ineffective if applied using a supply voltage for which the defect does not cause malfunction. It would be desirable to determine the test patterns for each supply voltage settings, so that the test patterns effectively contribute to the overall defect coverage. A methodology for achieving this is presented in Section 3.2.4.

3.2 Software Tool Suite

This section describes an effective test generation method for resistive bridging faults in circuits with multiple supply voltage settings. The method is implemented in a flow using a software test generation suite called Software for Multi-Voltage RBF Analysis and Test (SMuVoRBAT). The software test generation suite has been developed for the purpose of this study. SMuVoRBAT is capable of bridge location list generation, supply voltage characterisation of gates, supply voltage-aware fault simulation and multi-voltage-aware test pattern generation. The software tools and existing design, test generation and simulation tools are integrated into a flow used to conduct experiments. The flow is shown in Figure 3.14. The input data required to run the flow include a benchmark design with a set of supply voltage settings and a gate library. Synthesis is performed on the benchmark design to produce a synthesised gate-level netlist using the gate library. The synthesis tool used within this flow has been Synopsys DC Compiler and the gate library was a 0.12μm CMOS library from ST Microelectronics. The synthesised netlist is then used in a place-and-route tool to produce a layout for the design. The place-and-route tool used was Cadence Encounter and the layout was used to identify possible bridge locations, with the purpose of providing the test generator with a list of bridge locations to target. The bridge location identification was done through extraction of coupling capacitance as discussed in Section 1.4.8, also using Cadence Encounter, and subsequently, feedback bridges were identified and removed. The gate library is characterised for the supply voltage settings using a software tool developed for this purpose, which in turn performs simulations using Cadence Spectre. The outcome of the characterisation is a list of supply voltage specific logic threshold voltages for each input of each gate in the library. The logic threshold voltages are used to translate the input voltage to a logic value, as explained in Section 1.4.8. When the flow has performed the above steps, it is possible to apply the Multi-Voltage Test Generator (MVTG) to produce supply voltage specific test sets, or use the Supply Voltage-Aware Resistive bridging Fault Simulator (SVARFS) to evaluate test sets for different supply voltage settings.

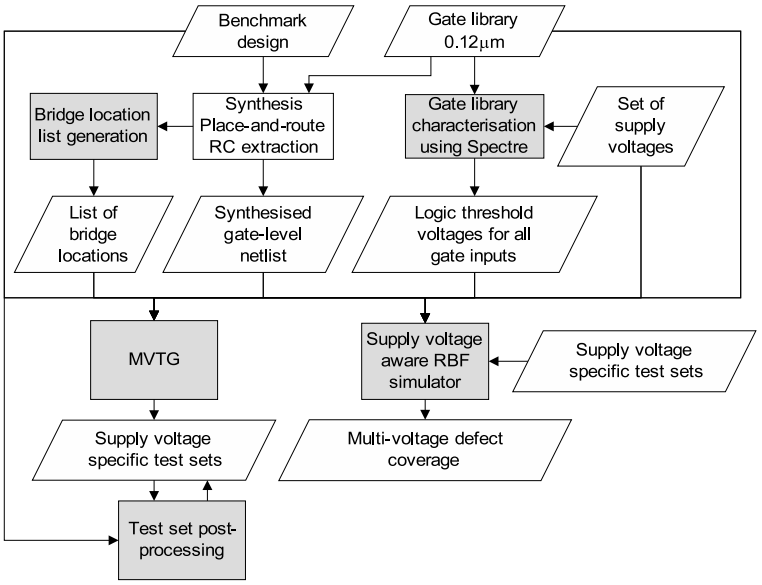


FIGURE 3.14: SMuVoRBAT tool flow

3.2.1 Bridge Location List Generation

The SMuVoRBAT tool suite includes bridge location list generation. To identify possible bridge locations, the flow in Figure 3.14 includes place-and-route of the considered design using Cadence Encounter. In the place-and-route tool, there is an option to extract coupling capacitance between signal nets. This feature is called ExtractRC. The coupling capacitance value for two nets can be used to determine if they are close to each other in the layout. The higher capacitance value, the more chance of a bridge occurring between the two nets. The capacitance becomes high if the two nets run in parallel for a long distance or if they cross, on different metal layers. ExtractRC reports the coupling capacitance in a file of the format SPEF (Standard Parasitic Exchange Format). Entries to the file are nets with coupling capacitance higher than a limit of 0.1aF. The bride location list generation tool reads the SPEF file and makes a list of the net pairs of the entries. These net pairs are possible bridge locations. Some of these bridge locations constitute feedback bridges. To find the net pairs that are non-feedback bridges, the bridge location list generation tool performs a two-way search for each pair of nets. For a pair of nets A and B, the tool searches for a path from net A to net B. First the search is in the output-cone of net A, meaning looking through the fanout of net A, then the fanout of the gates that are driven by net A and so on until the primary outputs are reached, or net B is found. The output-cone of net A is the set of nets that can depend on the logic value on net A. If net B is in the output-cone of net A, the pair of nets A and B is a feedback-bridge. Similarly, a search is performed in the output-cone of net B to find net A. Searching through the pairs of nets as described above makes it possible to identify and remove feedback bridges. The remaining net pairs are the list of possible bridge locations identified by the bridge location list generation tool.

3.2.2 Gate Library Characterisation

This section explains the tool flow step that determines the logic threshold voltages. The logic threshold voltage Th for an input is defined in Equation 3.4.

$$Th = V_{in},\ so\ that\ G\left(V_{in}\right) = \frac{Vdd}{2} \tag{3.4}$$

The logic threshold voltage Th is defined as the voltage applied on the input (V_{in}) of a gate so that the output voltage ($G(V_{in})$) is half-swing, i.e. half the supply voltage. This input voltage sets the gate just between driving high and driving low.

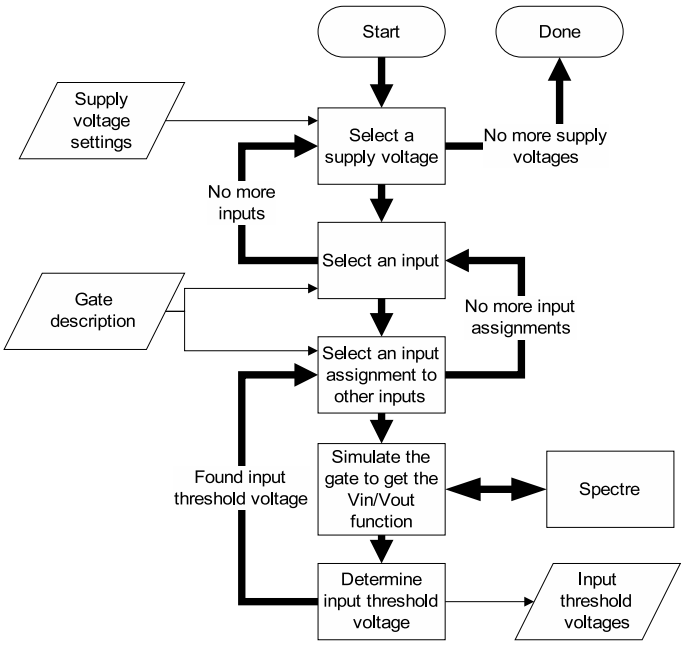


FIGURE 3.15: Gate library characterisation flow

Figure 3.15 shows the flow to generate the logic threshold voltages for a given gate. To characterise a gate library, the flow should be applied to each gate in the library. The input consists of the name of the gate to analyse, the gate library given in the form of Cadence Spectre transistor-level netlists (one for each gate, a.k.a. *subcircuits*) and the set of supply voltage settings. The gate description is taken from the gate library and the logic threshold voltage is determined for each input and supply voltage setting in an iterative process, where each iteration is trying a combination of assignments to the other inputs, i.e. the inputs that are not being characterised in a particular iteration are assigned logic values. This process continues for the characterised input for all possible assignments to the other inputs. The iteration over the combinations of assignments to the other inputs has two purposes. Firstly, to find assignments that allow propagation from the characterised input. Secondly, to find the highest and lowest logic threshold voltage, as this value varies with the assignment to the other inputs. The simulation step itself consists of performing sweeping Cadence Spectre simulations, varying the input

voltage until the output voltage is half swing (as required by Equation 3.4) and the logic threshold voltage is found.

Table 3.1 shows some statistics of the measured logic threshold voltages for a 0.12μm gate library from ST Semiconductors for three supply voltage settings, 1.2V, 1.0V and 0.8V. The left-most column shows the supply voltage setting. The next two columns show the lowest and highest logic threshold voltage encountered while processing the gate library. Within parenthesis are the logic threshold voltages relative to the supply voltage. The right-most column shows how this range compares with the supply voltage.

TABLE 3.1: Logic threshold voltage ranges for three supply voltages

Supply voltage	**Logic threshold voltage**		$\frac{Column2-Column3}{Column1}$
	Highest	**Lowest**	
1.2V	0.655V (55%)	0.472V (39%)	0.152
1.0V	0.536V (54%)	0.411V (41%)	0.125
0.8V	0.422V (53%)	0.348V (44%)	0.092

It can be seen from Table 3.1, the range of logic threshold voltages shrinks with reduced supply voltage. This range represents the ambiguity region, where an input voltage in the range may be interpreted as either Logic-1 or Logic-0 depending on the input and the gate that is driven. The range does not only shrink with supply voltage, it shrinks relative to the supply voltage. Another important observation from Table 3.1 is that column two and column three correspond to particular gate inputs (i.e. there was one gate input with consistently low logic threshold voltages and another gate input with consistently high logic threshold voltages) and that for these two different gate inputs, the logic threshold voltage scales differently with supply voltage. In total, 322 inputs from 85 gates were characterised when the SMuVoRBAT flow was applied in the experiments presented in this chapter.

3.2.3 Supply Voltage-Aware Resistive Bridging Fault Simulation

The fault simulator tool of the SMuVoRBAT tool suite is called Supply Voltage-Aware Resistive bridging Fault Simulator (SVARFS) and is used to evaluate tests, i.e. to determine the defect coverage they achieve, and used to aid the test generation tool MVTG. For simulating resistive bridging faults in a multi-voltage environment, there are two requirements that are not fulfilled in the standard off-the-shelf fault simulator tools. Firstly, the covered bridge defect resistance must be included in the defect coverage. This work uses the definition of defect coverage that is given in Equation 3.3. Secondly, the bridging fault site (Appendix A) needs to be simulated so that the supply voltage is considered. This is accomplished in the proposed fault simulator by employing a database of pre-calculated results from Cadence Spectre. The database stores the voltages on the bridged nets as a function of bridge resistance and supply voltage. The

database is accessed with a supply voltage, a pair of gates (corresponding to those driving the bridged nets) and the input assignments to the considered gates. The database returns the voltages for the bridged nets as a function of the bridge resistance as in the example of the V_{N1} curve in Figure 3.4. Using the database replaces CPU-intensive and time consuming Cadence Spectre simulation of the bridge fault site. The first time the SMuVoRBAT flow is used for a gate library, the database needs to be generated, which is indeed time consuming but is only required once for a gate library. When the database was prepared for the study of this chapter, it took nearly a week to perform all the simulations, using eight computers working in parallel. The same database is also used for test generation, which is discussed in Section 3.2.4.

Figure 3.16 shows the flow of the Supply Voltage-Aware Resistive bridging Fault Simulator (SVARFS). For each bridge location and supply voltage, the first step (marked "Logic fault generator" in 3.16) determines the corresponding logic faults. This involves getting the bridged net voltage as a function of the bridge resistance from the database (mentioned above). This function gives different results depending on the supply voltage specified for the simulated test set or test pattern. To determine the logic faults, the voltage function is compared to the logic threshold voltages of driven gate inputs. The comparison determines the corresponding defect resistance ranges and the logic behaviour, as in the example of Figure 3.4 where it can be seen how the logic behaviour is associated with resistance intervals depending on the resistance values that make the voltage on a bridged net the same as a logic threshold voltage. There are many variables to consider in the processing including supply voltage, bridge resistance and logic behaviour. To keep these variables in context, a set of tuples is defined for each bridge location b, such that each tuple consists of an LSC lsc that models faulty behaviour, a test pattern tp, a supply voltage setting v and a resistance interval ri. The tuple is called Logic Fault Tuple (LFT), with the following semantic: a test pattern tp that detects the faulty behaviour described by lsc covers the resistance interval ri when applied at supply voltage setting v. The semantics is illustrated in Figure 3.11, where there are three faulty logic behaviours LF1, LF2 and LF3. Assume that a test pattern called TP2 detects LF2. Detecting LF2 covers the interval $[R_{1A}, R_{2A}]$ for supply voltage VddA and $[R_{1B}, R_{2B}]$ for VddB and this gives two tuples $< lsc_{LF2}$,TP2,VDDA,$[R_{1A}, R_{2A}] >$ and $< lsc_{LF2}$,TP2,VDDB,$[R_{1B}, R_{2B}] >$. The fault simulation is performed by simulating a test pattern for two designs D and Df, where D is the original IC design and Df is the same design modified by the logic behaviour as specified by an LSC. Any discrepancy between the logic values on the outputs of D and Df mean that the test pattern has detected the faulty behaviour of the LSC. The implementation of a software tool for such simulation will be explained in detail below with regard to Figure 3.17. The fault simulation is performed for all test patterns in the test set and for all the LSCs corresponding to the set of LFTs that were identified previously in the method. The bridge resistances that are covered by the considered test are found by taking the union of the resistance intervals that belong to LFTs with detected LSCs. The amount of

covered bridge resistance (MVCADI, see Equation 3.2) is compared with the amount of detectable bridge resistance (MVGADI, see Equation 3.1) to calculate the defect coverage MVDC as given by Equation 3.3. The defect coverage is calculated both per supply voltage, as if the test sets were applied by themselves only, and for the complete test, including all supply voltages and their test sets. It should be noted that the fault simulator tool does not determine the amount of detectable bridge resistance MVGADI. Determining MVGADI, or GADI as it is known for single-voltage designs, is the main difficulty in fault simulation for resistive bridging faults [96]. To find MVGADI, it is required to determine if the logic behaviours represented by the LSCS are detectable or not. The typical method for determine the detectability of an LSC is to perform test generation for it. If the test generation produces a test pattern for the LSC, it is detectable, otherwise it is undetectable. Instead of performing test generation, the fault simulation tool uses MVGADI as provided by a previous run of the Multi-Voltage Test Generation tool MVTG, which is presented in Section 3.2.4.

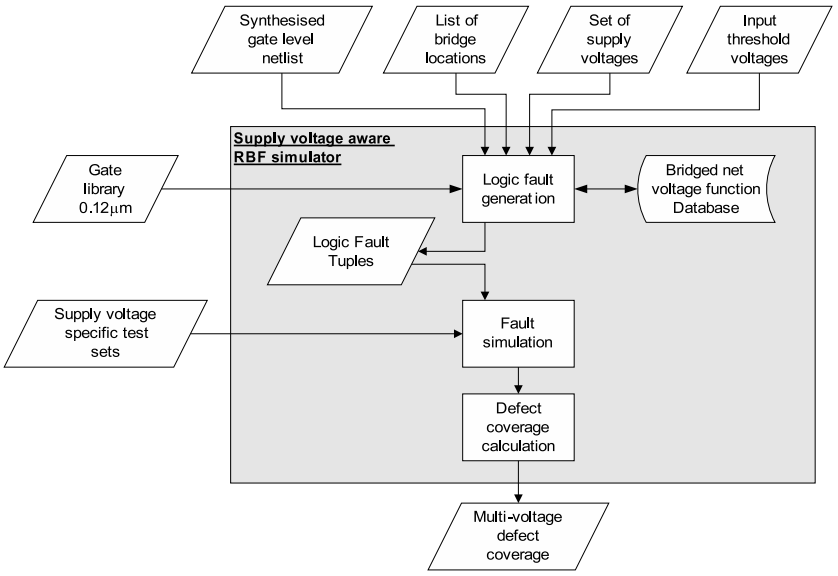


FIGURE 3.16: Detailed flow for the fault simulator SVARFS

To implement a fault simulation software tool for the step of Figure 3.16 that is marked fault simulation, the algorithm shown in Figure 3.17 is employed. The fault simulation algorithm takes a design D as a gate-level netlist, a bridge location b, a logic fault represented by an LSC lsc and a test pattern tp. The algorithm determines if the test pattern tp detects the logic fault that is represented by lsc. The first task for the fault simulation algorithm, lines 1-6, is to check if tp activates the bridge according to lsc. If tp does not activate the bridge according to the lsc, it cannot detect the lsc, which is why the algorithm can return **false** which means undetected on line 4. The FindLogicValue function is given in Figure 3.18 and further explained below. Then, line 7 makes a copy of the D netlist, performs fault insertion by modifying the copy with lsc and names the copy Df. Df is different from D by the logic values that are defined by lsc for the inputs that are driven by the bridged nets. The remaining steps of the fault simulation

FIGURE 3.17: Fault simulation algorithm used in SVARFS

Input: Netlist D,
Bridge location b
Logic State Configuration lsc
Test pattern tp
Output: **true** if detected, **false** otherwise

```
1:  for all net i that is an input to a gate that drives a bridged net do
2:    if FindLogicValue(D,i,tp) != L(lsc(i)) then
3:      // see Figure 3.18 for the FindLogicValue method
4:      return false
5:    end if
6:  end for
7:  construct netlist Df by copying D and inserting lsc at b
8:  // For the nets driven by the bridged nets in b, set the logic values that are specified
    in lsc
9:  OutputsD := The set of output nets for D
10: OutputsDf := The set of output nets for Df
11: while OutputsD is not empty do
12:   nD := a net from OutputsD
13:   remove nD from OutputsD
14:   nDf := the net in OutputsDf that corresponds to nD
15:   if FindLogicValue(D,nD,tp) != FindLogicValue(Df,nDf,tp) then
16:     return true
17:   end if
18: end while
19: return false
```

algorithm, lines 9-19, iterates over the outputs of the design and compares their logic values that are caused by tp and lsc. If a discrepancy is found, so that the logic value on an output net nD is different from the logic value on the corresponding output net nDf, the faulty behaviour of lsc is detected by tp. If all outputs are processed in this way without finding a discrepancy, the lsc is not detected by tp.

It should be noted that the function FindLogicValue is crucial to the algorithm in Figure 3.17 and is used on line 2 and on line 15. This FindLogicValue function is given in Figure 3.18. It takes a netlist NL, which can be D or Df, a net n in the netlist and the test pattern tp. FindLogicValue returns the logic value on net n as it depends on the test pattern tp. Also, FindLogicValue assigns logic values to the nets that it has already processed, which is a modification to the netlist NL. So in subsequent calls to FindLogicValue with the same netlist and the same test pattern, n does not have to be evaluated again; it already has a logic value. To get the logic value that has been assigned to a net x as described above, the function $L(x)$ is used. If net n has a logic value since previous calls to FindLogicValue, this logic value is returned on line 3. Furthermore, if net n is a primary input, its logic value is given by tp and returned on line 6. FindLogicValue is a recursive function, which means that to determine the logic value on a net n, it will first evaluate the logic values on the nets that drive the inputs to the gate g that drives

FIGURE 3.18: FindLogicValue method

Input: Netlist NL,
Net n
Test pattern tp
Output: the logic value on n caused by tp
Netlist NL is augmented with the logic value on n

```
1:  if n has a logic value then
2:     // This may happen if n is defined by an lsc or if n has already been processed in
       previous calls to FindLogicValue
3:     return L(n)
4:  else if n is a primary input of NL then
5:     L(n) := L(tp(n))
6:     return L(n)
7:  else
8:     g := the gate that drives n
9:     for all net i ∈ Ig, where Ig is the set of input nets for gate g do
10:       L(i) := FindLogicValue(NL,i,tp)
11:    end for
12:    L(n) := L(g(Ig))
13:    return L(n)
14: end if
```

n by calling itself (FindLogicValue) for each of those input nets. Once the logic values on the inputs of g are known (as the recursive calls to FindLogicValue returns), the logic function of gate g is used to determine the logic value on net n, which is returned on line 13. Through the recursive calls to FindLogicValue, it can proceed backward through the netlist, from the primary outputs (as FindLogicValue is used in the fault simulation algorithm Figure 3.17) to the primary inputs, where logic values are found in the test pattern tp and as the recursive calls return, the logic values are propagated through the netlist according to the logic functions of the gates until the logic values finally reach the primary outputs and the first call to FindLogicValue can return. Using the FindLogicValue function makes the worst case computational complexity of the fault simulation algorithm $\bigcirc(m)$, where m is the number of nets in the netlist. This is because FindLogicValue only evaluates each net once. When FindLogicValue has visited a net, that net will have a logic value and does not need to be evaluated again.

3.2.4 Multi-Voltage Test Generation

The proposed Multi-Voltage Test Generator (MVTG) produces supply voltage specific test sets for the given bridge list and the given circuit so that when the test sets are applied using their respective supply voltages, full defect coverage is achieved. An overview of the test generation method is given in Figure 3.19 and the algorithm for the method is given in Figure 3.21. It is shown in Figure 3.19 how the method starts by identifying the logic faults that can occur at each bridge location and for each logic fault, the method

creates a Logic Fault Tuple (LFT) as is described in Section 3.2.3.

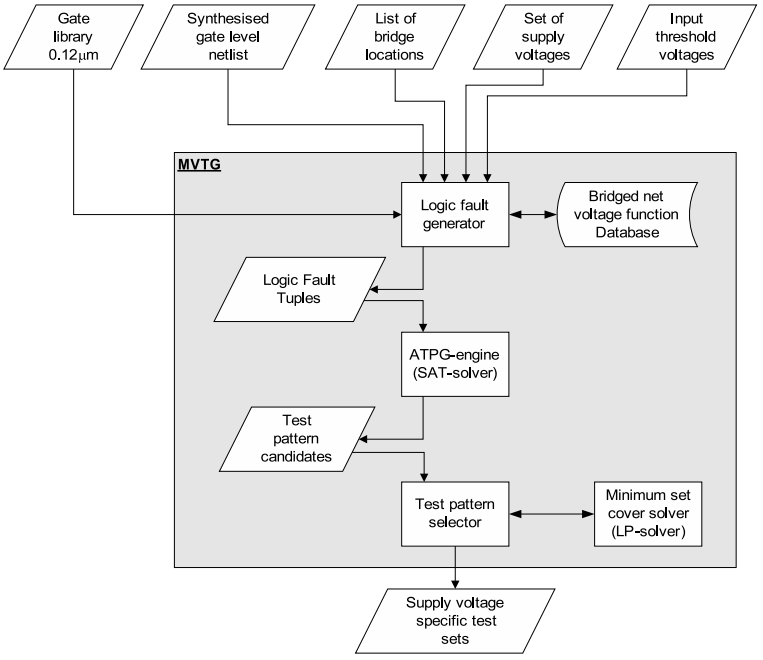


FIGURE 3.19: Detailed flow for the Multi-Voltage Test Generation method

To identify the logic faults and for each supply voltage setting and the corresponding resistance interval, the method employs a database, marked "Bridged net voltage function database" in Figure 3.19. This database is described in Section 3.2.3. The database supplies the voltage on the bridged nets as a function of the bridge resistance. The logic faults are identified by intersecting this voltage function is with the logic threshold voltages for the gate inputs that are driven by the bridged nets, as is demonstrated in the example of Figure 3.4. At this point in the processing, no test pattern has been generated, so the LFTs have the *tp* position of the tuple yet unassigned. The identified logic faults are processed by an ATPG-engine. In the presented implementation, the ATPG-engine consists of a Boolean Satisfiability Problem solver [44]. See Section 1.4.4 and Appendix B for an introduction on the Boolean Satisfiability Problem and how a solver can be used for ATPG. Other types of ATPG-engines would work as well, as long as it is possible to justify several logic values (to the fault site inputs) and to propagate one among several faulty signals (from the fault site outputs). The definition of fault site can be found in Appendix A. The output of the ATPG-engine is a set of candidate test patterns that are included in the appropriate LFTs. Only LFTs with detectable logic behaviour (LSC) get a test pattern, because the ATPG-engine determines if a logic fault is undetectable. So in the subsequent processing only detectable logic faults and their LFTs are considered. This means that the method can differentiate between CASE 1 and CASE 2 in Figure 3.11 and so determine that the resistance interval $[R_{1A}, R_{1B}]$ can only be detected using a test pattern for LF2 applied at voltage VddA. Thus, the supply voltage to use while testing for a given resistance interval is recorded in the supply voltage specific resistance intervals that are associated with the detectable logic

S1= $[R_{1A}, R_{1B}]$
S2= $[R_{1B}, R_{2A}]$
S3= $[R_{2A}, R_{2B}]$
S4= $[R_{2B}, R_{3A}]$
S5= $[R_{3A}, R_{3B}]$

(a) Sections

x2A=1 if TP2* is applied at VDDA, otherwise 0
x3A=1 if TP3** is applied at VDDA, otherwise 0
x2B=1 if TP2 is applied at VDDB, otherwise 0
x3B=1 if TP3 is applied at VDDB, otherwise 0
* TP2 is a test pattern for LF2
** TP3 is a test pattern for LF3

(b) Test pattern variables

minimise x2A+x3A+x2B+x3B
S1: x2A >= 1
S2: x2A+x2B >= 1
S3: x3A+x2B >= 1
S4: x3A+x3B >= 1
S5: x3B >= 1

(c) LP problem formulation

FIGURE 3.20: LP problem formulation used to select the minimum set of test patterns to cover the full range of detectable bridge resistance

faults. It should be noted that the candidate test patterns that are generated by the ATPG-engine achieve full defect coverage by finding all detectable logic faults but may cover overlapping resistance intervals, such that only a subset of the candidate test patterns are required to cover all detectable defect resistance (the scope of MVGADI). To determine such a subset of candidate test patterns from the candidate test patterns, the test pattern selector in Figure 3.19 determines the minimal set of test patterns to include in the final test set and the appropriate supply voltage setting for them so that all detectable defect resistance is covered. The selection problem is mapped to the Linear Programming (LP) domain by considering it a minimum set cover problem, which is solved by an LP-solver [156]. The scenario of CASE 2 in Figure 3.11 is used to illustrate the minimum set cover problem formulation.

Similar to the sectioning approach in [110], the resistance intervals are divided into disjunct sections S1, S2, S3, S4 and S5 as described in Figure 3.20(a), and all these sections should be covered by the final test, which provides the constraints shown in Figure 3.20(c). Variables are defined for the test patterns and their supply voltage settings in Figure 3.20(b). Using the sections and the variables for the test patterns, the LP-problem is given in Figure 3.20(c). The task for the LP-solver is to minimise the number of test patterns while keeping the constraints that arise from the requirement that all sections should be covered. From section S1 and section S5 it can be seen that x2A and x3B must be assigned 1 to keep the constraints. This leaves S3 not-yet-covered, which means that either x3A or x2B must be assigned 1 as well. From the above it can be concluded that the test sets {TP2} applied at VDDA and {TP2, TP3} applied at VDDB are minimal and achieve full defect coverage for CASE 2 in Figure 3.11.

The Multi-Voltage Test Generation algorithm is given in Figure 3.21. The algorithm starts by identifying for each bridge the resistance intervals that correspond to faulty logic behaviour (lines 1-12). This is identification process is performed for each supply

voltage setting (lines 3-10) and is achieved by comparing the voltages on the bridged nets with the logic threshold voltage the driven gate inputs. The voltages on the bridged nets are acquired from the database (marked "Bridged net voltage function database" in Figure 3.19). All input assignments to the gates that drive the bridged nets are considered to identify all the possible faulty logic behaviours and the corresponding resistance intervals. The faulty logic behaviours are captured in LSCs (as discussed in Section 3.1.2) and associated with supply voltage specific resistance intervals according to the definition of Logic Fault Tuple (LFT) above. The algorithm continues considering one bridge b at a time (lines 13-14). The logic behaviour of each LFT belonging to the considered bridge b is targeted by test pattern generation (line 18). Only LFTs that are not yet covered in terms of the resistance interval are targeted by this test pattern generation. All generated test patterns are included in the appropriate LFTs, such that the test pattern detects the logic behaviour of the LFT. It should be noted that each LFT only holds one test pattern, which is the first test pattern that is found to detect the LFT in the processing of the method. Some LFTs are undetectable as no test pattern exists that can detect the logic behaviour of the LFT. Only LFTs for which the ATPG-engine could find a test pattern are considered and these LFTs are added to $covered\_LFT(b)$ on line 21. When the processing has reached line 24, $covered\_LFT(b)$ consists of all the detectable LFTs for bridge b, each with a test pattern. The resulting set of test patterns (that is represented by the LFTs in $covered\_LFT(b)$) is a test set that fully covers all detectable bridge resistance for bridge location b. To cover all detectable bridge resistance is to cover the whole scope of MVGADI (Equation 3.1) and therefore such a test set achieves full multi-voltage defect coverage (Equation 3.3). However this test set comprises multiple possible ways of covering the detectable resistance intervals corresponding to the bridge b. To select as few of these test patterns as possible, while still covering all detectable bridge resistance for bridge b, a minimum set cover problem is solved in select_TPs_and_Vs on line 25. An example of how a set cover problem is formulated is given in Figure 3.20. The result $Selected\_TPs\_and\_Vs$ is a minimum set of pairs of voltage settings and test patterns, so that the test patterns covers all detectable resistance for bridge b when applied according to the corresponding voltage settings. Each selected test pattern and supply voltage pair $< tp, v >$ in $Selected\_TPs\_and\_Vs$ are added to the final supply voltage specific test sets $TestSet(v)$. The test patterns are fault simulated for the corresponding supply voltage settings using all remaining bridges. In this process, the resistance intervals corresponding to detected LFTs are marked as covered, so they are not targeted anymore in the test generation for remaining bridges (lines 26-32).

It should be noted that MVTG determines the MVGADI (Equation 3.1), which is needed for in the fault simulation tool (Section 3.2.3) to determine the defect coverage. Therefore, it is required to run MVTG before calculating the defect coverage of any test set. However, MVGADI is the set of detectable bridge resistance and therefore not dependent on the test set. On the other hand, the MVGADI generated by MVTG depends

FIGURE 3.21: The Multi-Voltage Test Generation method (MVTG)

Input: Netlist,
Supply voltage settings $V = \{v1, v2, ...\}$,
Bridges $B = \{b1, b2, ...\}$
Logic threshold voltages LTV
Output: Supply voltage-specific test sets with 100% defect coverage, $TestSet(v)$, $\forall v \in V$

1: **for all** Bridge $b \in B$ **do**
2: **for all** input assignments ia to the driving gates of b **do**
3: **for all** $v \in V$ **do**
4: get the bridge net voltage function $F(b, ia, v, r)$ from database
5: compute critical resistances by comparing $F(b, ia, v, r)$ with $LTV(b)$
6: identify resistance intervals RI and $\forall ri \in RI$, the corresponding LSCs lsc
7: **for all** $ri \in RI$ **do**
8: $LFT(b) := LFT(b) \bigcup \{(lsc, v, ri)\}$
9: **end for**
10: **end for**
11: **end for**
12: **end for**
13: **while** $(B \neq \emptyset)$ **do**
14: Get a bridge b from B
15: Remove b from B
16: **for all** $lft \in LFT(b)$ **do**
17: **if** ($LSC(lft)$ not yet covered) **then**
18: ATPG$(b, LSC(lft))$
19: **if** (ATPG found a test pattern tp for $LSC(lft)$) **then**
20: $TP(lft) := tp$
21: mark $RI(lft)$ as covered by adding lft to $covered\_LFT(b)$
22: **end if**
23: **end if**
24: **end for**
25: $Selected\_TPs\_and\_Vs :=$ select_TPs_and_Vs $(covered\_LFT(b))$
26: **for all** $< tp, v > \in Selected\_TPs\_and\_Vs$ **do**
27: $TestSet(v) := TestSet(v) \bigcup \{tp\}$
28: **for all** $b \in B$ **do**
29: fault simulate b with tp
30: add any detected $lft \in LFT(b)$ to $covered\_LFT(b)$
31: **end for**
32: **end for**
33: **end while**

on the considered supply voltage settings.

3.2.5 Test Set Post-Processing

An optional post-processing step can be employed to further reduce the size of the supply voltage-specific test set size obtained as described in Section 3.2.4. The flow is

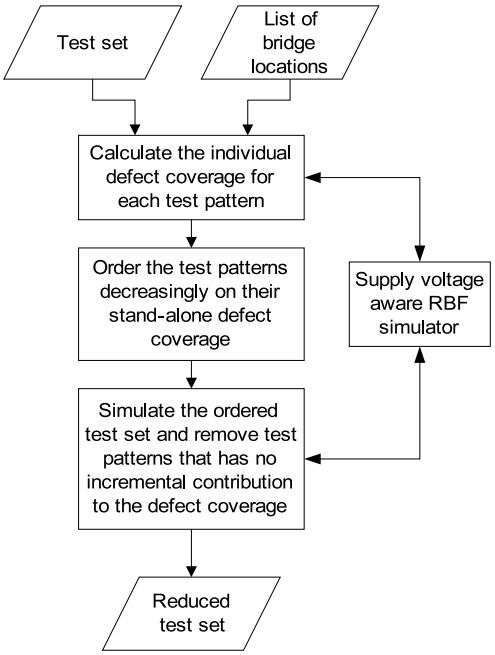


FIGURE 3.22: Test set post-processing flow for test set size reduction

described in Figure 3.22. Initially, each test pattern is fault simulated by employing the algorithm in Figure 3.17 using the entire bridge list to compute individual defect coverage of the test pattern. The test set is then sorted in descending order of the test pattern defect coverage. The test patterns in the ordered test set are fault simulated again, this time while marking resistance intervals as they are detected. If no resistance interval was marked for a particular test pattern tp because all the intervals covered by tp had already been marked by previous test pattern, tp is removed from the test set as it does not contribute to the overall defect coverage. The computational complexity of the test set post-processing step is $\bigcirc(m \cdot t \cdot b \cdot l)$, where m is the number of nets in the design, t is the number of test patterns before the post-processing step, b is the number of bridge locations and l is the worst case number of logic faults that can occur at a bridge location.

3.3 Experimental Results

The proposed Multi-Voltage Test Generation method MVTG has been implemented as part of a tool suite as described in Section 3.2 and has been validated experimentally using a number of ISCAS85 and ISCAS89 benchmark circuits. The sequential circuits were treated as combinatorial by assuming full scan-chains and only non-feedback bridges have been targeted. The benchmark circuits were synthesised using the ST Microelectronics gate library for 0.12μm technology where the nominal supply voltage is 1.2V. Three supply voltage settings were used during the experiment, 0.8V, 1.0V and 1.2V.

3.3.1 Considering a Bridge Resistance Occurrence Distribution

Although this work considers equal probabilities for defects with different resistance values, the real life occurrence distribution of bridge resistance may show that some resistance values are unlikely to be found on a fabricated circuit. Therefore, targeting defects with such resistance values would not lead to a real contribution in terms of test quality. The impact of a real life distribution of bridge resistance values is addressed in Figure 3.23. It shows the histogram for the distribution of defects which cannot be detected at 0.8V supply voltage. Here it is assumed that testing will be performed at 0.8V supply voltage, in a system where the nominal supply voltage is 1.2V, because of the observed trend that resistive bridge defects are better detectable at a low supply voltage [112, 114]. The experiment aims to determine if exceptions to this trend, as reported in [112, 114, 115], belong to a limited range of bridge resistance. The distribution in Figure 3.23 is based on experiments using seven of the medium and large size ISCAS benchmarks. The experiment is performed by applying the test generation method in Section 3.2.4 for only 0.8V supply voltage and then the resulting test sets are fault simulated using the fault simulator described in Section 3.2.3 and the defect coverage metric in Equation 3.3. This procedure is used to identify the amount of bridge resistance that is not detected by the test set, but could be detected if test was performed using the appropriate supply voltage. The random spread of these defects across the resistance range suggests that testing with only 0.8V supply voltage will not be sufficient, even if the real-life defect occurrence distribution of a particular manufacturing process is concentrated around a certain resistance range. This range would also contain defects that cannot be detected using 0.8V supply voltage. For example, if the real life occurrence distribution of bridge resistance was restricted to resistances in $[0\Omega, 2k\Omega]$, testing would still be required for at least another supply voltage setting as indicated by the bars for this range of bridge resistance. The bars are not at 0%, which shows that some bridge defects cannot be detected using 0.8V supply voltage. Therefore, testing using more than one supply voltage is required to achieve high test quality.

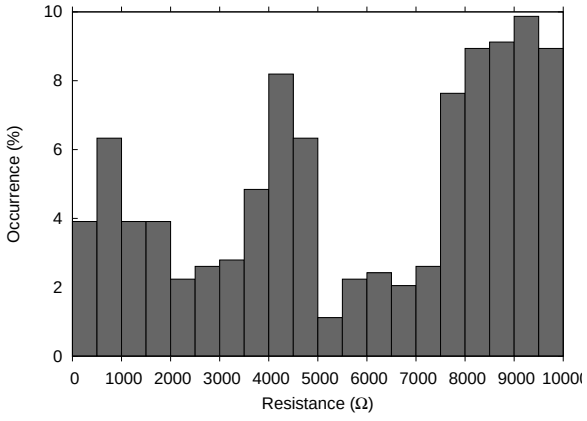


FIGURE 3.23: The distribution of resistance values that cannot be detected at 0.8V supply voltage

3.3.2 Multi-Voltage Test Generation Results

Table 3.2 and Table 3.4 show the test-set sizes generated and the CPU time for the algorithm in Figure 3.21 and the optional test set post-processing step respectively. Table 3.2 shows the results of running the Multi-Voltage Test Generation method (Figure 3.21) to generate supply voltage specific test sets. The left-most column shows the benchmark circuits, where an initial "C" means that the circuit is combinatorial and an "S" means that the circuit is sequential. The second column from the left shows for each design, how many non-feedback bridges have been identified from the circuit layout. The next three main columns, marked with "Vdd 0.8V", "Vdd 1.0V" and "Vdd 1.2V", show the test pattern count in the corresponding test-set, TS1, TS2 and TS3 respectively. The sixth column, marked "Sum #tp" shows the total number of test-patterns necessary to achieve 100% defect coverage (Equation 3.3). The column that is marked with "CPU time" shows the CPU time required to achieve these results, in seconds. The CPU time should be understood in context of the computer used to perform the experiment. The experiment was performed on an Opteron CPU with 8GB of RAM running the Red Hat Linux operating system. The method uses a solver for the Boolean Satisfiability problem (SAT) as ATPG-engine [44] (Appendix B). The second last column shows the fraction of time spent inside the ATPG-engine (line 18 of Figure 3.21). This fraction can be significantly reduced if a more efficient commercial ATPG-engine is available. The right-most column gives the average number of fault-site simulations per bridge location. These simulations would have been performed using Cadence Spectre if the method was not using a pre-compiled database of bridge simulation data. The number of simulations depends on the bridge locations, as some bridge locations lead to many possible logic behaviours, depending on the number of inputs to the gates that drive the bridged nets and the fanout of the bridged nets. The gate library used in the experiments has gates with one to nine inputs. The fault-sites are also simulated for each supply voltage setting.

As can be seen from Table 3.2, the Multi-Voltage Test Generation method MVTG generates test sets for all the three supply voltage settings, with the majority of test patterns in the test set TS1 for 0.8V which is the lowest supply voltage setting. The fact that the lowest supply voltage setting is used so much by the method is in-line with previous studies [112, 114] and observations that testing for resistive bridge defects is more effective with a lowered supply voltage. The test sets for the two remaining supply voltage settings, TS2 and TS3 (1.0V and 1.2V supply voltage respectively), also contain test patterns for many of the benchmarks. These test patterns, in TS2 and TS3, detect defects that could not be detected using 0.8V supply voltage.

Table 3.3 shows the defect coverage achieved by the test-sets that are defined in Table 3.2, as calculated using the supply voltage aware RBF simulator. Column two until five show the incremental defect coverage of applying the test-sets in order from lowest voltage

TABLE 3.2: Results of Multi-Voltage Test Generation

		Vdd 0.8V	**Vdd 1.0V**	**Vdd 1.2V**				
Design	**Bridge locations**	**#tp TS1**	**#tp TS2**	**#tp TS3**	**Sum #tp**	**CPU time**	**ATPG %**	**Simulations per bridge location**
C1355	80	39	0	0	39	21	71	557
C1908	98	57	0	0	57	13	58	296
C2670	104	67	0	0	67	27	41	269
C3540	363	184	6	1	191	340	62	568
C7552	577	281	0	1	282	1049	66	552
S838	34	26	2	0	28	6	54	243
S1488	435	144	2	0	146	193	10	265
S5378	305	214	0	0	214	308	32	914
S9234	223	132	2	0	134	130	43	1068
S13207	358	192	5	1	198	2454	53	1291
S15850	943	324	4	5	333	11835	42	417
S35932	1170	547	50	63	660	11233	53	263

setting to highest voltage setting. This defect coverage is defined in Equation 3.3. Column two is the defect coverage of only applying TS1 (for 0.8V supply voltage). Column three is the defect coverage achieved by applying TS1 at 0.8V and TS2 at 1.0V. It should be noted that, if the number of test patterns in TS2 is zero in Table 3.2, the result is the same as applying TS1 at 0.8V (Column two). In the same way, column four is the defect coverage of TS1 and TS3 (where TS3 is applied at 1.2V) and if TS3 has zero test patterns, the result is the same as in Column two. The last column shows the defect coverage achieved by applying all test-sets at their respective supply voltage settings.

The observation of 100% defect coverage in Table 3.3 was made empirically by fault simulating the test sets in Table 3.2 with respect to the set of bridge defects that can be detected by logic testing over the given set of supply voltage settings. It should be noted that the defect coverage defined in Equation 3.3 only includes bridge resistances that are covered by detectable logic faults. As can be seen from Table 3.3, for some circuits, 100% defect coverage can be achieved using a single supply voltage during test (Column two). However, for other circuits, such as S35932, achieving full defect coverage requires testing at more than one supply voltage setting.

3.3.3 Test Set Size Reduction

The test sets in Table 3.2 have not been compacted or compressed. The fact that the Multi-Voltage Test Generation algorithm performs fault simulation of every test pattern that is added to the final test sets and avoids targeting resistance intervals that have

TABLE 3.3: Defect coverage for the supply voltage-specific test sets from MVTG

	TS 1	**TS 1&2**	**TS 1&3**	**TS 1&2&3**
Design	**defect coverage**	**defect coverage**	**defect coverage**	**defect coverage**
C1355	100.0	100.0	100.0	100.0
C1908	100.0	100.0	100.0	100.0
C2670	100.0	100.0	100.0	100.0
C3540	99.20	99.96	99.20	100.0
C7552	99.95	99.95	100.0	100.0
S838	95.04	100.0	95.04	100.0
S1488	99.98	100.0	99.98	100.0
S5378	100.0	100.0	100.0	100.0
S9234	99.87	100.0	99.87	100.0
S13207	99.57	99.92	99.57	100.0
S15850	99.84	99.94	99.84	100.0
S35932	93.95	98.48	93.95	100.0

already been covered by previously added test patterns does help to keep the test set size small, but there is still room for improvement. Table 3.4 shows the results of applying the optional post processing step to the test-sets in Table 3.2. The columns that are marked with # show the final number of test patterns in the test set for the respective supply voltage settings. The complimentary %-columns give the relative reduction in test patterns in the respective test sets. So for circuit S15850, the outcome of the post-processing was 235 test patterns for supply voltage 0.8V, which is 27% less than before the post-processing (Table 3.2, column 3). In the fifth main column is the relative reduction in the total number of test patterns. The last column shows the CPU time in seconds. So for circuit S15850, it took 1910s to conclude the post-processing, which is ≈32 minutes. Table 3.4 demonstrates that it is possible to achieve up to 27% reduction in test set size at the expense of increased CPU time.

3.3.4 Integrated Flow with TetraMAX

To see how the proposed test generation method work compared to the off-the-shelf test generation tools, an experiment was made using Synopsys TetraMAX and the Multi-Voltage Test Generation (Section 3.2.4) as a combined test generation flow. The comparison is performed using the fault simulation tool (Section 3.2.3). First, a test-set targeting bridging faults is generated with TetraMAX, using the same bridge list as in the experiment of Table 3.2. Then the TetraMAX test-set was fault simulated at supply voltage 0.8V (since higher resistive bridging defect coverage is achieved at a lower supply voltage [114]) using the supply voltage aware RBF simulator. The defect coverage achieved and the number of test patterns in the TetraMAX test-set is given in the

TABLE 3.4: Reduced test-set sizes using the post-processing step on the test sets from Table 3.2

Design	**0.8V**		**1.0V**		**1.2V**		**Tot.**		**CPU time**
	#	%	#	%	#	%	#	%	(s)
C1355	32	18	*	*	*	*	32	18	529
C1908	47	18	*	*	*	*	47	18	704
C2670	57	15	*	*	*	*	57	15	1057
C3540	151	18	6	0	1	0	158	17	2380
C7552	229	19	*	*	1	0	230	18	2915
S838	22	15	2	0	*	*	24	14	163
S1488	121	16	2	0	*	*	123	16	1611
S5378	175	18	*	*	*	*	175	18	5347
S9234	109	17	2	0	*	*	111	17	4165
S13207	158	18	3	40	1	0	162	18	4415
S15850	235	27	4	0	3	40	242	27	1910
S35932	459	16	43	14	51	19	553	16	2367

* not applicable as the original test set for the Vdd was empty

second main column of Table 3.5. Please note in Table 3.5, that Synopsys TetraMAX generates test-sets that may yield defect coverage as low as 83% at supply voltage 0.8V. TetraMAX uses the 4-way model of bridging faults, discussed in Section 2.1. This model is combined with a scheme to drive one of the bridged nodes with maximum strength and the other node with minimum strength by controlling the inputs to the gates that drive the bridged nets. This way, the likelihood of detecting a bridge defect is increased, even though TetraMAX does not take the defect resistance value into account. The fact that the defect coverage of resistive bridging faults is so low with TetraMAX is in itself a motivation for using MVTG to add the test patterns that are needed to achieve full defect coverage. Subsequently, MVTG is used on the bridges that were not fully covered by the TetraMAX test-set, to supply the remaining defect coverage up to 100%. The sizes of the test sets generated by the MVTG top-up run are given in the third column for each supply voltage setting. The fourth column of Table 3.5, marked "Tot." shows the total test pattern count. The last column is the CPU time (in seconds) for simulating the TetraMAX test-set and running MVTG to top-up the test-set. The results show that the combined test generation flow, of TetraMAX and MVTG, in comparison with Table 3.2 and Table 3.4 sometimes produces smaller test sets, and also sometimes has a reduced CPU time. This is however not always the case as can be seen by design S9234, which has 142 test patterns in the combined flow and 134 test patterns in Table 3.2, and as can be seen by design C2670 where it takes 36s to produce the test sets in the combined flow which should be compared to 27s in Table 3.2. The results in Table 3.5 show that the Multi-Voltage Test Generation method can be integrated with existing tool flows.

TABLE 3.5: Results of using TetraMAX and MVTG as a combined test generation flow

	TMAX		**MVTG top-up**				
	0.8V		**0.8V**	**1.0V**	**1.2V**	**Tot.**	**CPU**
Design	**MVDC**	**#tp**	**#tp**	**#tp**	**#tp**	**#tp**	**time (s)**
C1355	83	33	32	0	0	65	18
C1908	98	42	27	0	0	69	11
C2670	90	27	50	0	0	77	36
C3540	96	72	126	6	1	205	239
C7552	95	44	198	0	1	243	789
S838	88	17	17	2	0	36	2
S1488	96	82	82	2	0	166	123
S5378	95	60	123	0	0	183	214
S9234	89	48	92	2	0	142	105
S13207	95	60	89	5	1	155	1625
S15850	98	56	144	4	5	209	1954
S35932	96	33	89	36	66	224	11511

3.3.5 Summary

From the experimental results reported in this chapter, several observations can be made:

- As a proof of concept Table 3.2 and Table 3.3 shows that to achieve full defect coverage (with respect to Equation 3.3) of logic-testable resistive bridging faults in a circuit that operates using multiple supply voltage settings, test patterns may have to be applied at more than one supply voltage setting.
- The general trend has been observed in previous reports [114] and is here observed again (Table 3.2), that most resistive bridging faults are detectable at the lowest available supply voltage setting. Exceptions have also been observed, in line with [112, 114, 115].
- The proposed Multi-Voltage Test Generation algorithm (Section 3.2.4) is effective in covering all the detectable defects for a set of supply voltage settings, however there is still room for improvement in terms of test set size and CPU time. A method for reducing the test set size (Section 3.2.5) was demonstrated in Table 3.4.
- The proposed Multi-Voltage Test Generation method (Section 3.2.4) can be combined with other test generation tools such as those available commercially as was demonstrated in Table 3.5. This is achieved by using the supply voltage aware fault simulator tool (Section 3.2.3) to identify bridge defect resistance ranges that have not been covered by the test set from the commercial test generation tool and target the identified bridge defects using the MVTG method.

3.4 Concluding Remarks

Modern low-power designs employ multiple supply voltages to implement power modes (Section 1.1), which provides a challenge for manufacturing testing in terms of defects with supply voltage dependent behaviour. The challenge is to generate test patterns and apply them using the supply voltage for which they are effective in detecting the defects. No previous work has addressed such test generation for multi-voltage designs. An important defect type with supply voltage dependent behaviour is the resistive bridge defect. Previous work on resistive bridge defects has shown that such defects are generally better detectable using a lowered supply voltage [112, 113, 114, 109, 116], but exceptions to this trend has been found, where testing using a particular supply voltage is required to cover the defect. Indeed, because of the bridge defects that require testing using particular supply voltage settings, this chapter has demonstrated that designs which employ multiple supply voltages should be tested using more than one supply voltage to achieve full defect coverage of resistive bridges. This result was produced by analysis of the impact of different supply voltage values on the behaviour of resistive bridge defects. To conduct such analysis, this chapter has extended a fault model for resistive bridge defects [104] to fit a multi-voltage scenario. Several examples provided in the chapter showed how resistive bridging defects lead to supply voltage dependent behaviour and the detection of such defects was discussed.

To test for resistive bridging faults in designs with multiple supply voltages, this chapter has shown how supply voltage-specific test sets can be generated. The proposed solution is called the Multi-Voltage Test Generation method (Section 3.2.4) which improves on previous test generation methods [109, 110, 111] by considering multiple supply voltage settings and delivers full defect coverage for resistive bridges. With regard to the method, a software suite called SMuVoRBAT was presented, including:

- Identification of realistic bridge locations from circuit layout (Section 3.2.1).
- Supply voltage characterisation of library gates (Section 3.2.2).
- A multi-voltage aware fault simulator for resistive bridging faults (Section 3.2.3).
- A multi-voltage test generation method for resistive bridging faults (Section 3.2.4).
- A test-set size reduction procedure (Section 3.2.5).

Experimental results on synthesised and place-and-routed ISCAS benchmark designs using the software suit showed that full defect coverage was achieved for resistive bridging faults over a set of supply voltage settings, providing a low-cost and effective test solution. The experiments included a study on how the proposed test generation method can be integrated in a commercial test generation flow.

Chapter 4

Testing for Full Open Defects under Supply Voltage Variation

As it can be seen from the literature review in Section 2.2 and Section 2.3.2, opens represent a major class of defects for deep submicron CMOS and can be divided into two classes: *resistive opens* and *full opens*. It is known that resistive opens cause additional transition delay that depends on the supply voltage. Therefore, recent research [79,134] concluded that resistive opens are in general better detected at elevated supply voltage (i.e. higher than the nominal supply voltage level). Full open defects, on the other hand, have not been addressed explicitly in [79, 134]. Full opens (complete break between nodes that should be connected) happen in practise [69,72,70,71] and it has been shown that full open defects on interconnect cause supply voltage-dependent behaviour [123]. The supply voltage-dependency raises the question of how supply voltage-dependent behaviour impacts the effectiveness of testing designs that employ more than one supply voltage, as is the case in many low power designs (Section 1.1). Therefore, this chapter provides an analysis on the impact of varying supply voltage on the detectability of full opens. To perform such analysis, this chapter presents a study of full opens based on two models. The first model describes the defect behaviour depending on the influence of capacitive coupling from circuitry physically close to the defect (Section 2.2.3) and the second model describes the defect behaviour depending on gate tunnelling leakage (Section 2.2.2). Both models describe the voltage on a net that has been separated from its driver by a full open. This voltage is then compared to the logic threshold voltage for gate inputs that are driven by the affected net to determine the logic behaviour.

In this chapter, the mechanism behind supply voltage-dependent behaviour of full open defects is studied, to complement the observations made from the literature review in Section 2.3.2, that test results for full opens have been reported not to be influenced by supply voltage [70], and that some full open show supply voltage-dependent behaviour [123]. To analyse and quantify the impact of supply voltage variation on testing

full open defects, a supply voltage aware fault simulator is developed. With regard to such fault simulation, the study explores the factors that cause supply voltage-dependent detectability of full open defects. Supply voltage-dependent detectability depends on the voltage on the net that is affected by the defect and the surrounding circuitry. To take both these factors into account, the experiments presented in this chapter consider realistic voltage values and synthesised benchmark designs. Experimental results are presented for both fault-site analysis and from the wider perspective of defect detection taking the whole design into account. The experiments were performed on ISCAS85 and -89 benchmark circuits that were synthesised using a 0.12μm gate library and then placed-and-routed so that coupling capacitance data, necessary for modelling and simulation, could be extracted from layout.

The chapter is structured as follows. Section 4.1 provides background information on full open defects and a brief review of relevant literature. In Section 4.2 and Section 4.3 two models for the behaviour of full open defects are discussed, these models consider capacitive coupling and gate tunnelling leakage respectively. A fault simulation tool for analysing the supply voltage-dependent behaviour of full open defects is presented in Section 4.4 and experimental results are given in Section 4.5. Concluding remarks are given in Section 4.6.

4.1 Analysis and Statement of Problem

This section gives a more detailed background review on full open defects than was given in Chapter 2. This is done to put the background into context of the study conducted in this chapter and to make the chapter self-contained.

Full open defects are complete breaks between circuit nodes that should be connected, as in the example of Figure 4.1. As can be seen from Section 1.4.9 and Section 2.2, relevant research on full open defects have considered the defect location [85, 118, 72] and found that most full open defects occur on the interconnects [80]. Other relevant research has addressed fault modelling. Several fault models have abstracted from the voltage on the net that is affected by the open [97, 129, 122, 82] and therefore, they are not useful in studying the supply voltage-dependent behaviour of full open defects, which is the focus in this chapter. There has also been research on fault modelling based on the voltage on the node that is separated from its driver by the full open defect [127], with regard to two complementary mechanisms that influence this voltage, namely capacitive coupling to neighbouring nodes [121, 82, 122, 126, 127] and gate tunnelling leakage [83, 84, 29]. This chapter will study the supply voltage-dependent behaviour of full open defects with regard to these two mechanisms.

Full opens have not been analysed in detail by previous studies in terms of supply voltage-dependent behaviour. Full opens show supply voltage-dependent behaviour according

to [123] but according to [70], varying the supply voltage has no influence on the test results. This chapter will study the behaviour of full open defects in detail to determine if detectability of such defects depends on the supply voltage, which would have impact on how tests should be performed.

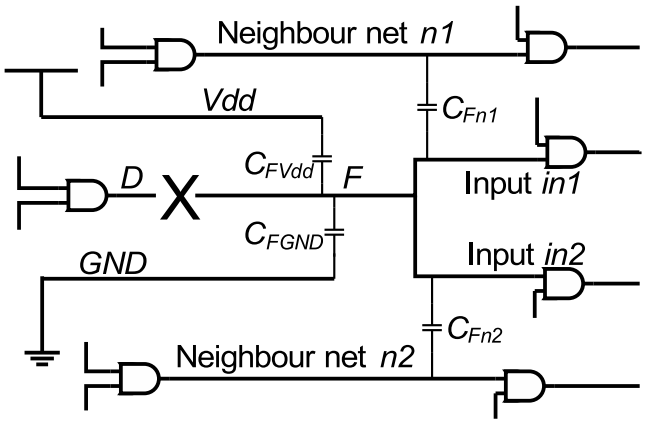
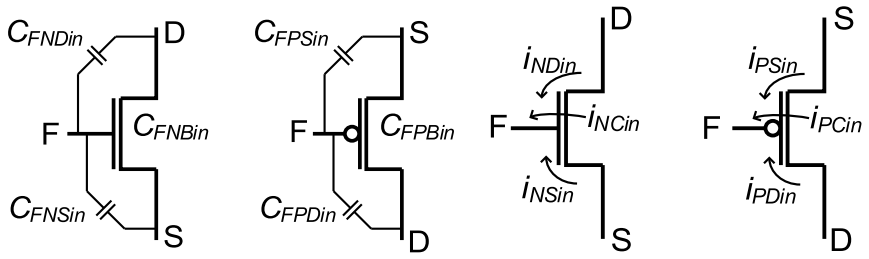


FIGURE 4.1: A full open defect

An example full open defect is shown in Figure 4.1. Net F is separated from its driver, net D, by a full-open defect marked with X. Net F is called the victim net. The original net, including both net D and net F, is called the net-under-test. Defect location refers to the location of the defect on the net-under-test. Two inputs, Input $in1$ and Input $in2$, are driven by net F. This means that Input $in1$ and Input $in2$ may see faulty behaviour due to the defect. For a general defect, the inputs that are driven by the victim net define a set IN. Similarly, the neighbouring nets define a set N. The neighbouring nets are those that have a significant capacitive coupling to the victim net. For the case shown in Figure 4.1, the set N consists of net n1 and net n2, and the capacitances C_{Fn1} and C_{Fn2} represent the corresponding capacitive coupling. The victim net is also capacitively coupled to ground and supply voltage (Vdd) as shown by the capacitances C_{FGND} and C_{FVdd}. The capacitances are really distributed along the length of the victim net, but are here shown as lumped capacitances. The victim net (net F), the driver (net D), the inputs driven by net F (the IN set) and the neighbouring nets (the N set) together form the fault site for the full open defect. See Appendix A for the definition of the concept of a fault site.

Figure 4.2 shows how the victim net can be influenced by the voltage on transistor nodes of driven inputs (the inputs in set IN) by capacitive coupling [123, 80, 121] or by gate tunnelling leakage [84, 29]. For each input $in \in IN$ there are capacitances from the victim net F to the source, drain and bulk nodes for the corresponding NMOS and PMOS transistors. This is shown in Figure 4.2(a) by the capacitances C_{FNSin}, C_{FNDin}, C_{FNBin}, C_{FPSin}, C_{FPDin} and C_{FPBin}. Furthermore, gate tunnelling leakage currents to or from the transistor nodes of driven inputs can affect the victim net. Such leakage current has three components for each driven transistor as shown by the arrows labelled I_{NSin}, I_{NDin}, I_{NCin}, I_{PSin}, I_{PDin} and I_{PCin} in Figure 4.2(b). The directions of the



(a) Capacitances to transistor nodes (b) Gate tunnelling leakage currents

FIGURE 4.2: Influence from input $in \in IN$ on the victim net F

arrows determine the sign associated with the currents so that current in the direction of the arrow is positive (flowing onto the victim net) and current in the opposite direction is negative (flowing from the victim net). The influence of capacitive coupling on the voltage on the victim net is discussed in Section 4.2 and the corresponding influence of gate tunnelling leakage currents is discussed in Section 4.3.

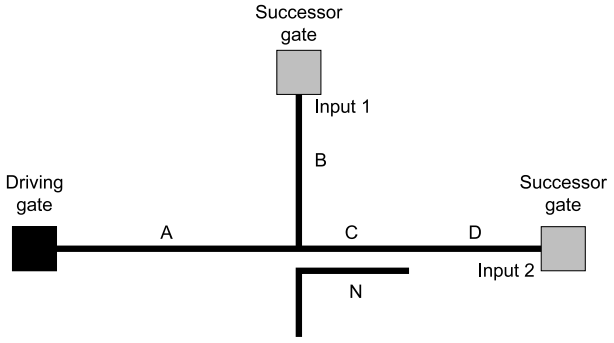


FIGURE 4.3: Net segmentation representing possible defect locations

The review of research on full open defects Section 2.2 has addressed the importance of the defect location on the considered net. It was found that all vias and contacts on the net are likely locations for open defects and that different defect locations cause different defect behaviour. Figure 4.3 shows an example where the behaviour seen at the two driven inputs, Input 1 and Input 2, depends on which wire segment, A, B, C or D, has the defect. The behaviour of a defect on segment A or segment C could be influenced by the neighbouring net N, but the behaviour of defects on segment B or D could not be influenced by net N. For the purpose of studying the impact of supply voltage on the behaviour and detectability of full opens, it is not necessary to study all potential defect locations. The results presented in this chapter are based on the assumption that the defect occurs close to the driver (such as the left end of segment A in Figure 4.3). This gives enough statistical material to study to observe the effect of varying supply voltage on the detectability of full open defects.

4.1.1 Logic Behaviour

The logic behaviour of a full open defect can be expressed in terms of the logic values seen by gate inputs driven by the victim net, based on how the victim net voltage compare with the logic threshold voltage of each input [77]. The logic threshold voltage is a simplified view of the transfer characteristics of a gate, for a particular input, see Section 3.2.2. A voltage above the logic threshold is seen as Logic-1 and otherwise as Logic-0. Therefore it can be seen that the logic behaviour of a full open defect depends on the voltage on the victim net and the logic threshold voltages of the inputs that are driven by the victim net (these inputs are in set IN). The logic threshold voltage is specific to each input and gate type and because of this, two inputs that are driven with the same voltage can see different logic behaviour.

From this discussion it can be seen that modelling of full opens should aim at providing the victim net voltage that is used to determine the logic behaviour in comparison with the logic threshold voltages of inputs driven by the victim net.

4.1.2 Test Methods for Full Open Defects

The main difficulty in testing for full open defects arises from the fact that the voltage on the victim net is a parameter which is influenced by the surrounding circuitry, such as neighbouring nets and transistor nodes of driven inputs as shown in Figure 4.1 and Figure 4.2. In [70], it was reported that some full opens studied in diagnosis were not timing dependent nor testable by IDDQ testing. However, some delay behaviour does occur in the presence of full open defects. In Section 4.2 and Section 4.3 it is shown that this delay behaviour does not change faulty/non-faulty status of the final static behaviour, for full opens without and with influence of gate tunnelling leakage respectively. Therefore, logic testing is effective if applied slow enough to overcome the first transient that is due to launching a test pattern, because the final voltage value for the victim net is detectable as a logic fault. This chapter does not discuss the test application time required to test for full opens in a static context. It could be more practical to apply a delay fault test to detect these full open defects, but as long as limitations to the tests that can be applied by launch-on-shift and launch-on-capture methods prevail [47, 46], some full open defects cannot be detected by delay fault testing. Therefore, there are cases when logic testing is the preferred method. This reasoning and the observation in [70] motivates using static, voltage-based testing for full opens. It should be noted that related defects such as resistive opens, tunnelling opens or floating gate opens require other test methods such as delay fault testing or IDDQ testing [121]. A floating-gate defect is an open that affects a single transistor, i.e. it is a gate-internal defect [75, 76, 77]. In [70], five floating gate open defects were found by diagnosis and none of them showed supply voltage-dependent behaviour. With this

observation in mind and for simplicity, the floating gate defect is left outside the scope of this chapter. Opens with a significant tunnelling effect (through the defect, not to be confused with gate tunnelling leakage) were studied in [81] and found to cause delay fault type behaviour and it was concluded that opens with tunnelling effect across the defect are well detectable by a delay fault test applied at Very-Low-Voltage (Section 2.3). Therefore, the study of this chapter will not include tunnelling leakage across the defect in the modelling, and instead address opens which separate the nodes involved in the defect to such an extent that no tunnelling occurs at the defect.

Several studies have recommended stuck-at-fault testing or similar to detect full open defects. For example, the work presented in [97] avoids the complexity of the analogue behaviour of the full open defect in order to achieve time-efficient simulation and test generation. The study in [97] has the benefit of being abstracted from the parameters of the design layout. However, a methodology that abstracts from the analogue behaviour of the physical defect cannot be used to determine the supply voltage-dependent behaviour of the defect. Another study [129] recommends an N-detection approach using stuck-at fault test patterns. N-detection testing means that a minimum number N of unique test patterns are targeting each potential fault location. It should be noted that N-detection testing is abstracted from the analogue behaviour of the defect. Instead, N-detection testing detects defects with high confidence due to the many test patterns. It has been observed that logic testing has a high defect coverage of full open defects if the nets that are physically close to the net-under-test (neighbouring nets, see Figure 4.1) are well controllable [122]. That is why three studies, [82,127,126], recommend controlling the neighbour nets to Logic-1 in order to test for Stuck-At-1 type behaviour and to Logic-0 in order to test for Stuck-At-0 type behaviour. The fact that neighbouring nets influence the circuit behaviour at the fault site can be understood by considering the capacitive coupling between nets that are physically close to one another as will be reviewed in Section 4.2. The effectiveness of test methods such as [97,129,122,82,127,126] can be understood by reviewing a test method that has full confidence in detecting full opens as will be shown next.

As it was noted above, the main difficulty in detecting full opens is that the behaviour of full open defects depends on the surrounding circuitry. Therefore, it is beneficial to perform both stuck-at-0 test and stuck-at-1 test, while maintaining the same logic configuration of the neighbouring nets. That way, the voltage on the net-under-test will remain the same for both tests. This voltage will be either above or below the logic threshold voltage for the input that is used to propagate the signal, leading to Logic-1 and Logic-0 respectively, so one of the two tests will detect the defect. It should be noted that this test approach requires no model for the voltage on the net-under-test or for the influence of supply voltage. Furthermore, such tests are easy to generate when the net-under-test is close to the primary inputs, or if the neighbourhood nets are otherwise easily controllable. So it can be seen that tests that exercise the same net many times

or makes detection probable by deliberately controlling the voltage on the neighbour nets are effective in detecting full opens even though they do not explicitly consider the voltage on the net-under-test.

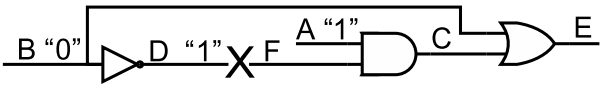


FIGURE 4.4: Defect that can only be detected through stuck-at-0 test

To show how the surrounding circuitry can make it impossible to generate stuck-at-1 or stuck-at-0 test patterns for the dual stuck-at test mentioned above, the example in Figure 4.4 is given for a particular full open fault-site. To be able to test for a full open between net D (the intended driver) and net F (the victim net), it is necessary to propagate a signal along the path F-C-E. To set up this path, it is required that net A is at Logic-1 and that net B is at Logic-0. This means that net D is at Logic-1. When net D is at Logic-1, the only possible fault is Logic-0 on net F as seen by the AND gate. Therefore, the only possible test is a stuck-at-0 test. This example shows that the requirement to propagate a signal from the net-under-test to the primary outputs can determine which tests that can be applied. It should be noted that net A is also required to be Logic-1 in order to propagate a signal from net F. If net A is a neighbouring net to net F, then capacitive coupling between net A and net F should be considered and the logic value on net A can influence the voltage value on net F and therefore influence the logic value seen by the AND gate. By this example it can be seen that it is not always possible to control the neighbouring nets freely, because of constraints due to the requirement to propagate from the net-under-test to primary outputs. This has the consequence that any test methods that rely on controlling the neighbour nets, such as the methods suggested in [82,127,126], will not be effective in all cases, because it is not always possible to control the neighbouring nets freely.

4.1.3 Supply Voltage-Dependent Detectability

A recent paper, [123], showed that there are cases when full opens cause supply voltage-dependent behaviour, but it was not quantified as to how common such supply voltage-dependent opens are. In [123], the impact of coupling capacitance on full open defect on interconnect was studied by measurement on an experimental circuit. The main focus of [123] was to study the impact of opens on a bus-like structure. It was found that the circuit studied in [123] becomes more sensitive to the influence of capacitive coupling from adjacent nets at lower supply voltage settings due to the fact that the logic threshold voltage depends on the supply voltage. This is a valid observation if coupling capacitance is the main factor that influences the voltage on the victim net. It is feasible that gate tunnelling leakage determines the victim net voltage. Therefore, the influence of gate tunnelling leakage should be studied in the context of supply voltage

variation, which is the focus in Section 4.3. The fact that full opens have supply voltage-dependent behaviour raises a challenge in manufacturing testing for full opens in designs with multiple supply voltage settings such as DVFS-enabled designs (Dynamic Voltage and Frequency Scaling, Section 1.1). The challenge is to determine the supply voltage to use while testing, because depending on supply voltage, an open defect may cause faulty or fault-free circuit operation. Consequently, it is necessary to perform testing at the supply voltage that makes the test effective in detecting the faulty behaviour from the defect, which leads to a test solution that involves testing using more than one supply voltage.

In the context of the example given in Figure 4.4, the supply voltage-dependent behaviour can cause some full open defects to be detected only for a particular supply voltage setting. The defect in Figure 4.4 is only detectable if it manifests as a Logic-0, because Logic-1 is the fault-free value. If supply voltage-dependency causes the defect to manifest as Logic-1, the defect is undetectable for that supply voltage setting. Otherwise, if for another supply voltage setting, the defect manifests as Logic-0, the defect is detectable for that supply voltage setting. This shows how supply voltage-dependent behaviour and circuitry that limits the detection possibility combine to lead to supply voltage-dependent detectability. This leads to loss of defect coverage, unless the test for the defect is applied for the right supply voltage.

Further analysis is required to determine the true impact of supply voltage variation on full open defects and detectability of such defects. It is the aim of this chapter to provide such an analysis. The analysis is performed using two models of the full open defect behaviour that focus on the static behaviour. A model that is based on the influence of capacitive coupling to neighbouring nets (Section 4.2) is compared to a model that is based on gate tunnelling leakage (Section 4.3). The two models complement each other so that one predicts the behaviour in the absence of leakage and the other predicts the behaviour in the presence of leakage.

4.2 Full Opens Influenced by Capacitive Coupling

Several studies, [123, 122, 72, 80, 77, 129, 75, 157, 119] and [118], have used a model for full opens that determines the voltage on the victim net (the net that is separated from its driver by the defect) by the following three factors. (1) The capacitance to the neighbouring nets (adjacent nets with capacitive coupling to the victim net) and power rails (Figure 4.1). (2) The capacitance to nodes of the driven transistors (Figure 4.2(a)). (3) Trapped charge on the victim net. The model is illustrated in Figure 4.1 with an example fault site with two neighbour nets $n1$ and $n2$ (N is the set of neighbouring nets) and two driven inputs $in1$ and $in2$ (IN is the set of gate inputs that are driven by the victim net).

Net F is the victim net, separated from its driver, represented by net D. The victim net F is influenced by capacitive coupling to the neighbouring nets, $n1$ and $n2$, capacitive coupling to transistor nodes of driven inputs, $in1$ and $in2$, and capacitive coupling to ground C_{FGND}, and supply voltage (Vdd) C_{FVdd}. The capacitive coupling to neighbouring nets is represented in Figure 4.1 by nets $n1$ and $n2$ and capacitance C_{Fn1} and C_{Fn2}. In the general case there is a set N of nets that are close enough to influence the victim net through capacitive coupling. The capacitive coupling to transistor nodes of driven inputs is represented in Figure 4.2(a) by C_{FPSin} (source), C_{FPBin} (bulk) and C_{FPDin} (drain) for PMOS and C_{FNSin}, C_{FNBin} and C_{FNDin} for NMOS of the in input, which is driven by net F. In the general case there is a set IN of inputs that are driven by the victim net F. For each input $in \in IN$ there is a set of nodes $M_{in} = \{PSin, PBin, PDin, NSin, NBin, NDin\}$ with capacitances $T_{in} = \{C_{FPSin}, C_{FPBin}, C_{FPDin}, C_{FNSin}, C_{FNBin}, C_{FNDin}\}$.

4.2.1 Model for the Final Victim Net Voltage

$$V_F = \frac{C_1}{C_1 + C_0} \cdot Vdd + \frac{Q_{trap}}{C_{FGND}} \tag{4.1}$$

The equation for the victim net voltage V_F is in Equation 4.1. Equation 4.1 shows how capacitances on net F function as a voltage divider to determine the voltage on net F. Nodes that carry Logic-1 (supply voltage, Vdd) are represented by C_1 and nodes that carry Logic-0 (GND voltage) are represented by C_0. The term Q_{trap}/C_{FGND} corresponds to voltage due to trapped charge on the net. It should be noted that the capacitances C_1 and C_0 change if the nodes in N or $M_{in \in IN}$ change logic value, but the sum of the capacitances $C_1 + C_0$ remains constant. Q_{trap} is charge that may be trapped on the victim net. This value will constant for a given defect but vary for different defects. The model for full opens may be considered with or without trapped charge Q_{trap}, considering the possibility of eliminating this charge during IC fabrication [123]. Taking trapped charge into account is associated with difficulty, because the value of Q_{trap} is not well known and hard to predict. Because of the difficulty in determining the Q_{trap} value, the model has inherent inaccuracy. Seen from a different perspective, it is not possible to make an accurate model without full knowledge of the value of Q_{trap}. The model in Equation 4.1 was used in [80] for simulation and in [123] it was used to reason about measurements from manufactured test structures and diagnosis. It has been shown in [72] that this model is useful for diagnosis of full open defects, because it can be used to predict the behaviour of some full open defects.

The defect coverage metric for the model is given in Equation 4.2 and it is not necessary to extend the defect coverage metric to include supply voltage variation, because the supply voltage is taken into account in Equation 4.1. A defect is counted as covered when the faulty behaviour predicted by the model has been activated by a test pattern

and a faulty behaviour has appeared on the primary outputs of the design. It should be noted that the same defect may have different faulty behaviours depending on the test pattern because the logic values on the neighbouring nets influence the behaviour. Only one of these faulty behaviours is required to be detected to cover the defect.

$$DC(T) = \frac{\text{The number of defects detected by test } T}{\text{The number of considered defects}} \tag{4.2}$$

4.2.2 Simulation of Full Open Defect Under Supply Voltage Variation

The analysis should first determine if supply voltage-dependent behaviour occur for full open defects. The following example, involving the fault site in Figure 4.5, demonstrates supply voltage-dependent behaviour caused by an open defect (Appendix A gives a definition of the fault site concept).

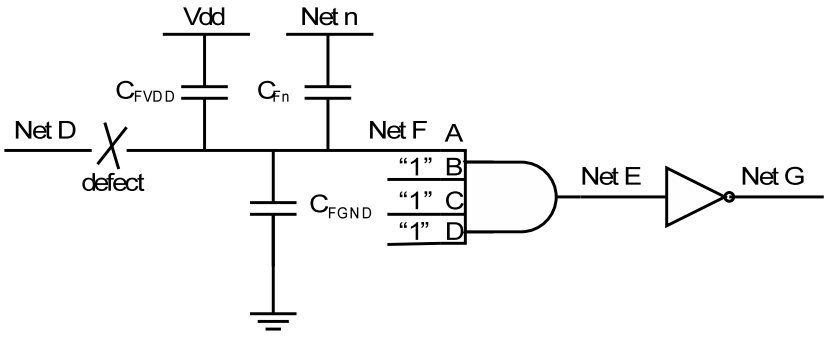
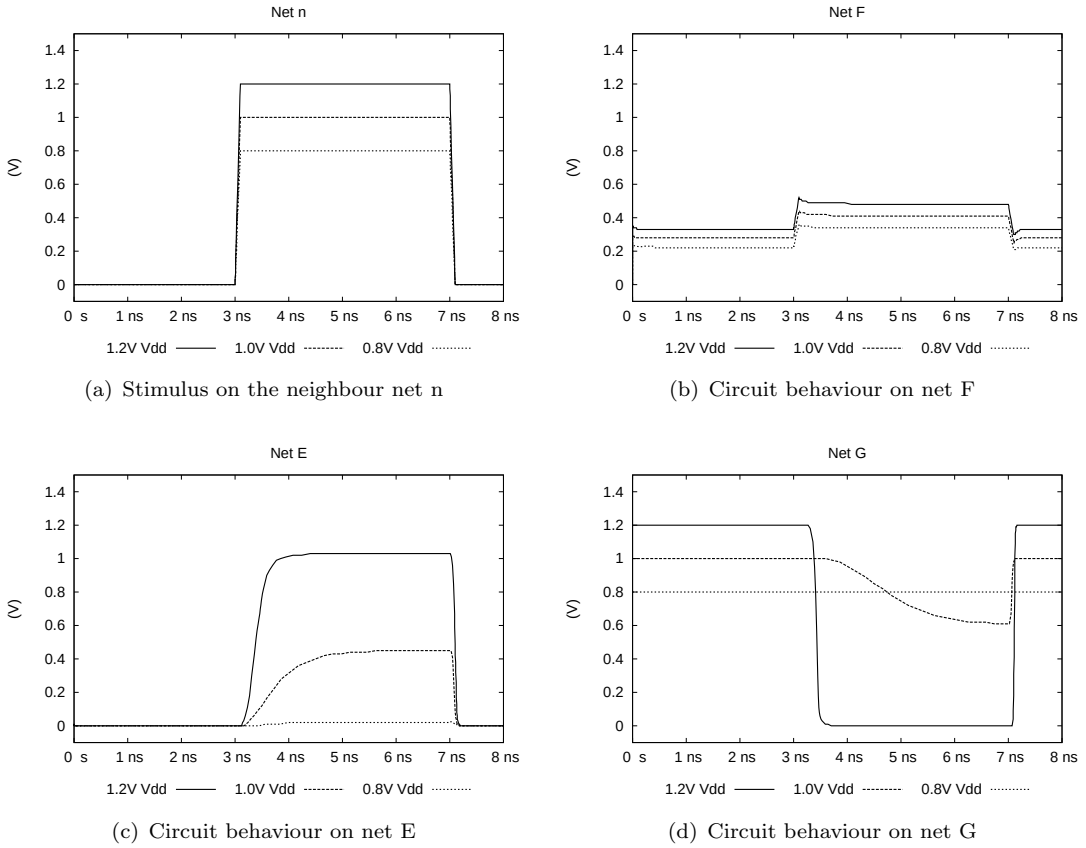


FIGURE 4.5: Simulated fault site with full open defect on interconnect

Net F in Figure 4.5 is separated from its driver, net D, by a full open defect. Net F drives the A input of a 4-input AND gate. Net F has a coupling capacitance C_{Fn} to a neighbouring net n. The fault site in Figure 4.5 was simulated in SPECTRE. The capacitances provided in the simulation set-up was $C_{Fn} = 1fF$, $C_{FVDD} = 1.675fF$ and $C_{FGND} = 1.7fF$. The trapped charge Q_{trap} was set to 0C for the simulation. These values were chosen for demonstration purposes. The capacitances to nodes of driven transistors are included in the Cadence Spectre model for the 4-input AND gate. The fault site in Figure 4.5 was simulated using transient simulation for the three supply voltage settings 1.2V, 1.0V and 0.8V. The waveforms are shown in Figure 4.6.

Figure 4.6 shows the transient simulation waveforms of simulating the fault site for three supply voltage settings, 1.2V, 1.0V and 0.8V. A stimulus was applied on net n as shown in Figure 4.6(a). Net n is at Logic-0 from time 0 to 3ns, then at Logic-1 from 3.1ns to 7ns (in this time frame, the voltage is at supply voltage level) and then at Logic-0 for the rest of the simulation. The other graphs in Figure 4.6 show how the circuit in Figure 4.5 behaves for net F (Figure 4.6(b)), net E (Figure 4.6(c)) and net G (Figure 4.6(d)). Figure 4.6(b) shows that the voltage on net F is influenced by the voltage on net n such



(a) Stimulus on the neighbour net n

(b) Circuit behaviour on net F

(c) Circuit behaviour on net E

(d) Circuit behaviour on net G

FIGURE 4.6: Waveforms for simulation on the fault site in Figure 4.5

that between 3.1ns and 7ns, the voltage on net F is elevated due to the change of logic value on net n. It should be noted in Figure 4.6(b) that the voltage on the victim net depends on supply voltage and the logic value on net n as stated by Equation 4.1. There are two voltage levels on net F, one during 0ns to 3ns and during 7.1ns to 8ns, and the other during 3.1ns to 7ns. The first voltage level is interpreted by the A input of the 4-input inverter as Logic-0 for all supply voltage settings, as propagated to net E and as can be seen in Figure 4.6(c) during 0ns to 3ns and during 7.1ns to 8ns. The second voltage level causes different behaviours on net E for different supply voltages, which is shown in Figure 4.6(c) during 3.1ns to 7ns. Consider Figure 4.6(c) for the time 6ns and supply voltage setting 1.2V, for which the voltage on net F is ≈1V, well above half swing ($Vdd/2 = 0.6V$), and detected as Logic-1 by the inverter in Figure 4.5, causing Logic-0 on net G, as shown in Figure 4.6(d). For 1.0V supply voltage and time 6ns, net E (Figure 4.6(c)) is in the middle of the swing, so it is difficult to predict what logic value the inverter would see. It depends on the transfer characteristics of the inverter. Indeed, as Figure 4.6(d) shows, net G has 0.6V for 1.0V supply voltage. That is above half swing ($Vdd/2 = 0.5V$) indicating that a weak Logic-0 was seen by the inverter. Further, at 0.8V supply voltage and the time 6ns, the AND gate input sees a Logic-0, and net E is not influenced by the signal on net n. From the above, it can be observed that full opens cause supply voltage-dependent circuit behaviour, as there was Logic-1

on net E for supply voltage 1.2V and Logic-0 for supply voltage 0.8V.

Delay behaviour can be seen on net E (Figure 4.6(c)) for the 1.0V supply voltage. This delay is due to the fact that the voltage on the victim net is so close to the voltage for which the AND gate switches its output behaviour. This means that both the NMOS and the PMOS network in the AND gate are active, leading to a current that goes directly through the AND gate from supply voltage (Vdd) to GND. Only a remaining small current charges the load capacitances of the AND gate, which becomes a slow process. It should be noted that not all delay behaviours are delay fault type behaviours, because if the final logic value is faulty, it is not a delay fault type behaviour.

The voltage on the victim net in this example can be said to be intermediate, which means that it is not a GND voltage level or supply voltage level but more or less half-swing. This intermediate voltage on net F is according to the model in Equation 4.1 caused by capacitive coupling to neighbouring nets. supply voltage-dependent behaviour occurs when such intermediate voltage is interpreted as Logic-1 at one supply voltage and as Logic-0 at another voltage due to the transfer characteristics of the gate. As the amount of intermediate voltage scale with supply voltage and as the transfer characteristics of a gate change with supply voltage, supply voltage-dependent behaviour is the outcome.

4.2.3 Coupling Capacitance-Aware Analysis of Full Open Behaviour

The supply voltage-dependent behaviour of full open defects is analysed in this section using the model in Equation 4.1 with regard to the impact of full opens on testing. An example is studied regarding supply voltage-dependent behaviour and its mechanisms.

To examine how the full open defect behaves with different neighbour net assignments and supply voltage settings, consider the open fault site shown in Figure 4.7. Victim net F is influenced by the capacitance of nets n1 and n2, the capacitance to nodes of the transistors in the driven input *in* and the capacitance to ground and supply voltage. Net F drives an input *in* with the logic threshold voltage Th_{in}. The logic assignment to n1 and n2 determines the voltage V_F (Equation 4.1), as shown by Table 4.2. The V_F values are calculated using the values given in Table 4.1, which are assumed for demonstration purposes. The typical total gate capacitance ($C_{FPBin} + C_{FNBin} + C_{FPSin} + C_{FNSin} + C_{FPDin} + C_{FNDin}$) is in the magnitude range of $[1fF, 2fF]$ for the considered 0.12μm gate library. Coupling capacitance to neighbouring nets can get up to 2fF for neighbouring nets that run in parallel for a long distance, according to the coupling capacitance data collected as described in Section 4.4.1 for use in experiments such as those described in Section 4.5. In the example at hand, net n1 runs in parallel with net F for 14.4μm, both on metal layer M3 with a minimum distance between them. Similarly, net n2 runs in parallel with net F for 22.8μm.

As can be seen in Table 4.2, the highest voltage for V_F (0.745V) is achieved by setting all neighbour nets to Logic-1. As both nets n1 and n2 are at Logic-1 they contribute to C_1 (Equation 4.1) but not to C_0. Similarly, when all neighbour nets are at Logic-0, C_{Fn1} and C_{Fn2} contributes to C_0 but not C_1. This means that the capacitive coupling term of Equation 4.1 has no impact on V_F, which is then only determined by the trapped charge Q_{trap} and the capacitance to ground. The V_F values for the neighbour net assignments are shown in Table 4.2. The example shows that the victim net voltage V_F strongly depends on the neighbour net assignments.

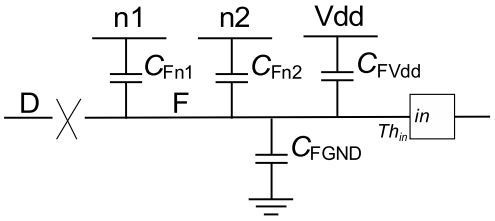


FIGURE 4.7: Single supply voltage example of full open defect

TABLE 4.1: Values used to generate Table 4.2

Vdd	1.2V	C_{FPBin}	0.1fF
Th_{in}	0.65V	C_{FNBin}	0.1fF
C_{FVdd}	0.5fF	C_{FPSin}	0.75fF
C_{FGND}	1fF	C_{FNSin}	0.5fF
C_{Fn1}	0.6fF	C_{FPDin}	0.75fF
C_{Fn2}	0.95fF	C_{FNDin}	0.5fF
Q_{trap}	0.14fC		

TABLE 4.2: Victim net voltage for various neighbour net assignments

n1	**n2**	V_F
0	0	0.422V
0	1	0.620V
1	0	0.547V
1	1	0.745V

To translate the analog behaviour of the defect in Figure 4.7 into the digital domain, the logic threshold voltage Th_{in} for the driven input *in* is shown in Table 4.1. The value for Th_{in} (0.65V) is the logic threshold voltage for the A input of a four input NOR gate in the considered 0.12μm gate library at 1.2V supply voltage. The voltage $V_F = 0.745V$ for neighbour net assignment n1:1 n2:1 translates to a Logic-1, as seen by the driven input *in*, because it is above the logic threshold voltage Th_{in}. Assignment n1:0 n2:1 causes $V_F = 0.620V$ which is below the logic threshold voltage, so the input *in* sees a Logic-0. The remaining two assignments also cause voltages on the victim net that are below the logic threshold voltage, causing Logic-0 to be seen on the input *in* in both cases. The

lowest voltage (for n1:0 n2:0) is 0.422V and the highest voltage (for n1:1 n2:1) is 0.745V. The example shows how the logic behaviour is determined by the victim net voltage and the logic threshold voltages of the driven inputs.

The full open defect causes faulty behaviour only when the victim net voltage is such that one or more of the driven inputs sees the opposite logic value from the intended "good" logic value. The intended value is represented in Figure 4.7 by net D. If net D is held at Logic-0 (the "good" value'), the neighbour net assignment n1:1 n2:1 would expose faulty behaviour (Logic-1) on the input *in*. For the complementary case, when D is held at Logic-1, the other assignments would cause faulty behaviour (Logic-0) on the input *in*. Consequently, the logic values exposed by the defect are faulty or non-faulty depending on the intended value on the victim net, which is independent from the mechanism of the defect (i.e. it does not depend on the supply voltage, neighbouring nets or logic thresholds).

TABLE 4.3: The victim net voltages for given neighbour net assignments and supply voltage settings

n1	**n2**	V_F**@0.8V Vdd**	V_F**@1.0V Vdd**	V_F**@1.2V Vdd**
0	0	0.328V	0.375V	0.422V
0	1	0.460V	0.540V	0.620V
1	0	0.411V	0.479V	0.547V
1	1	0.544V	0.644V	0.745V

TABLE 4.4: The logic threshold voltage Th_{in} for three supply voltage settings

	0.8V Vdd	**1.0V Vdd**	**1.2V Vdd**
Th_{in}	0.42	0.54	0.65

The discussion so far has focused on how full open defects behave in a single voltage scenario. Next, consider what happens to the behaviour of the defect when the circuit is operated with three supply voltage settings (0.8V, 1.0V and 1.2V). Table 4.3 shows the impact of supply voltage on V_F. Table 4.3 shows the victim net voltage V_F for all the possible assignments to the neighbour nets n1 and n2, as supply voltage is varied. In Table 4.3, the V_F values scale linearly with supply voltage, as described by Equation 4.1. The logic threshold voltage Th_{in} is shown in Table 4.4 for each supply voltage setting. The Th_{in} values are for the A input of a 4-input NOR gate. The logic threshold voltage also scales with supply voltage as can be seen in Table 4.4. Similarly to the single supply voltage situation in Table 4.2, the V_F values for assignment n1:1 n2:1 are higher than Th_{in}, and therefore the input sees Logic-1, for all supply voltage settings. Similarly, for the assignments n1:1 n2:0 and n1:0 n2:0, the V_F values are lower than the logic threshold, manifesting Logic-0, for all supply voltage settings. However, assignment n1:0 n2:1 causes a V_F value higher than Th_{in} for supply voltage 0.8V but lower than Th_{in} for supply voltage 1.2V. For supply voltage 1.0V, V_F is too close to the threshold voltage

Th_{in} to reliably predict the logic behaviour of the gate. This means that, for supply voltage 0.8V, the input sees Logic-1 and for supply voltage 1.2V, the input sees Logic-0. This shows that the logic behaviour depends on the supply voltage setting.

The discussion above mentioned that the behaviour of full open defects depends on the coupling capacitance to neighbouring nets and transistor nodes in driven inputs. This following discussion demonstrates how the victim net voltage depends on the capacitance value. It also shows the method used in this work to consider the capacitive coupling between the victim net and nodes of driven transistors.

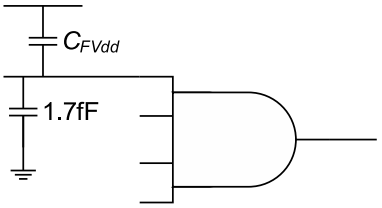


FIGURE 4.8: Setup for studying the capacitance between the gate input and nodes of driven transistors

In order to determine the impact of coupling capacitance on the victim net voltage, Cadence Spectre simulations were performed for supply voltage 1.0V for the circuit in Figure 4.8 as C_{FVdd} is varied, with $C_{FGND} = 1.7fF$ and $Q_{trap} = 0C$. Figure 4.9 shows the victim net voltage V_F dependence on the coupling capacitance in the range from 0F to 6fF.

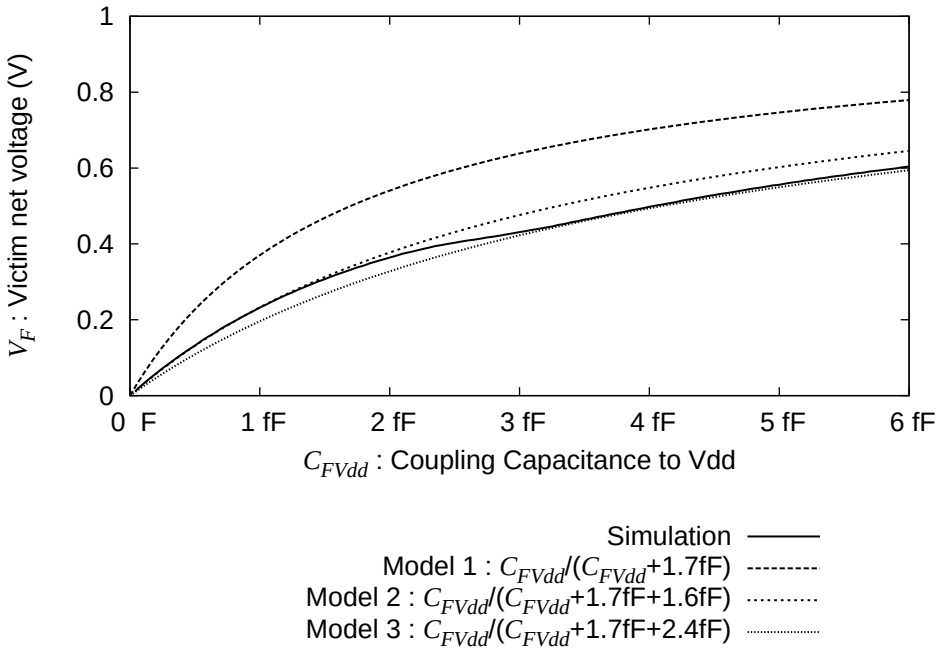


FIGURE 4.9: Victim net voltage versus coupling capacitance to supply voltage

As the capacitance to nodes with supply voltage (C_{FVdd}) increases, the victim net voltage increases. As can be seen in Figure 4.9, the victim net voltage (solid curve) as

generated by Cadence Spectre simulation is 0V when C_{FVdd}=0F and increases when C_{FVdd} increases. Also included in Figure 4.9 are three curves that show how the dependence on the coupling capacitance can be modelled. The dashed curve (Model 1) is modelling without regard to the capacitances to nodes of the driven input (the A input to the 4-input AND gate of Figure 4.5) and only includes C_{FVdd} and C_{FGND}. The function for Model 1 is $V_{Model1} = \frac{C_{FVdd}}{C_{FVdd}+C_{FGND}} \cdot Vdd = \frac{C_{FVdd}}{C_{FVdd}+1.7\cdot10^{-15}}$. As can be seen from Figure 4.9, Model 1 is following the general shape of the curve for the victim net voltage, but is not able to predict the victim net voltage. In order to overcome this failure to predict the victim net voltage, the model is adjusted as marked with Model 2 and a dotted curve in Figure 4.9. Adjusting Model 1 with an extra capacitance to ground $C_{extra} = 1.6fF$ produces the dotted curve Model 2 in Figure 4.9. The extra capacitance is represents the capacitance between the victim net transistor nodes the driven gate input. The function for the dotted graph (Model 2) is $V_{Model2} = \frac{C_{FVdd}}{C_{FVdd}+C_{FGND}+C_{extra}} \cdot Vdd = \frac{C_{FVdd}}{C_{FVdd}+1.7\cdot10^{-15}+1.6\cdot10^{-15}}$. As can be seen by comparing the dotted graph (Model 2) with the solid graph (Spectre simulation), the updated model predicts the victim net voltage well in the range 0F to 2fF (V_F <0.35V). After this point, the victim net voltage diverges from the model curve. The behaviour can be explained by the fact that the victim net voltage at this point is high enough for the driven AND gate to switch from seeing a Logic-0 to seeing a Logic-1, which affects the voltage on nodes of the driven transistors. As the AND gate switches, the voltage on the drain nodes of the driven transistors change. This change in voltage on the transistor nodes appears in Figure 4.9 as though the extra adjustment capacitance C_{extra} changed. Therefore, instead of taking into account the changing voltage on the nodes of driven transistors, the model can regard the capacitance to the driven transistors as the changing factor. Thus, there should be two values, $C_{extra,LOW}$ for low victim net voltages (<Vdd/2) and $C_{extra,HIGH}$ for high victim net voltages. In Figure 4.9, Model 2, dotted line, correspond to use of $C_{extra,LOW} = 1.6fF$ and Model 3, grey line, correspond to use of $C_{extra,HIGH} = 2.4fF$. Together, Model 2 and Model 3 represent the voltage on the victim net. The $C_{extra,LOW}$ and $C_{extra,HIGH}$ capacitance values have been generated with the procedure of the example above (fitting the V_F equation to SPECTRE simulation) for all the inputs in the gate library and are used in the simulations of full open defects in this chapter.

From the above it can be seen that the victim net voltage is influenced by capacitance to neighbouring nets and capacitances to transistor nodes of driven inputs. Capacitive voltage division is an appropriate model for calculating the victim net voltage without analogue simulation if there is no gate tunnelling leakage.

4.3 Full Opens Influenced by Gate Tunnelling Leakage

The leakage-unaware model in Equation 4.1 has been criticised [84] because it fails to include the effect of gate tunnelling leakage. Gate tunnelling leakage [158, 159] occurs predominately for deep submicron technology designs because of the very thin gate oxide/dielectric used. A thin gate oxide is preferred for performance reasons, but leads to gate tunnelling leakage. Gate tunnelling leakage allows charge to travel through the gate oxide, i.e. the thin gate oxide used in deep submicron technologies is not an ideal isolator. Because of gate tunnelling leakage, the victim net for a full open defect is not electrically isolated from the nodes of driven transistors [83]. To maintain high performance and to reduce gate tunnelling leakage, some deep submicron technologies use materials with a high dielectric constant compared to silicon dioxide, because this makes it possible to make the gate oxide/dielectric thicker [120].

In the presence of gate tunnelling leakage, the capacitive coupling to neighbour nets or initially trapped charge (discussed in Section 4.2) has no influence on the final static voltage on the victim net. Instead, the victim net voltage depends on the charge that is entering and leaving the victim net due to gate tunnelling leakage.

In [84] it was found that the delay between applying a test pattern and finding a stable voltage on the victim net of a full open defect can be as large as tens or hundreds of μs. However, as was discussed in Section 4.1.2, this study focuses on logic testing and does not discuss the test application time.

4.3.1 Simulation of Full Open Defect Under Supply Voltage Variation

To show that gate tunnelling leakage can cause supply voltage-dependent behaviour, the circuit in Figure 4.10 was simulated in Cadence Spectre with a gate tunnelling leakage-aware transistor model for the 0.12μm VLSI technology from [160] and a 0.12μm gate library from ST Microelectronics. The simulation is a transient simulation, which means that waveforms are generated where the voltage on the nodes of the circuit changes over time, as shown in Figure 4.11. To get realistic delays in the simulation, net H, net L and net N each have a 5fF capacitor to ground. The 5fF capacitors are chosen for demonstration purposes but correspond to the typical input capacitance of the input of an inverter.

Figure 4.10 shows how the victim net (net F) is separated from its driver (net D) by the defect (marked with an X) and drives three inputs, (1) the A input of a 4-input AND gate, (2) the B input of a 2-input AND gate and (3) the A input of another 2-input AND gate. The fact that net F drives three inputs means that there is leakage from three NMOS transistors and three PMOS transistors influencing the voltage on net F (one NMOS and one PMOS transistor per driven input, see Figure 4.2(b)). However,

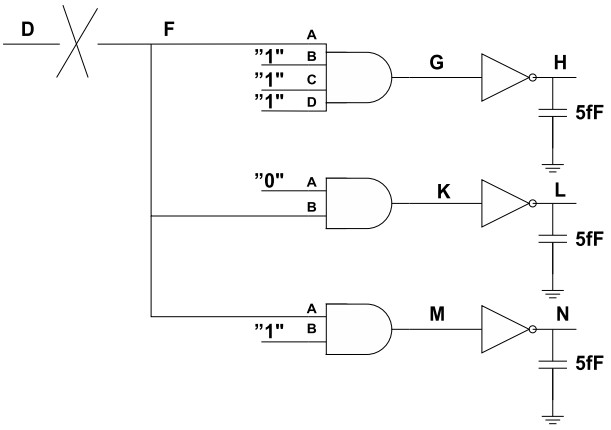
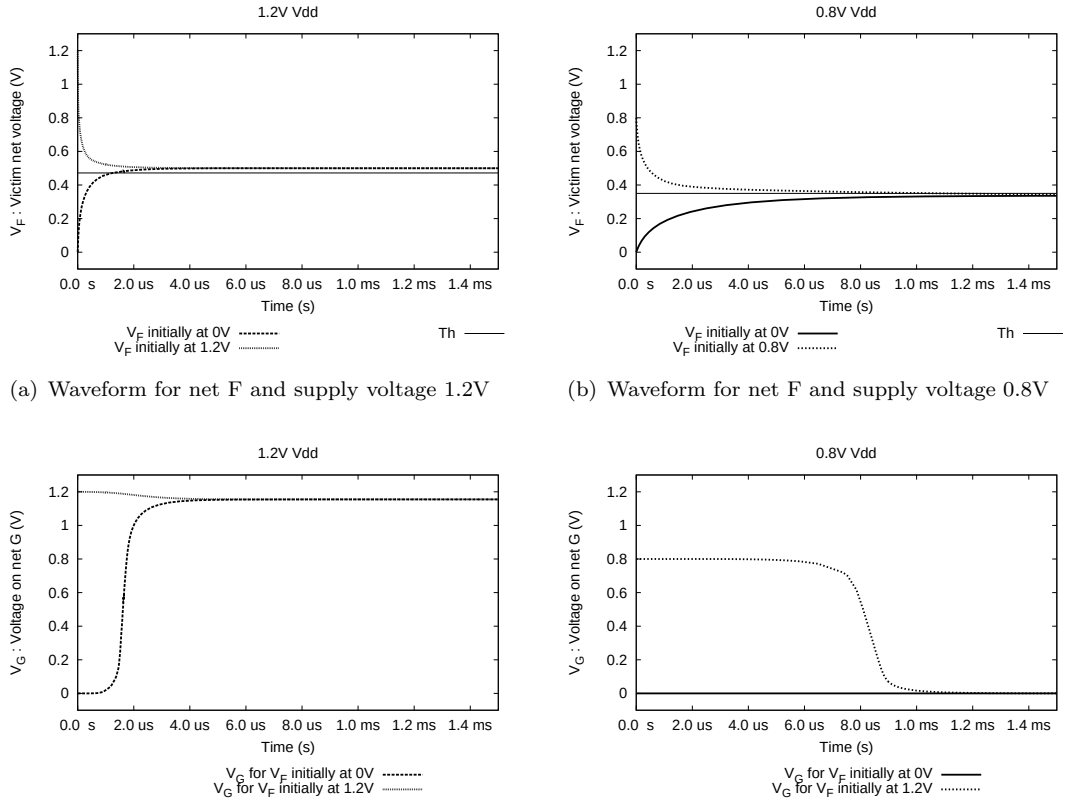


FIGURE 4.10: Simulated circuit where net F is influenced by gate leakage

the AND gate that drives net K in Figure 4.10 has Logic-0 on the side input (input A). Due to the internal structure of the AND gate, this separates the B input of the gate from ground, so in practise, this gate will only cause leakage current from the power supply onto net F (through the corresponding PMOS transistor). Such inputs, that are separated from a power rail by the logic values on other inputs, are called non-controlling inputs, because the input cannot affect the output behaviour of the gate.

The waveforms from simulation of the circuit in Figure 4.10 are shown in Figure 4.11. Figure 4.11(a) and Figure 4.11(b) show the waveforms for net F, for 1.2V supply voltage and 0.8V supply voltage respectively. In each graph in Figure 4.11, two curves are shown. There are two curves in each graph to describe the behaviour for the case that the initial (time 0s) victim net voltage was high (Vdd) and for the case that the initial victim net voltage was low (ground). There are cases when the victim net voltage is bi-stable [29], i.e. the final static value will depend on the initial value, where the pivot point is the logic threshold voltage (the voltage for which the gate switches its output behaviour). The fact that the final voltage on the nets is the same for both cases, as can be seen for time 1.4ms in all graphs, means that the final static voltage of the circuit in Figure 4.10 is independent of the initial voltage on the victim net. The final voltage on the victim net is 0.500V for 1.2V supply voltage and 0.337V for 0.8V supply voltage. The final voltage for net F is not achieved instantly, but is subject to the delay time it takes to charge involved capacitances accordingly using gate tunnelling leakage currents. The final victim net voltage is found after 0.3μs for 1.2V supply voltage and after 1ms for 0.8V supply voltage. The delay of 1ms for 0.8V supply voltage is small in comparison with results from [84], which reported for another but similar experiment that the time until the voltage on the victim net reaches its final static value will be in the order of tens or hundreds of μs.

Figure 4.11(c) and Figure 4.11(d) show the waveforms on net G. The voltage on net G reveals the logic behaviour seen for net F on the A input of the 4-input AND gate. For 1.2V supply voltage net G has a high voltage, Logic-1, and for 0.8V supply voltage it has



(a) Waveform for net F and supply voltage 1.2V

(b) Waveform for net F and supply voltage 0.8V

(c) Waveform for net G and supply voltage 1.2V

(d) Waveform for net G and supply voltage 0.8V

FIGURE 4.11: Waveforms for simulation of the circuit in Figure 4.10

low voltage, Logic-0. This shows that the same defect causes different logic behaviour for different supply voltages.

4.3.2 Gate Tunnelling Leakage-Aware Analysis of Full Open Behaviour

This section presents an analysis of the behaviour of full open defects in the presence of gate tunnelling leakage. To understand the impact of gate tunnelling leakage on full open defects, one must consider the gate inputs that are driven by the victim net. Each gate input is associated with two transistors, one NMOS and one PMOS, so that the input net is connected to their gate nodes. The gate tunnelling leakage for the NMOS transistor of a gate input *in* has three components [84], as shown in Figure 4.2(b). IN is the set of inputs that are driven by the same net, not to be confused with the currents in the figure. I_{NSin} is the current through the source node, I_{NCin} is current associated with the transistor channel and I_{NDin} is the current through the drain node. Similarly, the PMOS transistors for input *in* has the corresponding leakage current components I_{PSin}, I_{PCin} and I_{PDin}. The arrows shown in Figure 4.2(b) do not specify the direction of the current but the convention for the sign associated to the corresponding current component. A current from supply voltage to the victim net is considered positive and

a current from the victim net to ground is considered negative. The direction of the channel component of the leakage current is into the victim net for PMOS transistors (I_{PCin}) and from the victim net for NMOS transistors (I_{NCin}). The direction of the current for source and drain nodes of the transistors depend on the voltages on the gate and on the source and drain nodes.

The current components of the gate tunnelling leakage are regulated by the difference in voltage between the gate and the other transistor nodes. It should be noted that the channel component is only present when the transistor is conducting. The current component for the channel is up to ten orders of magnitude larger than the leakage through the source and drain nodes [158], due to the larger overlap area between the gate and the channel compared to that of the source/drain nodes. The gate tunnelling leakages for NMOS and PMOS transistors are similar, except that NMOS transistors have higher current density (up to 20A/cm^2) than PMOS transistors (up to 2A/cm^2) [158]. On the other hand, PMOS gates have larger area.

To show how the gate tunnelling leakage depends on the gate voltage and the voltage on the drain nodes, simulations were performed on the transistors in an inverter from a 0.12μm gate library from ST Microelectronics [14] using tunnelling leakage aware transistor models from [160]. The transistor model is aware of gate tunnelling leakage through the BSIM4 definition [161]. The NMOS transistor was 260nm wide and the PMOS transistor was 470nm wide.

Figure 4.12 shows the gate tunnelling leakage for NMOS (Figure 4.12(a) and Figure 4.12(b)) and PMOS (Figure 4.12(c) and Figure 4.12(d)) transistors. Furthermore, two supply voltages settings are used, 1.2V for Figure 4.12(a) and Figure 4.12(c), and 0.8V for Figure 4.12(b) and Figure 4.12(d). It should be noted that the gate leakage current depends on the supply voltage. The gate tunnelling leakage current is overall less for 0.8V than for 1.2V supply voltage, as can be seen by the grading of the vertical axes. Furthermore, the gate leakage current through PMOS transistors is significantly smaller than the gate leakage current through NMOS transistors. For example, the peak magnitude of the current at 1.2V supply voltage is ≈0.85nA for NMOS (the right side of Figure 4.12(a)) but only ≈9pA for PMOS (the dip at V_g=0.9V in Figure 4.12(c)). Figure 4.12 shows curves for three different voltages on the drain node. The fact that the curves depend on the drain node voltage indicates the effect of tunnelling current through the drain node. The gate tunnelling leakage through the drain node of a transistor has influence on the gate voltage in two ways. Firstly, when the gate voltage is low and the voltage on the drain node is high, the current through the drain node is charging the gate. This can be seen in the left half of Figure 4.12(a) and V_d=1.2V. Secondly, for high gate voltages and low drain node voltages, it is discharging the gate, as shown in the right half of Figure 4.12(a) and V_d=0.0V. The same effect occur by the source node, but in the experiments that generated Figure 4.12 the source node was connected to the power rail, ground voltage for NMOS and supply voltage for PMOS.

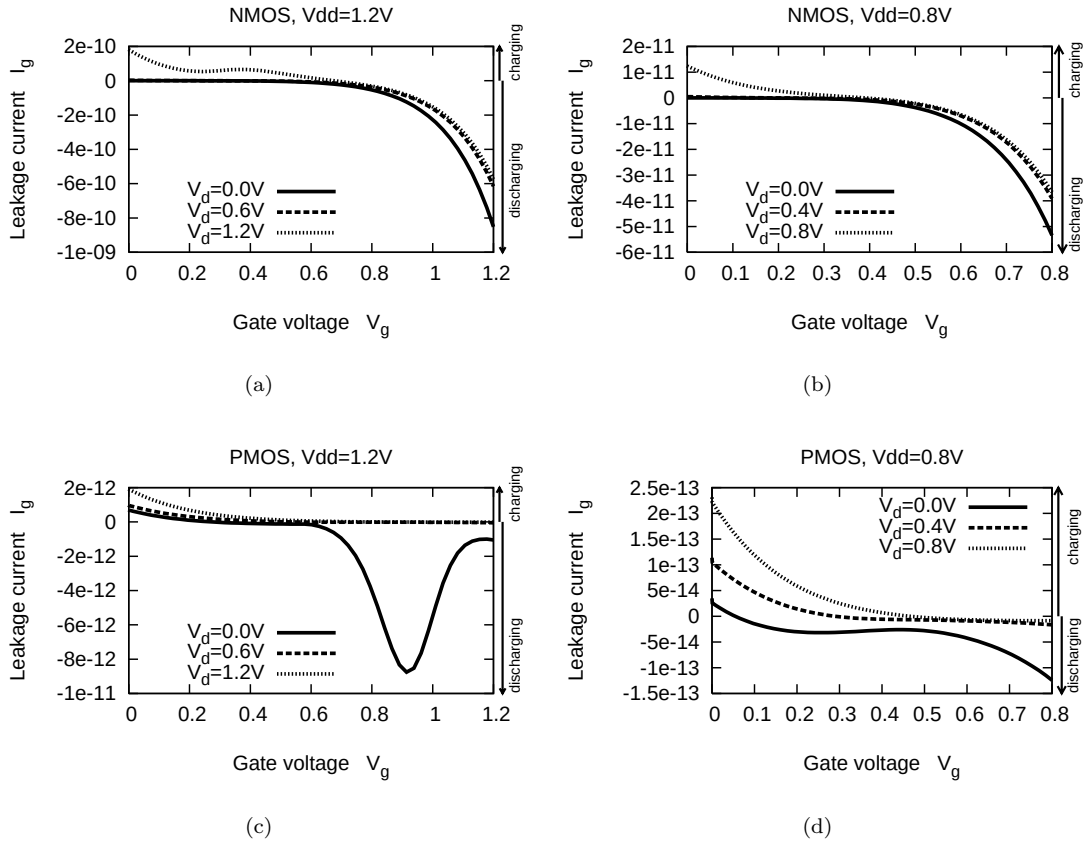


FIGURE 4.12: Gate tunnelling leakage current for 1.2V and 0.8V supply voltage

The leakage from the gate channel can be seen for high gate voltages V_g for NMOS (Figure 4.12(a) and Figure 4.12(b)), as such high gate voltages will cause NMOS transistors to have a conducting channel. It can be seen that the magnitude of the gate tunnelling leakage from the channel increases significantly with V_g. The leakage current from the channel of NMOS transistors is negative, which means that the gate is discharged, as marked by the arrows to the right of the graphs. Correspondingly for the PMOS transistor, the leakage from the channel can be seen for low gate voltages V_g (Figure 4.12(c) and Figure 4.12(d)). The leakage from the channel of PMOS transistors is positive, which means that the gate is charged.

In Figure 4.12(c), there is a strong dip in the gate leakage current around $V_g = 0.9V$ (and correspondingly in Figure 4.12(a) there is a small peak around $V_g = 0.35$). These effects are due to leakage through the drain node and channel pinch-off which enlarges the effective area of the drain node underneath the gate and causes more tunnelling leakage. Channel pinch-off occurs when the transistor operates in saturation mode ($V_{GS} > VTN$, $V_{DS} > (V_{GS} - VTN)$ for NMOS and $V_{GS} < VTP$, $V_{DS} < (V_{GS} - VTP)$ for PMOS), which is why it occurs for high gate voltages on PMOS and for low gate voltages on NMOS. Only conducting transistors have saturation mode. When V_g decreases below VTN or increases above Vdd+VTP, the transistor stops conducting for NMOS and

PMOS respectively, which means that the drain node reverts to its original size and the tunnelling current is the regular drain node leakage without the enlarged drain node. That is why the negative peak in Figure 4.12(c) is close to V_g=Vdd+VTP and the local maximum in Figure 4.12(a) is close to V_g=VTN.

4.3.3 Model for the Final Victim Net Voltage

This section aims to define a simplistic fault model for full opens based on [84, 158, 83, 159, 29] that imitates the static behaviour caused by full opens in the presence of gate tunnelling leakage. The purpose of defining the model is to enable statistical experimental analysis regarding the voltage on the victim net of full open defects influenced by gate tunnelling leakage. The experimental analysis seeks the distribution of victim net voltages to determine the typical victim net voltage value and its spread. This is done by considering many full open defect locations in benchmark designs, corresponding to a multitude of configurations of inputs in different gates and the leakage through them. Analog simulation with Cadence Spectre could be employed to determine the victim net voltage for the considered full open defect, but a less CPU-demanding modelling approach is sought to enable simulation of multiple full open defect locations in benchmark designs within a reasonable time. However, less CPU-demanding modelling typically means that less details are considered, which implies reduced accuracy. It should be noted that to determine the distribution of typical victim net voltage values, it is not necessary to have full accuracy for each defect. As long as the modelling is realistic in any assumptions taken, the results should be representative. The modelling used for experimentation and the assumptions taken are described below. It should be noted that the model is intended as a tool for analysis and not intended to guide test generation.

The model proposed in this section can be applied for different supply voltage settings and involves two mechanisms that can influence the leakage through the gate oxide of a transistor which has the victim net on the gate node. These two mechanisms are (1) the value of the voltage on the drain node (Section 4.3.2), and (2) the impact of the logic input on other inputs to the same gate, as transistors from such inputs can cut off the considered input from the supply voltage or ground power rails (described below). The first mechanism, the dependence on the drain node voltage, strongly depends on the second mechanism, the input assignment to other inputs. The complexity of these two mechanisms is present in all gates except inverters and buffers, where there is only one input and the drain voltage of involved transistors can be described by an equation corresponding to the input to output transfer characteristics of an inverter. For other gates it is more difficult to consider these two mechanisms accurately because it requires detailed knowledge of the internal structure of each gate. Therefore, the proposed model employs the simplifying assumption that most inputs behave like the inputs of inverters,

except for randomly chosen inputs, for which the leakage from either power supply of ground is cut off, as would be the effect of having a non-conducting transistor between the input and the corresponding power rail (described further below). 30% of the inputs are in this way cut off from power supply and 30% are cut off from ground. The implications of setting the percentage to 30% is further discussed in Section 4.5.1.2 and Section 4.5.3. The simplifying assumption can lead to situations when the model gives a different victim net voltage than Cadence Spectre, but each modelled defect instance still corresponds to realistic defects. Therefore, the collected statistics about the victim net voltage still gives a realistic distribution showing the typical victim net voltage value and spread.

As mentioned in Section 4.3.2, for high voltages on the victim net ($V_F > Vdd + VTP$), leakage through the channel of NMOS transistors will dominate and drain the victim net voltage of charge, i.e. reduce the voltage on the victim net V_F. Similarly, for low victim net voltages ($V_F < VTN$), leakage through the channel of PMOS transistors and the drain leakage of NMOS transistors will dominate and increase the charge of the victim net, i.e. increase V_F [29]. This means that, for the typical full open defect, the final static victim net voltage is likely to be within the range $[VTN, Vdd + VTP]$. However, it should be noted that a transistor can be cut off from the corresponding power rail (supply voltage for PMOS and ground voltage for NMOS) by another transistor that is situated in-between, and this second transistor is not conducting [158]. Inputs that have either the NMOS or PMOS transistor cut off like this are called non-controlling inputs. An example of this was discussed in Section 4.3.1 for the AND gate that drives net K in Figure 4.10. Transistors that are cut off from a power rail in this way, will have very low or no leakage current through the source and channel nodes. Because of this, there can be more leaking NMOS transistors than PMOS transistors and vice verse, which means that the victim net voltage V_F is adjusted accordingly and can even reach one of the supply rail voltages (supply voltage or ground voltage). As mentioned above, determining which inputs are disconnected from a power rail requires detailed information about the internal structure of each gate. Instead, the proposed model assumes that all inputs behave as if they were inverters except for some randomly selected inputs that are non-controlling.

The model used for experimentation with full opens and gate tunnelling leakage determines the victim net voltage for which the current to and from the victim net is equal. This means that the amount of charge on the victim net is not changing. The current through each involved transistor is determined by curves such as those shown in Figure 4.12. The current is proportional to the width of the transistor. The difficulty involved in this model is to determine the drain voltage for each involved transistor which will vary with V_F according to the input-to-output transfer characteristics of the considered input. For simplicity, the drain voltage is assumed to follow the gate input-to-output transfer characteristics of the inverter used in the examples above, but adjusted to the different gates of the gate library using the W_N/W_P ratio for the considered gate

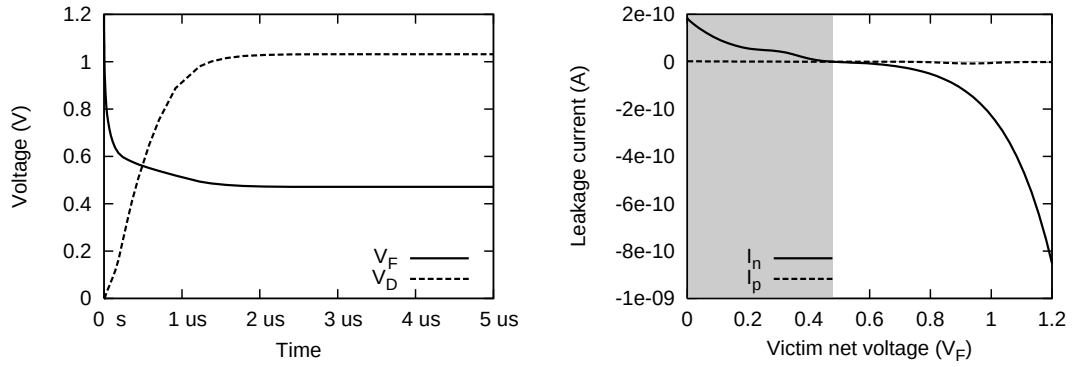
input. To find the final static victim net voltage, $V_{F,static}$, the following steps are taken:

- Adjust the transfer characteristics curve $T(V_F)$ according to the W_N/W_P ratio for each considered input in, $T_{in}(V_F) = T(V_F + f(W_{Nin}/W_{Pin}))$.
- Calculate the leakage current curve $I_t(V_F)$ through each involved transistor t by selecting, for each V_F value, a curve in Figure 4.12. The selection is based on the V_D value that best corresponds to the assumed drain voltage value $T_{in}(V_F)$ curve. Only transistors that have a connection between the source node and the appropriate power rail (supply voltage for PMOS and ground for NMOS) are considered.
- Adjust $I_t(V_F)$ with the W_N/W_P ratio for the corresponding input in, $I_{t,in}(V_F) = g(W_N/W_P) \cdot I_t(V_F)$.
- Sum all the leakage currents for each transistor t of each considered input in, $I_{Tot}(V_F) = \sum_{in} \sum_t I_{t,in}(V_F)$.
- The final static V_F value is that which causes this sum to be zero, $I_{Tot}(V_{F,static}) = 0$.

The f and g functions used in the steps above were fitted to Cadence Spectre simulations to fulfil the respective tasks. The simplifications used in the model enables generation of reasonable victim net voltage values, which has been verified by Cadence Spectre simulation of more than 50 full open defects. By reasonable victim net values, it is meant that in most cases the model give the same victim net voltage as Cadence Spectre and the overall distribution of victim net voltage values looks similar to that produced when simulating the same set of defects in Cadence Spectre. As has been mentioned above, the model is intended only for statistical evaluation of supply voltage-dependent behaviour of full open defects in benchmark circuits. The error in evaluating the behaviour for one full open defect location will be small compared to the overall observation of supply voltage-dependent behaviour over all the considered defect locations. When the victim net voltage has been determined, the logic behaviour is evaluated by comparison with the logic threshold voltages for the driven inputs, as discussed in Section 4.1.

The following example illustrate how the victim net voltage is calculated for the inverter used in Figure 4.12 with the suggested model for full opens influenced by gate tunnelling leakage.

Figure 4.13 shows how the model determines the victim net voltage V_F. Figure 4.13(a) shows the Cadence Spectre simulation results of V_F and V_D over time, when V_F is initially high. The final static voltage value is ≈0.47V. This is the voltage that the model needs to predict. In Figure 4.13(a) it can be seen that V_D changes over time, because of V_F. This observation, and the fact that Figure 4.12 showed that the leakage



(a) The victim net voltage V_F and the drain voltage V_D (b) The leakage current through the NMOS and PMOS transistors I_N, I_P

FIGURE 4.13: Example: modelling the victim net voltage in the presence of gate tunnelling leakage

current varies with V_D, is why the model has to consider V_D. Figure 4.13(b) show I_N and I_P which are gate tunnelling leakage currents for the NMOS and PMOS transistors respectively, as determined by the model. I_N is in fact an appropriate combination of the three curves in Figure 4.12(a). As can be seen from Figure 4.13(b), the leakage currents are dominated by the NMOS transistor and the final static victim net voltage V_F is estimated to ≈0.47V as shown by the border between grey and white background.

The same defect coverage metric as for the other model is used also in the presence of gate tunnelling leakage, Equation 4.2. A full open defect that is influenced by gate tunnelling leakage is covered if the behaviour predicted by the model is detected by a test pattern.

4.4 Analysis Methodology

In the previous sections of this chapter, it has been shown by theoretical examples that full open defects on interconnect may cause supply voltage-dependent behaviour and for particular cases cause supply voltage-dependent detectability. Section 4.4.1 describes a prototype simulation tool that was implemented for the purpose of generating experimental results (Section 4.5) that determine the quantitative impact of varying supply voltage on the detectability of full open defects.

4.4.1 Prototype Tool Flow

A prototype tool was developed for the purpose of analysing the impact of varying supply voltage on the behaviour and detectability of full open defects. The tool is shown in Figure 4.14 with respect to its input requirements and output data. The purpose of the tool is to facilitate analysis, by calculating the logic behaviour of a full open defect at

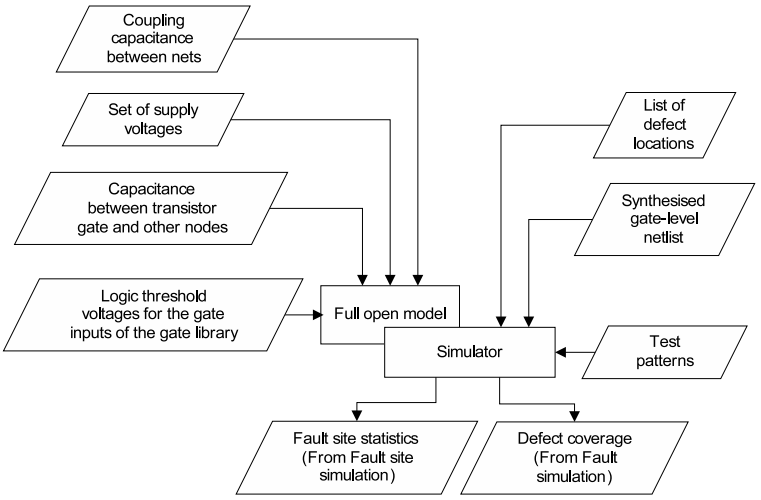


FIGURE 4.14: Tool flow to study the detectability of full open defects

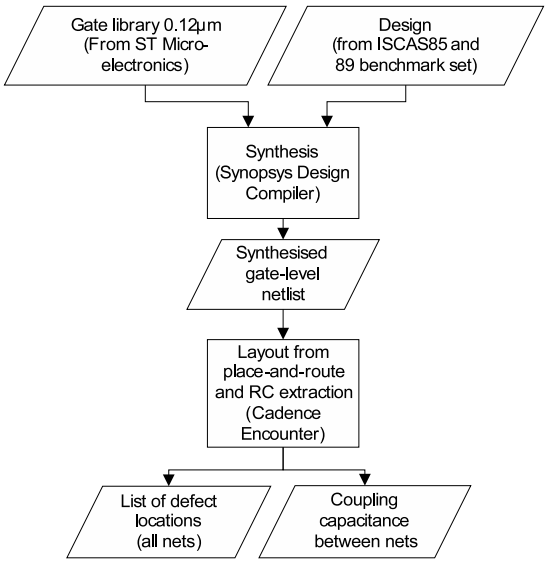
the fault site according to the models in Section 4.2 and Section 4.3, and by performing supply voltage-aware fault simulation with respect to full open defects.

The tool includes the leakage-unaware defect model (Equation 4.1) and the leakage-aware defect model (Section 4.3.3). The models make the tool aware of the behaviour of interconnect open defects and supply voltage. The simulator has two modes of operation. The first mode of operation is to calculate the logic behaviour at the fault sites of given defects. This mode is a direct application of the models and the outputs are the victim net voltage and the logic values seen by the driven inputs. The output of the tool for this mode is marked "From fault site simulation" in Figure 4.14. The other simulator mode is fault simulation, which requires a test pattern. In fault simulation mode, the simulator determines if the defect is detected or not, and calculates the defect coverage. The output of the tool for this mode is marked "Fault simulation" in Figure 4.14. The two modes are further described in Section 4.4.2 and Section 4.4.3.

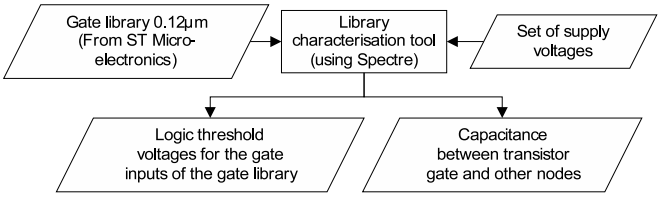
It should be noted that the tool in Figure 4.14 requires particular input data. Section 4.15 shows the steps that are used to prepare the input data. Figure 4.15(a) shows how the netlist, the coupling capacitances between nets and the list of open defect locations are prepared. Similarly, Figure 4.15(b) shows how the logic threshold voltages and capacitances to transistor nodes are determined.

The details of the preparation processes presented in Figure 4.15 are listed below.

- A synthesised gate-level netlist is prepared as follows. A Verilog gate-level description of an ISCAS benchmark circuit is synthesised for a 0.12μm gate library using Synopsys Design Compiler as shown in Figure 4.15(a). The sequential circuits of the ISCAS89 benchmark set are treated as combinatorial by assuming full scan-chains. The synthesised netlist is used as input to the main tool as shown in



(a) Synthesis and Place-and-Route



(b) Supply voltage characterisation of gate library

FIGURE 4.15: Flows used to prepare data for analysis of full open defects

Figure 4.14. It should be noted that this step is in common with the SMuVoRBAT tool flow developed for the study in Chapter 3.

- The synthesised netlist is processed using Cadence Encounter place-and-route to obtain a layout (Figure 4.15(a)).

- From the layout, capacitances are extracted using Cadence Encounter and ExtractRC. The capacitances are, for each net, coupling capacitances to neighbouring nets and capacitance to power rails (Figure 4.15(a)). For the benchmarks and the gate library used, it was found that the coupling capacitance values are in the range of 0.1aF to 6fF and the capacitances to power rails are in the range of 14aF to 6fF. The extracted capacitances are used as input in the main tool as shown in Figure 4.14. It should be noted that the layout was generated in the same way and with the same coupling capacitance extraction as in the tool flow presented in Chapter 3 and more details can be found in Section 3.2.1.

- Possible open locations should be found by analysing the layout. The process of analysing the layout for open defect location is represented in Figure 4.15(a). This process involves segmenting each net according to branches and the neighbourhood of adjacent nets as in the example of Figure 4.3. However, for the experiments presented in Section 4.5 a simpler approach is taken where only one defect per

net is considered, located at the driver of the net, even though opens may appear anywhere along the net [85], on the stem or on the branches. This choice is without loss of generality, as more defect locations would only mean more defects to consider. The list of open defect locations derived by this step is used as input to the main tool as shown in Figure 4.14.

- To find the capacitances between a floating gate input and nodes of the transistors of that input, the library characterisation tool from Section 3.2.2 was extended to perform the necessary simulations for each gate input in the gate library, employing the procedure used in the example given in Section 4.2.3. The capacitance values generated by this process for the experiments in Section 4.5 were in the range from 1fF to 7fF. The capacitances to driven transistors are used as input to the main tool as shown in Figure 4.14.

4.4.2 Fault Site Simulation

To study the influence of supply voltage variation and neighbour net assignments on the victim net voltage, the prototype tool is capable of simulating the fault site according to both the leakage unaware (capacitive coupling) model (Equation 4.1) and the model that takes gate tunnelling leakage into account (Section 4.3.3). For each defect location, the tool calculates the victim net voltage for all possible neighbour net assignments and each supply voltage setting. The logic faults at the open fault site are identified and compared over the supply voltage settings to identify supply voltage-dependent behaviour at the fault site.

For each defect location, neighbour net assignment and supply voltage, the victim net voltage is calculated and compared to logic threshold voltages. The logic behaviour found in each iteration is recorded and when all supply voltage settings have been processed, the recorded data is examined to find supply voltage-dependent behaviour.

4.4.3 Fault Simulation

The prototype simulation tool is also capable of supply voltage-aware fault simulation of full open defects. This means that it can determine if a given test pattern detects a given fault while considering the impact of supply voltage variation. The tool has been implemented by extending the supply voltage-aware resistive bridging fault simulation tool SVARFS from Section 3.2.3 to make it capable of simulating full open defects. Two main features had to be added to make the fault simulator capable of handling full open defects. Firstly, the concept of LSC (Logic State Configuration) was adapted to model the logic behaviour of full open defects. The LSC for full opens describe how the full open defect should be activated by specifying the logic value on the driving net (net D in

the example of Figure 4.1) and the logic values on the neighbouring nets (net n1 and net n2 in the example of Figure 4.1). The LSC also specify the logic behaviour seen at the inputs that are driven by the victim net, as determined by the model used for simulation. The second modification to the fault simulator software to make it able to handle full open defects is oscillation detection. If the logic value of a neighbouring net depends on the logic value for the victim net, a feedback loop is formed which can lead to oscillation. The oscillation detection feature of the fault simulator involves simulating the design a second time using the outcome of the first simulation as stimulus to the fault site. If the outcome changes, there is oscillation and detection of the defect is unpredictable. In the case of oscillation the fault is not counted as detected by the simulated test pattern.

The fault simulation tool is shown in further detail in Figure 4.16. The fault simulation takes one test pattern and applies it at a given supply voltage on a circuit with a particular defect. First, the fault site is analysed, so that a logic fault can be specified and injected into a netlist. The resulting faulty circuit is then simulated together with a fault-free circuit. The outcomes of simulating the test pattern on the two circuits are compared. If there is any difference in the primary output, that means that the injected defect was detected by the test pattern. To determine if the possibility of feedback leads to oscillation that invalidates the detection, the oscillation detection step is applied before the defect coverage is calculated.

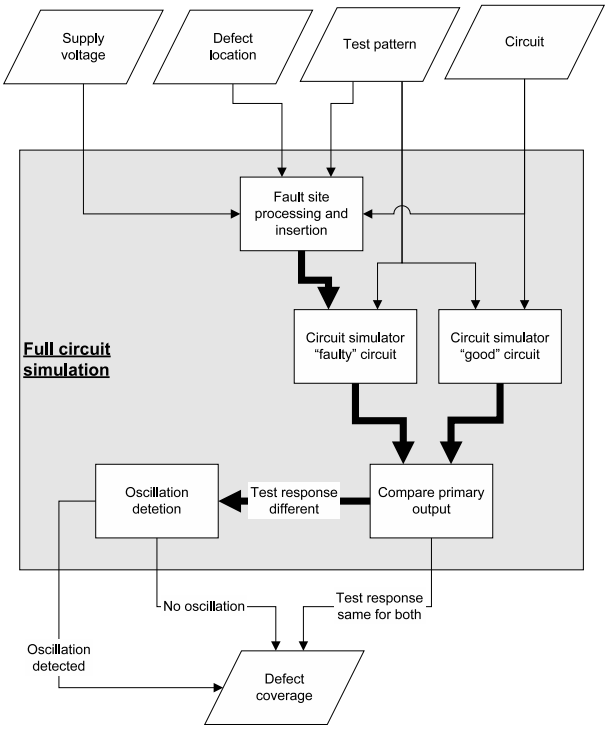


FIGURE 4.16: Fault simulation flow

The full open defect locations are considered one at a time. For each defect location, the test patterns of the test set are simulated. Each test pattern is simulated using all supply voltage settings, but the defect coverage data is recorded per supply voltage, so

that the effect of applying the test set can be determined separately for each supply voltage.

4.5 Experimental Results

Two experiments have been performed, using the two operating modes of the simulator, fault-site simulation and fault simulation, to determine the quantitative impact of varying supply voltage, firstly on the voltage on the victim net of full open defects and secondly on the detectability of full open defects.

4.5.1 The Distribution of Victim Net Voltage

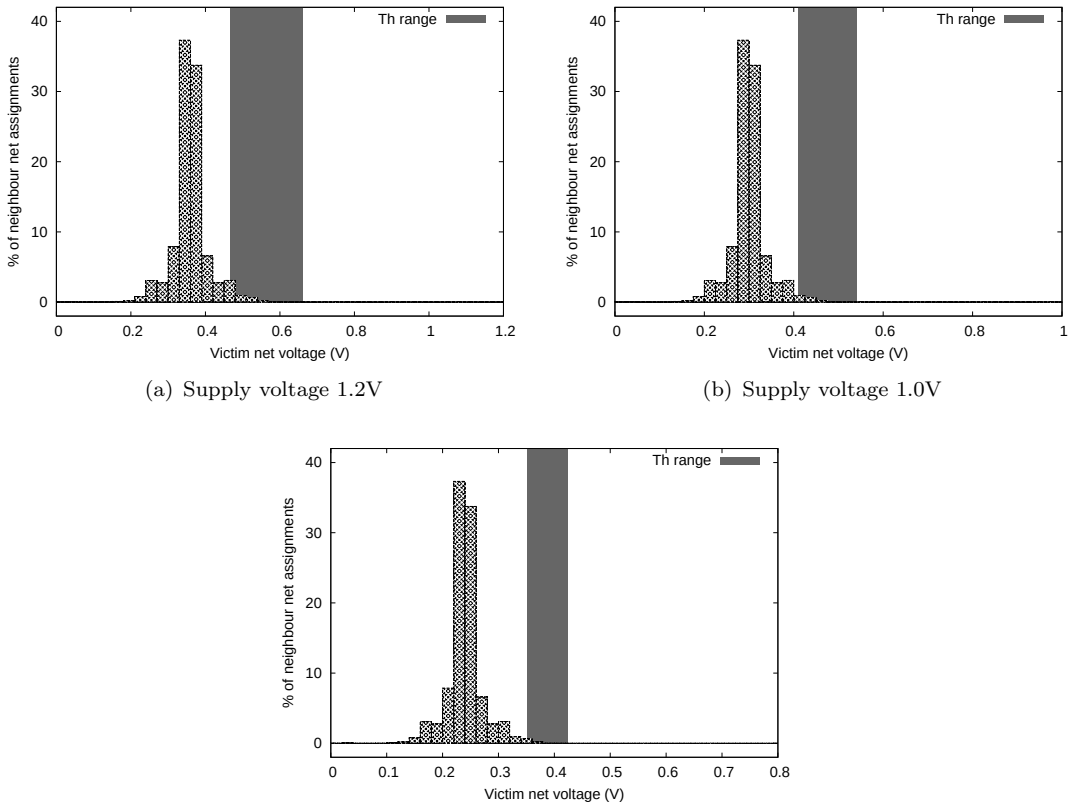
The experiment presented in this section deals with the question of what is the typical victim net voltage. To determine the victim net voltage values that are likely to occur in the presence of a full open defect, an experiment was performed on design C1355 and S641 using the fault site simulation capability of the simulation tool. In the experiment, more than 200 defect locations were considered.

4.5.1.1 Leakage Unaware Model

Experimenting with the leakage-unaware model, all possible neighbour net assignments were considered for each defect of design C1355. For each <defect,neighbour net assignment> pair, the victim net voltage was recorded and the resulting victim net voltage distribution is presented in Figure 4.17(a) for 1.2V supply voltage, in Figure 4.17(b) for 1.0V supply voltage and in Figure 4.17(c) for 0.8V supply voltage.

Figure 4.17 shows histograms of the victim net voltage distributions. The histograms should be read as follows. The horizontal axis represents the possible victim net voltage values from 0V to supply voltage. It is divided into 40 bins, each corresponding to 2.5 percent of supply voltage. Each bin has a column. The height of the column represents the number of <defect,neighbour net assignment> pairs that cause a victim net voltage that falls into the bin. For example, in the graph for supply voltage 1.0V (Figure 4.17(b)), the bin from 0.25V to 0.275V has a column that is almost 8% high. This means that almost 8% of the <defect,neighbour net assignment> pairs cause a victim net voltage that is between 0.25V and 0.275V. All neighbour net assignments of all the defects of design C1355 are considered in Figure 4.17, which gives a statistical view of the voltages that appear on the victim net.

The shaded boxes in Figure 4.17 represent the range of possible logic threshold values for the considered gate library. It should be noted that this range changes with supply



(a) Supply voltage 1.2V

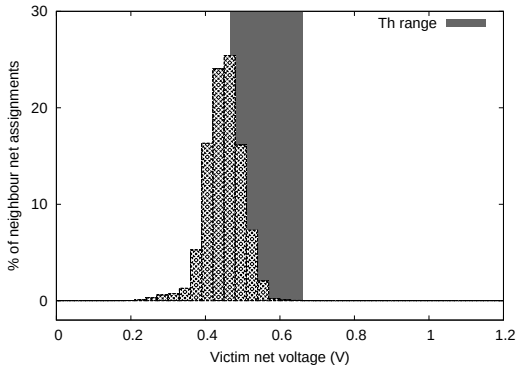
(b) Supply voltage 1.0V

(c) Supply voltage 0.8V

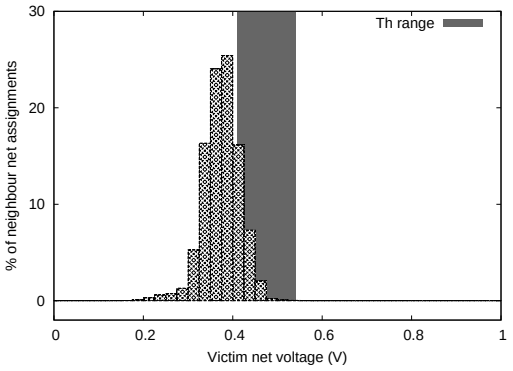
FIGURE 4.17: Distribution of victim net voltages, design C1355

voltage (Section 3.2.2). These are added to relate the information in the graph to the logic behaviour at different supply voltage settings. The victim net voltages that are less than (to the left of) the lowest logic threshold voltage appear as Logic-0. Similarly, victim net voltages above (to the right of) the highest logic threshold voltage appear as Logic-1. If the victim net voltage is outside the logic threshold voltage range for all supply voltage settings, there can be no supply voltage-dependent behaviour. Analysing this from a different perspective, assignments that cause a victim net voltage inside the logic threshold range for any of the supply voltage settings, may cause supply voltage-dependent behaviour, depending on the specific input in question. It can be seen from Figure 4.17 that only a fraction of the <defect,neighbour net assignments> pairs cause a victim net voltage that may cause supply voltage-dependent behaviour (bins inside the logic threshold voltage range).

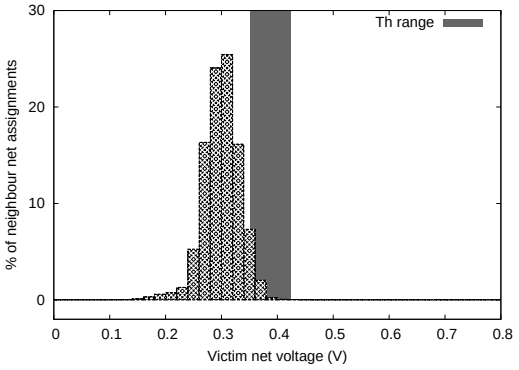
As a comparison, consider Figure 4.18, which shows the corresponding distribution for circuit S641. It should be noted that the columns for the bins inside the logic threshold voltage range are higher than the corresponding bins for C1355 (Figure 4.17). This means that circuit C1355 has less supply voltage-dependent behaviour due to opens than S641. The experiment has been carried out for 16 additional ISCAS benchmarks and the corresponding graphs show similar distributions to S641 and C1355.



(a) Supply voltage 1.2V



(b) Supply voltage 1.0V



(c) Supply voltage 0.8V

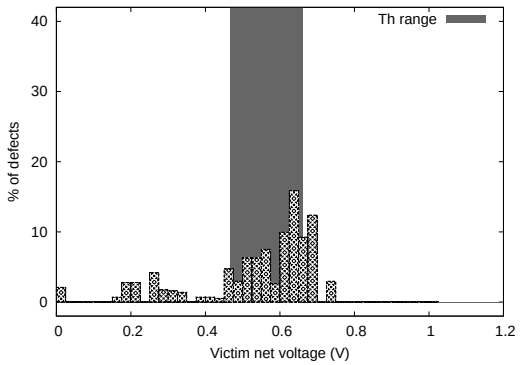
FIGURE 4.18: Distribution of victim net voltages, design S641

The results shown in Figure 4.17 and Figure 4.18 were generated assuming that $Q_{trap} = 0C$. It should be noted from Section 4.2.1 that the value of Q_{trap} for a given defect is not well known. For higher (lower) values of Q_{trap} the distribution would move to the right (left) in the figure, so that more neighbour net assignments cause a higher (lower) victim net voltage. However, varying Q_{trap} would not significantly impact the observation that on average the amount of <defect,neighbour net assignment> pairs that causes supply voltage-dependent behaviour varies from design to design. Furthermore, only a fraction of the neighbour net assignments cause a victim net voltage that corresponds to supply voltage-dependent behaviour.

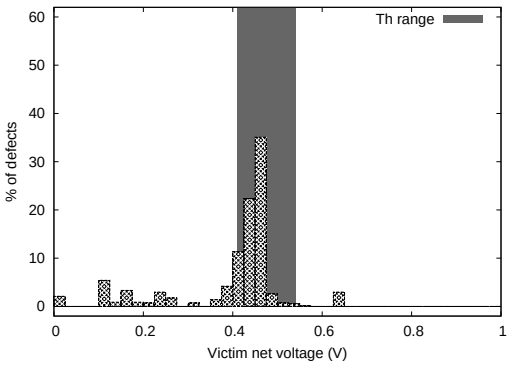
4.5.1.2 Leakage Aware Model

The same experiment as with the leakage unaware model shown above, was performed for the leakage aware model. The victim net voltage distribution graphs are shown in Figure 4.19. The experiment was performed for design C1355 for the three different supply voltage settings, 1.2V in Figure 4.19(a), 1.0V in Figure 4.19(b) and 0.8V in Figure 4.19(c).

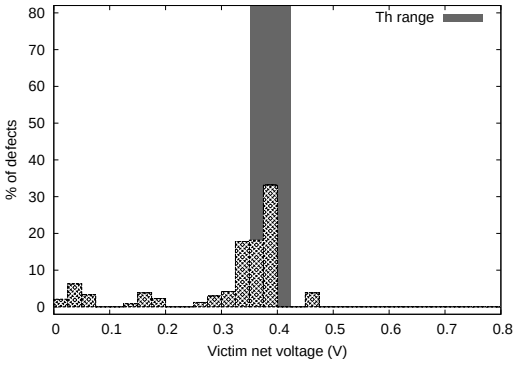
It should be noted in Figure 4.17 that the victim net voltage is widely distributed



(a) Supply voltage 1.2V



(b) Supply voltage 1.0V



(c) Supply voltage 0.8V

FIGURE 4.19: Distribution of victim net voltages using the leakage aware model, design C1355

when the full open defect is influenced by gate tunnelling leakage. As discussed in Section 4.3.3, these results were generated assuming that 30% of the inputs are cut off from the supply voltage power rail and that 30% are cut off from ground. Experiments have been conducted with different values, other than 30%, and it was found that a low percentage leads to a very narrow range of victim net voltages, whereas a large percentage range leads to a wide spread of victim net voltages.

Comparing the victim net voltage distribution of the leakage-aware model in Figure 4.19 with the victim net voltage distribution of the leakage-unaware model in Figure 4.17, it can be seen that in a scenario that is influenced by gate tunnelling leakage, there are relatively more defects with a victim net voltage inside the range of possible logic threshold voltages than in a scenario that is not influenced by leakage. This suggests that full open defects with gate tunnelling leakage are more sensitive to supply voltage variation. Particularly for low supply voltage settings, it can be difficult to predict the logic behaviour of a full open defect that is influenced by gate leakage, because the victim net voltage is so close to the logic threshold voltage.

TABLE 4.5: Defects with supply voltage-dependent neighbour assignments

		Leakage Unaware	Leakage Aware
Design	**Total Defects**	**Defects with Vdd dep.**	**Defects with Vdd dep.**
C432	123	16 (13%)	43 (34%)
C499	197	14 (7%)	88 (44%)
C880	222	21 (9%)	76 (34%)
C1355	236	13 (6%)	91 (38%)
C1908	214	14 (7%)	94 (43%)
C2670	472	76 (16%)	145 (30%)
C3540	468	52 (11%)	181 (38%)
C5315	623	103 (17%)	280 (44%)
C7552	887	123 (14%)	364 (41%)
S641	107	20 (19%)	33 (30%)
S1488	290	56 (19%)	129 (44%)
S5378	634	115 (18%)	249 (39%)
S9234	483	92 (19%)	189 (39%)

4.5.2 The Quantity of Defects with Supply Voltage-Dependent Behaviour

To work out how many defects in a given circuit that show supply voltage-dependent behaviour, Table 4.5 was generated, with the simulation tool in fault-site simulation mode. Columns one and two give the benchmark name and the total number of defects. Column three gives the number and percentage of defects that have supply voltage-dependence for the leakage unaware model. Similarly, column four gives the number and percentage of defects that have supply voltage-dependence for the leakage aware model. As can be seen, the defects that have supply voltage-dependence range from 6% in case of C1355 to 19% in the case of S9234 for defects that are not influenced by gate leakage (marked Leakage Unaware) and from 30% in case of C2670 to 44% in the case of S1488 for defects that are influenced by gate leakage (marked Leakage Aware). From Table 4.5 it can be seen that full opens have more supply voltage-dependent behaviour in the presence of gate tunnelling leakage than in a scenario without leakage.

Table 4.5 shows that many full open defects have some supply voltage-dependent behaviour in the scenario without gate leakage (marked Leakage Unaware in Table 4.5) where the behaviour depends on the capacitive coupling to neighbouring nodes. However, defects are counted in Table 4.5 if at least one neighbour net assignment causes supply voltage-dependent behaviour. If there is more that one possible neighbour net assignment and one of them does not cause supply voltage-dependent behaviour, that assignment can be used to make a test pattern to cover the defect independent of the supply voltage. To be able to determine how many of the neighbour net assignments cause supply voltage-dependent behaviour, a *Vdd-dependency factor* (VDF) for a de-

fect is introduced. The Vdd-dependency factor is defined as the number of neighbour net assignments that causes supply voltage-dependent behaviour divided by the total number of neighbour net assignments for that defect. A Vdd-dependency factor of zero means that the defect has no supply voltage-dependent behaviour. A Vdd-dependency of one would mean that all neighbour net assignment cause supply voltage-dependent behaviour. Such defects could require a particular supply voltage while testing. Table 4.6 shows the average and maximum Vdd-dependency factor for the supply voltage-dependent defects of the benchmark circuits. The values are calculated using the two expressions at the bottom of the table. The low average Vdd-dependency factors mean that most of the defects with supply voltage-dependent behaviour can be detected using any supply voltage, because there are many neighbour net assignments to test with for which the logic behaviour is independent of supply voltage. However, it can be seen from column four that for some circuits there is at least one defect that has supply voltage-dependent behaviour more than half (Vdd-dependency factor 0.50) of the assignments (circuit C880, C3540, S1488 and S5378). These defects with high Vdd-dependency factor are of interest because there is supply voltage dependent behaviour for most of the neighbour net assignments. It is possible that constraints from the surrounding circuitry (as in the example of Figure 4.4) limits the set of neighbour net assignments that can occur, so that only those which cause supply voltage dependent behaviour can occur. If there are such constraints from the surrounding circuitry, these defects may require testing at a particular supply voltage.

TABLE 4.6: Vdd dependency factor for supply voltage-dependent defects (leakage unaware model)

Design	**Defects with Vdd dep.**	**Vdd dependency factor**	
		Average	**Maximum**
C432	16	0.13	0.25
C499	14	0.14	0.25
C880	21	0.17	0.75
C1355	13	0.15	0.37
C1908	14	0.08	0.18
C2670	76	0.10	0.37
C3540	52	0.14	0.57
C5315	103	0.11	0.31
C7552	123	0.11	0.39
S641	20	0.15	0.31
S1488	56	0.11	0.53
S5378	115	0.11	0.62
S9234	92	0.09	0.34

Average:

$$\frac{\sum_{d \in defects} VDF_d}{total\ no.\ of\ defects}$$

Maximum:

$$\max\{VDF_d | d \in defects\}$$

4.5.3 Supply Voltage-Dependent Detectability

The second experiment performed fault simulation to determine the impact of supply voltage variation on the defect coverage when testing for interconnect full open defects. As was mentioned in Section 4.2.1, the amount of trapped charge on the victim net Q_{trap} is unknown. Therefore, this experiment uses a random but fixed value for $V0 = \frac{Q_{trap}}{C_{FGND}}$, taken from the range [-0.4V,0.6V], for each net. Because of this, the trapped charge will have a large random influence on the victim net voltage. However, the value for $V0$ was kept constant as the supply voltage was varied in the experiment. Therefore, a study that considers many full open defects can see an indication of the effect of supply voltage variation on detectability even in the presence of the random influence of the trapped charge.

To evaluate the detectability of full open defects for different supply voltage settings, this experiment simulated 1000 pre-generated pseudo-random test patterns. Experiments were performed with various test set sizes and the overall conclusions in terms of the supply voltage-dependent detectability are similar, as is discussed below. The 1000 test patterns achieve high defect coverage overall for the same reason as N-detect tests are effective (Section 4.1.2). With such a large test set, it is likely that variations in the results are the cases that require specific supply voltage settings for test. The defect coverage is calculated as shown in Equation 4.2.

Table 4.7 shows the defect coverage achieved when applying 1000 pseudo-random test patterns to a set of ISCAS benchmark designs. The first two columns show the name of the benchmark design and the number of considered full open defects. The next three columns show the defect coverage achieved using the leakage-unaware model at three supply voltage settings, 1.2V, 1.0V and 0.8V respectively. The last three columns show the defect coverage for the supply voltage settings when gate leakage is taken into account. It should be noted that the defect coverage values reported in Table 4.7 refer to the set of considered defects and the amount of these are shown in column 2. The set of considered defects may contain completely undetectable defects (defects that cannot be detected by any test pattern). Considering column 3-5, circuit C432 has 123 defects. When the test is applied at 1.2V the defect coverage is 98.4% (121 out of 123 defects). The same test-set is applied at 1.0V, and 0.8V (column five and six) with the same result. As can be seen from Table 4.7, the defect coverage has little dependence on supply voltage as the defect coverage remains very high, without much change when tests are carried out at different supply voltage settings. This is true for both models. The fact that the defect coverage is found to be high with 1000 pseudo-random test patterns as shown in Table 4.7 indicates that a the majority of full opens are highly detectable, which is in line with previous studies, that used tests generated without consideration of the analog behaviour of full open defects and achieved high defect coverage (Section 2.2.4). The high detectability of full opens can be explained by

TABLE 4.7: Defect coverage for pseudo-random test patterns

Design	Defects	Defect coverage					
		Leakage Unaware			Leakage Aware		
		1.2V	1.0V	0.8V	1.2V	1.0V	0.8V
C432	123	98.4	98.4	98.4	98.4	98.4	98.4
C499	197	99.5	99.5	99.5	99.5	99.5	99.5
C880	222	99.6	99.6	99.6	99.6	99.6	99.6
C1355	236	98.7	98.7	98.7	97.4	97.0	97.9
C1908	214	99.5	99.5	99.5	99.1	99.5	99.1
C2670	472	86.4	86.2	86.2	86.7	86.7	86.7
C3540	468	99.1	99.1	99.1	99.3	98.9	99.1
C5315	623	99.8	99.8	99.8	100	100	100
C7552	887	97.0	97.0	97.0	97.1	97.3	97.3
S641	107	99.1	99.1	99.1	98.1	98.1	98.1
S1488	290	99.3	99.3	99.3	99.3	99.3	99.7
S5378	634	97.6	97.5	97.5	96.5	97.3	97.2
S9234	483	92.3	92.5	92.5	92.5	90.5	90.3

the fact that the defect behaviour is independent of the driver, from which the victim net is separated. The high detectability is demonstrated by high defect coverage for several designs. However, in a few circuits the defect coverage is affected slightly by supply voltage. Underlined results are examples of this. For example C2670, in the column marked "Leakage Unaware", has defect coverage 86.4% for 1.2V and 86.2% for 1.0V and 0.8V supply voltage.

The results with the leakage-aware model (columns 6-8) were generated with 30% of the inputs cut off from power supply and 30% of the inputs cut off from ground as discussed in Section 4.3.3. Experiments have been conducted with different values, other than 30%, and the percentage did not influence the overal conclusions regarding the supply voltage dependent detectability of full open defects that are influenced by gate tunnelling leakage. The numbers in the table did vary with the percentage, but the defect coverage remained high and no trend regarding which supply voltage to prefer when testing for full opens appeared.

4.5.4 Summary

In summary, some full open defects show supply voltage-dependency which leads to faulty behaviour (Table 4.5), but the defect coverage remains high independent of the supply voltage used while testing (Table 4.7). This indicates that for most full open defects there exists at least one test pattern that can cover it for a given supply voltage setting. Even if a defect does not show faulty behaviour at the considered supply voltage for a particular test pattern, there may be another test pattern for which the defect shows

faulty behaviour. The results in Table 4.7 indicate that this is true for most full open defects.

The study reveals that for open defects, such that the behaviour of the defect is independent of the driver, the defect is relatively easy to detect, because a test can be constructed so that if the defect exposes a faulty Logic-1 (Logic-0) it is detected by a test that puts Logic-0 (Logic-1) on the driving net. This observation is true for a majority of the simulated full open defects. Furthermore, this easy-to-detect property is independent on the source of influence on the victim net voltage, be it supply voltage, gate tunnelling leakage or capacitive coupling as long as the test is performed in a static context.

4.6 Concluding Remarks

A full open is a complete break between two circuit nodes that should be connected and is an important defect type in deep submicron designs. This chapter has explicitly studied the impact of supply voltage variation on the static behaviour of interconnect full open defects in the context of circuits that use multiple supply voltage settings, as is the case in many low power designs. This chapter investigated how supply voltage variation impacts the detectability of full open defects, as supply voltage dependent behaviour in such defects lead to situations where particular supply voltage settings are required for testing to achieve full defect coverage. In this context, previous studies [79, 134] have shown that the related defects called resistive opens are better detectable using an elevated supply voltage, but for full open defects, only one study [123] had previously indicated that the behaviour of full open defects depends on the supply voltage. To study such behaviour in detail, this chapter performed analysis using two models for the voltage on the node that is left floating by the full open defect. The first model considered the influence of circuitry physically close to the defect through capacitive coupling. The second model considered the influence of gate tunnelling leakage currents from transistors that are driven by the affected node.

To enable analysis on synthesised benchmark circuits, a fault simulation tool was developed that make use of the considered models for the behaviour of full open defects. Experiments were performed using the fault simulator tool and ISCAS benchmark designs that were synthesised to a 0.12μm gate library. The capacitance values required for analysis were extracted from layout. The experimental results showed that the logic behaviour depends on supply voltage for many full open defects (with or without the influence of gate tunnelling leakage) and many test patterns. Typically, a full open defect can be targeted by several different test patterns and for some of these test patterns, the full open defect causes supply voltage-dependent behaviour. Analysis showed that the real concern is for full open defects for which only one test pattern exists. Such full

open defects may manifest faulty behaviour only at particular supply voltage settings, leading to a need for supply-voltage aware test generation for full open defects. However, the full open defects that only manifest for particular supply voltage settings only make a marginal impact on defect coverage as demonstrated by experimental results. This observation indicates that most full open defects can be covered by at least one test pattern for a given supply voltage setting. Therefore, the majority of full open defects can be covered using a single arbitrarily chosen supply voltage and the benefit of supply voltage-aware test generation and testing using several supply voltages is only marginal. This observation was made using both the considered models of the full open defect behaviour.

Chapter 5

Process Variation-Aware Testing for Resistive Bridge Defects

Process variation is emerging as a challenge to deep submicron CMOS technology IC design [142, 24, 143] and recent research [151, 149, 150] has reported negative impact on manufacturing test quality. In particular, as it will be shown in this chapter, process variation influences the behaviour of resistive bridging faults (RBF), which is a major defect type in CMOS [64]. Addressing the issue of process variation and its impact of testing for RBFs is highly relevant in this thesis, because the work in Chapter 3 presented a test generation method for RBFs based on the parametric bridging fault model [112], which does not model the influence of process variation. The parametric bridging fault model relies on fixed and known values for the logic threshold voltage of each gate input and for the drive strength of each gate. These two parameters, logic threshold voltage and gate drive strength, vary from gate to gate due to process variation and consequently affect the bridge behaviour, as will be shown in Section 5.2. The aim of this chapter is to analyse the impact of process variation on the quality of tests for RBFs, quantify this impact and to propose a process variation-aware test generation method to ensure high test quality in the presence of process variation.

As mentioned in Section 1.3, process variation is usually considered either as variation across different dies or as variation within each die [24, 22]. In this chapter, process variation is considered within each die, influencing each transistor. Process variation is caused by several effects, in particular random dopant distribution [21, 32], line edge roughness and issues associated with sub-wavelength lithography, and cause variation in threshold voltage (VT), gate oxide (or dielectric) thickness (TOX) and transistor gate geometry (width W and length L) [24, 22]. This chapter shows how variation of VT, TOX, W and L affect the behaviour of RBFs through two parameters: (1) the gate drive strength and (2) the logic threshold voltage. This chapter explains how tests that are generated without consideration of process variation fail to detect some of the

logic faults that are induced by process variation. Tests suffer loss of defect coverage because process variation leads to logic faults that are not detected by the tests. Defects that pass the test undetected because they correspond to logic faults that the test fails to detect due to process variation are called test escapes. The detrimental impact of process variation through such test escapes on test quality is quantified using a novel metric called test robustness.

As will be shown in this chapter, test sets generated without consideration of process variation fail to provide adequate test quality. Therefore, there is a need for process variation-aware test generation. The challenge in process variation-aware test generation is that the unique effect of process variation on a given die cannot be predicted. Consequently, to achieve full defect coverage, a test set must create all possible activation conditions (all possible input combinations that cause the bridged nets to be driven to opposite logic values) and all possible logic behaviours at the gate inputs that are driven by the bridged nets, as suggested in [97]. In other words, such a test set must target all possible logic faults for each bridge location. It should be noted that the approach in [97] does not consider the probability of different logic faults to occur. A fundamentally different approach is taken in this chapter, by employing the observation that it is not necessary to target all possible logic faults, but instead the aim should be to cover all detectable bridge defect resistances. To ensure high defect resistance coverage, the suggested approach analyses the probable behaviour of each bridge in the presence of process variation and generates test patterns accordingly. The effect of process variation on the bridge behaviour is taken into account by targeting the logic faults that (if not detected) are most probable to cause loss in defect coverage.

This chapter provides experimental results that are part of the analysis of the impact of process variation on test quality for RBFs and that demonstrate the proposed process variation-aware test generation method. The experiments are performed for synthesised and placed-and-routed ISCAS85 and -89 benchmark circuits using a 45nm gate library [15]. Realistic bridge locations and bridge defect probabilities are extracted from layout.

The chapter is structured as follows:

- Section 5.1 gives background information and summarises prior work on process variation, RBF behaviour and testing in the presence of process variation.
- Section 5.2 shows how process variation influence RBF behaviour through two parameters, (1) shift in logic threshold voltage and (2) shift in gate drive strength. The mechanism behind the behaviour is analysed and it is shown how process variation induce logic faults that escape a test that is generated based on nominal process parameters.

- Section 5.3 proposes a metric, test robustness, for measuring the impact of process variation on the test quality of a given test. The metric is explained and demonstrated with an example.
- Section 5.4 proposes a process variation aware test generation methodology (PVAA) that is able to produce a test set that achieves a user-specified test robustness level.
- Section 5.5 provides experimental results that demonstrate the test robustness metric and the process variation-aware test generation method.
- Concluding remarks are given in Section 5.6.

5.1 Background and Prior Work

As can be seen in Section 2.4, several studies have explored the impact of process variation on delay [30, 147, 146], power supply current [145, 152, 153], frequency of ring oscillators [144] and testing analog ICs [154]. Recent research has reported that process variation has a negative impact on manufacturing test quality, including the studies in [146, 147, 151, 149, 148, 150] which considered the effect of process variation on delay fault testing. In this chapter, the focus is on the impact of process variation on logic testing, i.e. the behaviour of a static defect type, resistive bridging faults (RBFs). The following provides a more detailed review of relevant research than was given in Section 2.4, to put the previous studies into context of the research presented in this chapter and to make the chapter self-contained. This includes fault models and test methods that consider process variation and resistive bridging faults.

A new bridging fault model and a fault simulator for bridging faults and process variation was presented in [99]. The fault model is independent of circuit parameters like the bridge defect resistance and only the logic behaviour is considered, motivated by the need for fast fault simulation. This simulation methodology is incompatible with the parametric bridging fault model, which relies on logging of the bridge defect resistance values for each logic fault. To find the true impact of process variation on testing resistive bridging faults, it is necessary to take the bridge resistance into account, as will be shown in Section 5.2. Therefore, the research presented in this chapter employs another approach and considers in detail the bridge resistance and parameters such as transistor threshold voltage (VT), gate oxide thickness (TOX), transistor geometry (W,L), logic threshold voltage (Th) of gate inputs and gate drive strength (g, gate output conductance). The approach enables analysis of the influence of process variation on the logic behaviour of resistive bridges. The analysis includes the probability for each process variation induced logic fault and includes how the relationship between logic behaviour and bridge resistance varies with the above mentioned IC parameters. As a part of the analysis, a metric called test robustness is defined and used to quantify the impact

of process variation on the test quality. Such analysis was not performed in [99]. To perform this analysis, it is important to understand how process parameters vary in IC fabrication. In this topic, some relevant work has been mentioned in Section 1.3. It is often assumed that the transistor threshold voltage (VT) and the width (W) and length (L) of a transistor have Gaussian distributions under process variation [152]. A detailed study on the influence of dopant distribution [21] observed that the transistor threshold voltage distribution is bell-shaped like a Gaussian distribution. Based on these reports [152,21] and the fact that a sum of independent random variables with Gaussian distribution also has a Gaussian distribution (central limit theorem [162]), it is assumed that the logic threshold voltage variation of a gate input and the voltage on the bridged nets as resulting from gate drive strength variation both can be represented by Gaussian distributions.

Studies on test generation have suggested an approach that aims to apply all possible input assignments to the gates involved in a fault site, and to propagate through all gates that have the fault site as input, to detect a large set of logic faults including both modelled and unmodelled defects [42]. Some such studies investigate resistive bridging faults [106, 97, 155]. By applying all possible input assignments to the gates that drive the bridged nets and propagating through the gates that are driven by the bridge nets [106,97] it is not necessary to consider actual values of IC parameters, because all possible bridge-related logic faults are investigated. In [97] it was shown that tests for resistive bridges can be generated such that they are valid independent of process variation, by applying the above mentioned approach. The test generation method in [155] employs a similar technique, explicitly aimed to cope with process variation. However, these tests (from [97] or [155]) do not consider the probability for the logic faults to occur. Therefore, the generated tests may target logic faults that are unlikely, leading to an unnecessarily large test set. On the topic of how many test patterns that are needed to achieve acceptable test quality for resistive bridges in the presence of process variation, it was shown in [99] that a test generated for nominal parameter values allows a significant number of defects to escape when the device-under-test is influenced by process variation. In a subsequent report [155] it was shown that a single test pattern is not adequate to ensure acceptable detection probability for a bridging fault and that there is an upper bound to the number of required test patterns to achieve variation-independent detection. From these two studies [99,155], it can be seen that test generation for resistive bridging faults in the presence of process variation is non-trivial, but that if the appropriate test patterns can be found, the task can be accomplished with a reasonable test set size.

In contrast to previous methodologies, the research presented in this chapter does not aim to cope with process variation by applying a very large number of test patterns nor to provide a test that is guaranteeing defect detection for all possible configurations of process parameter values. Instead, it is the aim of this study to generate tests for the

most probable logic faults so that an adequate probabilistic test quality level is achieved in the presence of process variation. This chapter presents a test generation approach which targets the logic faults that are most likely to occur, while giving the user of the test generation tool the opportunity to trade off test quality and test set size. The proposed test generation approach exploits the observation that some logic faults are more likely to occur than other logic faults. Furthermore, some logic faults correspond to large ranges of bridge defect resistance. By targeting the logic faults that combine high probability and large amounts of bridge resistance, the method generates test patterns that are effective in detecting bridge defects in the presence of process variation.

5.2 Motivation

Addressing the task of analysing the impact of process variation on manufacturing test of resistive bridging faults, this section explains the behaviour of a resistive bridging fault as it is influenced by three parameters. These three parameters are the bridge resistance value, the drive strength of gates driving the bridged nets and the logic threshold voltages of gate inputs that are driven by the bridged nets. A fourth parameter that influences bridge behaviour is the supply voltage, which is discussed in Chapter 3. This section explains how process variation affects the behaviour of the circuit in the presence of a resistive bridging faults and shows how such process variation can lead to test escape, i.e. reduction in test quality.

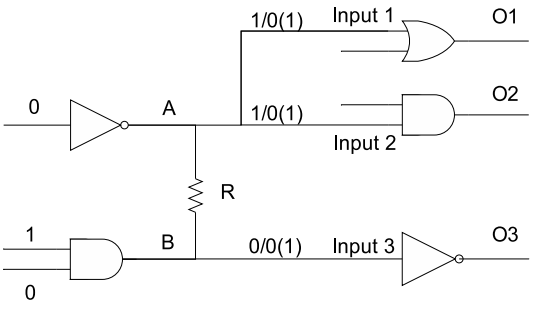


FIGURE 5.1: Example bridge location

A typical bridging fault site is shown in Figure 5.1 (Appendix A defines the concept of a fault site). Bridges are unintended resistive connections between two nodes. The following analysis is restricted to the subset of bridge locations that connects two signal nets and does not cause feedback. The restriction is employed for simplicity, but the considered bridges represent a significant set of defects, a fact that lends weight to the analysis as a first study on how process variation impacts quality of tests for resistive bridge defects. In the bridge location in Figure 5.1, nets A and B are bridged by a defect with resistance R. When net A and net B are driven to opposite logic values, the bridge affects the voltages on the bridged nets. For example, it may happen that the defect pull down the voltage on net A from the intended Logic-1 (power supply voltage level)

to a voltage which is seen by Input 1 and Input 2 as a faulty Logic-0 (marked with 1/0 in Figure 5.1). At the same time, the defect affects the voltage on net B so that it is not the intended Logic-0 voltage (close to ground potential). In this example the voltage on net B is still seen by Input 3 as a correct Logic-0 (marked with 0/0 in Figure 5.1). Due to the faulty Logic-0 on Input 1 and Input 2, a test may propagate the faulty signal through Input 1 to output O1 and detect the bridge. For the sake of the example, call the test that detects the defect by propagating through output O1 the test T1. However, T1 fails to detect the defect if process variation changes the logic value seen by Input 1. Consider the logic values that are marked with parentheses in Figure 5.1 as a logic behaviour induced by process variation. This process variation induced logic behaviour exposes a correct Logic-1 on Input 1 and Input 2, and instead Input 3 sees a faulty Logic-1. Test T1 would not detect the defect for the process variation induced logic behaviour because the logic value seen at Input 1 is fault-free. In this example, it can be seen that if process variation cause the logic behaviour of a RBF to change that leads to logic faults that are not detected by a test like T1, that has been generated without consideration of process variation.

The voltages on the bridged nets depend on the bridge resistance R [104], as shown by Figure 5.2. For sufficiently high values of resistance, V(A) and V(B) are close to the intended values, close to Vdd and ground respectively. On the other hand, for a 0Ω bridge, nets A and B have the same voltage. The voltage at R=0Ω is due to the balance in drive strength between the gates that drive the bridged nets, $V(A) = \frac{1/g_{AND}}{1/g_{INV}+1/g_{AND}} \cdot Vdd$. Therefore, it can be seen that the voltage on the bridged nets depends on both the bridge resistance R and the gate drive strength balance. Figure 5.2 also shows, as horizontal lines, the logic threshold voltages (Th) for the inputs that are driven by nets A and B (Th1, Th2 and Th3 for Input 1, Input 2 and Input 3 respectively). Logic faults (LF) are shown as grey boxes and critical resistances (CR) are shown on the horizontal axis.

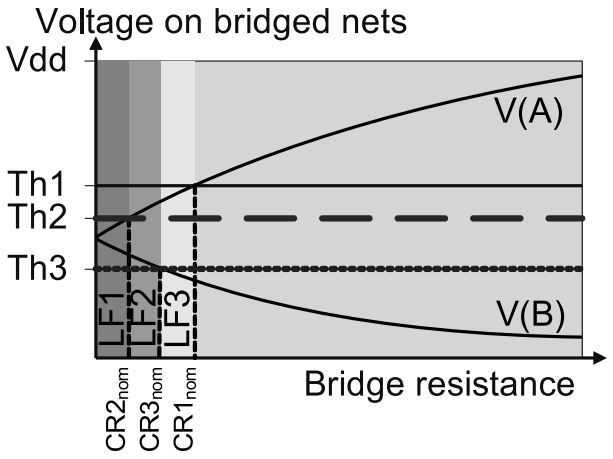


FIGURE 5.2: Nominal parameter behaviour of example bridge

The logic behaviour depends on the input-to-output transfer characteristics of the driven

gate inputs and the voltages on the bridged nets, which in turn depend on the inputs applied to the driving gates and the resistance on the bridge. The input-to-output transfer characteristic is simplified in this analysis by assuming a well defined logic threshold voltage (Th) for each input (Section 3.2.2). The logic threshold is the input voltage at which the gate changes its output behaviour (not to be confused with transistor threshold, VT). Input voltage above the logic threshold is seen as Logic-1, otherwise as Logic-0. In Figure 5.1, net A drives Input 1 and Input 2 with logic threshold Th1 and Th2 respectively, and net B drives Input 3 with logic threshold Th3. In Figure 5.2, V(A) is below Th1 in the resistance range [0,$CR1_{nom}$]. That means that for defect resistances in that range Input 1 will see Logic-0, which is the faulty value, and for the remaining resistance [$CR1_{nom}$,∞] Input 1 will see Logic-1. Similarly, Input 2 sees faulty Logic-0 for defects in [0,$CR2_{nom}$] and Input 3 sees faulty Logic-1 in [0,$CR3_{nom}$]. The resistances $CR1_{nom}$, $CR2_{nom}$ and $CR3_{nom}$ mark changes in the logic behaviour and are called critical resistances [111]. Resistances between the critical resistances cause the faulty logic behaviours LF1={0,0,1} (the notation is {L(1),L(2),L(3)} where L(i) is the logic value seen by Input i) for [0,$CR2_{nom}$], LF2={0,1,1} for [$CR2_{nom}$,$CR3_{nom}$] and LF3={0,1,0} for [$CR3_{nom}$,$CR1_{nom}$]. From this point on, Figure 5.2 is called the nominal scenario.

From the behaviour of resistive bridges described above, it should be noted that the logic behaviour depends on the resistance of the defect. In other words, a test that detects a logic fault covers a range of defect resistances. For this reason, this work employs the parametric bridging fault model [104], which considers the bridge resistance ranges for each logic fault. It should be noted that the parametric bridging fault model is not process variation-aware. In this work the fault model is employed in the context of additional consideration of process variation as will be explained in Section 5.3. The defect coverage of the parametric bridging fault model is expressed in terms of Covered Analogue Detectability Interval (CADI) and Global Analogue Detectability Interval (GADI), representing the covered and detectable defect resistance respectively [104,107]. The defect coverage DC for a bridge location b and a test T is given by:

$$DC(b,T) = \frac{\|CADI(b,T)\|}{\|GADI(b)\|} \tag{5.1}$$

5.2.1 Bridging Fault Analysis in the Presence of Process Variation

To analyse the behaviour of resistive bridges in the presence of process variation, SPICE-type simulations were performed for 45nm technology [15] with transistor models from [160]. It was found that variation in parameters such as transistor threshold voltage (VT), transistor geometry (W,L) and gate oxide thickness (TOX) gives rise to variation in the following two parameters: gate drive strength (g, gate output conductance) and logic threshold voltage (Th). As was shown in Section 5.2, these two parameters influence the logic behaviour of RBFs. The behaviour of RBFs is also dependent on the input assign-

ment to the gates that drive the bridged nets, temperature [116], supply voltage [114] and (for feedback bridges) previous logic states of the circuit [55,59]. However, only gate drive strength and logic threshold voltage are varying with IC parameters. Therefore the analysis considers variation on logic threshold voltage and gate drive strength. Process variation influences the behaviour of resistive bridges by affecting gate drive strength and logic threshold voltage as will be shown next.

The impact of gate drive strength shift on resistive bridges was investigated by performing Monte-Carlo simulation. The simulation was performed while varying the length of the transistors in the gates that drive the bridged nets according to a Gaussian distribution with mean μ=45nm and standard deviation σ=5nm, to model the effect of line edge roughness on the effective transistor length. The standard deviation is chosen based on the fact that the edge roughness of poly lines is typically on the order of 5nm and that as the line width is scaled down, for example from 90nm technology to 45nm technology, the roughness on the edge of the line does not scale [23]. Gaussian distribution was used to approximate the variation of line edge roughness, as in [23]. For the bridge in Figure 5.1, the graph in Figure 5.3 shows how the voltage on the bridged nets A and B vary with process variation and with the defect resistance R (horizontal axis). Shown as black lines in Figure 5.3 are the mean values of V(A) and V(B) for each resistance value. The voltage varies more for small bridge resistances (left side of the graph) compared with the voltage for higher bridge resistances. This is true for both V(A) and V(B). These voltages depend on the gate drive strength for the gates that drive the bridged nets, particularly for low bridge resistances. This means that the gate drive strength balance is affected by the variation on the transistor lengths.

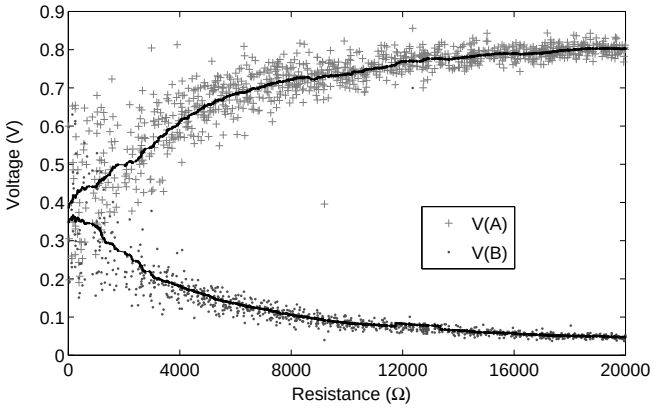


FIGURE 5.3: V(A) and V(B) for parameter samples and defect resistances from a Monte-Carlo simulation

Figure 5.4 shows one of the parameter configurations from the Monte-Carlo simulation, where process variation increased the drive strength of the gate that is driving high and elevated the voltages on the bridged nets. The increase in voltage is 0.025V (from 0.4V to 0.425V) for 0Ω bridge resistance (the left-most edge of the graph in Figure 5.4) and

decreases for increasing values of resistance. This shift is due to a reduced gate length on the PMOS transistor of the inverter, from 45nm to 44nm. The shift made the inverter stronger in driving high, resulting in increased voltage on the bridged nets. This shows that process variation may change the drive strength of a gate and consequently affect the voltages on the bridged nets.

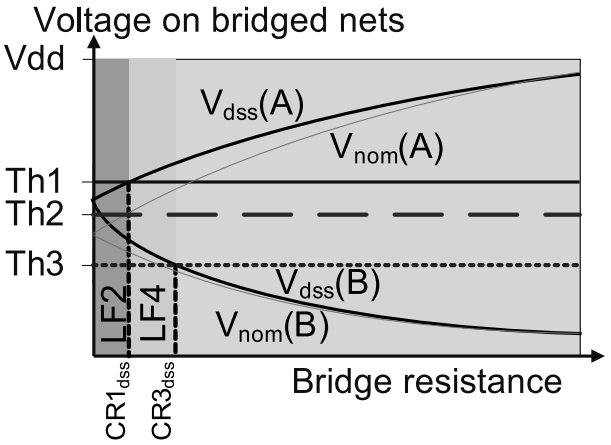


FIGURE 5.4: Shift in the drive strength balance of driving gates

Figure 5.4 shows how the voltage on nets A and B depend on R in the scenario of a shift in gate drive strength and the resulting logic faults LF2 and LF4. In Figure 5.4, the voltages have increased from $V_{nom}(A)$ and $V_{nom}(B)$ to $V_{dss}(A)$ and $V_{dss}(B)$ (Drive Strength Shift dss) and a new logic fault LF4 is introduced, which did not occur in the nominal scenario Figure 5.2. The faulty logic behaviours are LF2 in [0,$CR1_{dss}$] and LF4=$\{1,1,1\}$ in [$CR1_{dss}$,$CR3_{dss}$]. This shows that process variation, by drive strength shift, changes the logic behaviour of a bridge.

Similarly, the impact of logic threshold shift on the behaviour of resistive bridges was investigated. Monte-Carlo simulation was performed on an AND gate, such as the one that is driven by net A in Figure 5.1, while varying VT for the NMOS transistor that is closest to the second input, with a Gaussian distribution (mean μ=0.471V and standard deviation σ=0.045V). It was found that the logic threshold voltage had a bell-shaped (approximately Gaussian) distribution with mean 0.42V and standard deviation 0.05V. This distribution of logic threshold voltages from the Monte-Carlo simulation is shown in Figure 5.5.

It can be seen in Figure 5.5, that varying the transistor threshold voltage VT for the NMOS of the second input of an AND gate, causes variation in the logic threshold voltage Th. One of the parameter value configurations in Figure 5.5 is further analysed in Figure 5.6 to determine its impact. Figure 5.6 shows how increasing Th2 from 0.42V to 0.455V affects the bridge behaviour. Such influence on Th2 can be achieved by a shift in VT of the NMOS transistor closest to Input 2, from 0.471V (nominal value) to 0.59V.

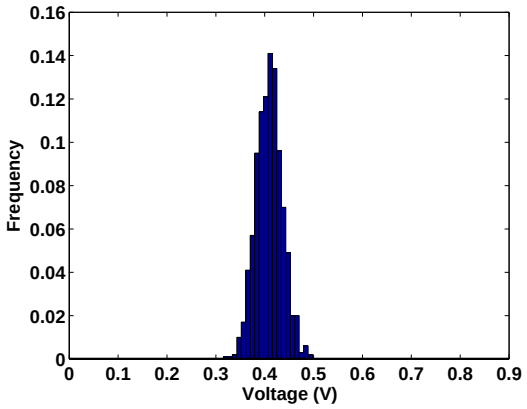


FIGURE 5.5: Process variation induced values of the logic threshold voltage for the example gate input

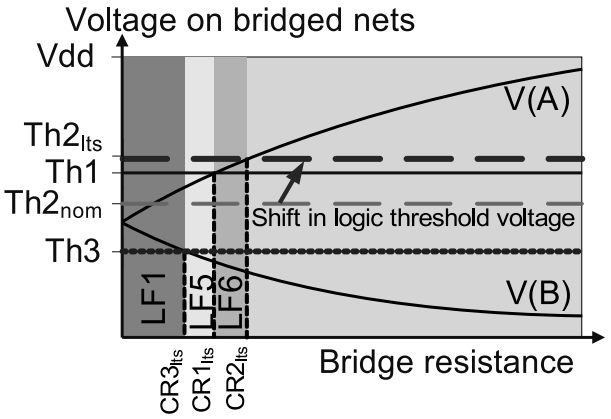


FIGURE 5.6: Shift in logic threshold voltage Th2

Figure 5.6 shows the logic threshold shifted from $Th2_{nom}$ to $Th2_{lts}$ (Logic Threshold Shift lts). $Th2_{lts}$ is higher than Th1, which is the other way around compared to the nominal scenario in Figure 5.2, causing a change in logic behaviour. The outcome is that logic fault LF1 covers the resistances in [0,$CR3_{lts}$] and there are two new logic faults, LF5 (for [$CR3_{lts}$,$CR1_{lts}$]) and LF6 (for [$CR1_{lts}$,$CR2_{lts}$]). This shows that a shift in logic threshold can change the logic behaviour of a bridge.

5.2.2 Test Escapes

Test escapes are undetected logic faults that correspond to bridge resistances that are not covered by the test. Each logic fault has three components: (1) a configuration of logic values on the inputs to the gates that drive the bridged nets, (2) a range of bridge resistance and (3) a configuration of logic values on the gate inputs that are driven by the bridged nets. The combination of the logic values on the driving gates and any bridge resistance in the range will cause the logic configuration on the driven gate inputs.

Possible causes of test escapes are process sensitivities [163], test conditions (tempera-

ture, supply voltage [114]) or inaccurate estimations of defect behaviour [39]. An example of the latter is to assume fixed process parameters in a scenario where process variation causes the parameters to vary and influence the defect behaviour. In this section, test escapes due to process variation are analysed. To put the focus on process variation on IC parameters, all other variables (like temperature) are kept constant.

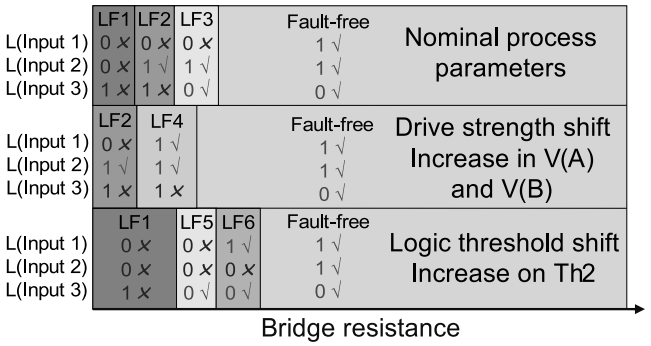


FIGURE 5.7: Logic behaviours for the three configurations of parameter values

Figure 5.7 shows how test escapes occur due to process variation by considering three scenarios: nominal parameters (Figure 5.2), gate drive strength shift (Figure 5.4) and logic threshold voltage shift (Figure 5.6). The shaded boxes are the logic faults and their corresponding defect resistances. Faulty logic values are marked 'x' and fault-free 'v'.

To be able to reason about test escape, a test is required. To see how Figure 5.7 relates to test escape, consider a test that is generated without consideration of process variation. Such test generation would only consider the nominal scenario (top row Figure 5.7) and would produce test pattern TP1 that detects all the three logic faults LF1, LF2 and LF3 by propagating the fault effect Logic-0 through Input 1 (Figure 5.1). By detecting LF1, LF2 and LF3 test pattern TP1 covers all detectable bridge resistance. Therefore it is assumed that test generation without consideration of process variation detects LF1, LF2, LF3 and LF5. (In Figure 5.7, L(Input 1) is a faulty Logic-0 for these logic faults.) The test, called T1 in Section 5.2, that is generated assuming the nominal scenario only includes one test pattern, TP1. That means that in the drive strength shift scenario (middle row Figure 5.7), bridges with logic fault LF4 are test escapes, because a faulty logic value is exposed only on Input 3 (Figure 5.1) and not on Input 1, as required by TP1. Similarly, in the logic threshold shift scenario (bottom row Figure 5.7), bridges with logic fault LF6 are test escapes, not detected by test pattern TP1, because a faulty logic value is exposed only at Input 2 (Figure 5.1). The above examples and reasoning shows that process variation may cause test escapes, i.e. cause the test to fail in detecting some logic faults and corresponding bridge defects.

5.3 Test Robustness

The discussion so far has shown that the logic behaviour of resistive bridges depends on the logic threshold voltage and the drive strength of gates at the bridge location. These two parameters, logic threshold voltage and gate drive strength are influenced by process variation. This means that process variation induces logic faults that were not considered in test generation and thus go undetected. If there are bridge defects with resistance specific to such undetected logic faults, they would not be covered by the test. Such process variation induced test escapes reduce the test quality. This section presents a metric for the impact of process variation on test quality.

5.3.1 Test Robustness Calculation

In this section a metric, called test robustness, is proposed for quantifying the impact of process variation on test quality. The concept of test robustness should not be confused with the concept of robust path delay fault testing, which has to do with preventing glitches from invalidating delay fault testing [164]. The main idea in the test robustness metric is to consider how process variation impacts a set of fabricated units. Each fabricated unit has a fixed configuration of the parameter values whereas studying a set of fabricated units reveals the variation. For example, the particular parameter values in one device cause a particular logic fault, whereas for another device (with other values for the parameters) the logic fault does not occur. For a sufficiently large set of fabricated units, it is possible to use statistical and probabilistic methods to draw conclusions about the impact of process variation.

In the calculation of test robustness, each fabricated device is modelled by a Parameter Value Configuration (PVC), which defines the values of parameters that affect the behaviour of a bridge location, i.e. the drive strengths of the gates that drive the bridged nets and the logic threshold voltages of the gates that are driven by the bridged nets. The test robustness metric finds the impact of process variation on test quality by simulating a large enough set of PVCs while combining the results of each simulation into one value called test robustness.

$$Robustness(b,T) = \frac{\sum_{c \in PP} P(c) \cdot DC(b,c,T))}{\sum_{d \in PP} P(d)} \tag{5.2}$$

$$DC(b,c,T) = \frac{\|CADI(b,c,T)\|}{\|GADI(b,c)\|} \tag{5.3}$$

The test robustness metric is presented in Equation 5.2. The equation gives the robustness of a test T regarding a bridge b for a set of PVCs PP. The PVCs in PP represent

different ICs that are influenced by process variation. The more PVCs that are considered, the better accuracy is achieved in the test robustness calculation. In Equation 5.2, $P(c)$ is the probability for PVC $c \in PP$ and $DC(b, c, T)$ is the defect coverage achieved by bridge b for the same PVC. The definition of $DC(b, c, T)$ is given in Equation 5.3, which is an extension of Equation 5.1 as CADI (the set of covered resistance values) and GADI (the set of detectable resistance values) depend on the PVC c. The resistance ranges for different logic faults are required to calculate CADI and GADI. These resistance ranges can be determined by a series of simulations with Spectre, following the process outlined by the example in Figure 5.2 and Figure 5.7. Another method for determining the resistance ranges that does not rely directly on Spectre is presented in Section 5.3.4. The denominator in Equation 5.2 ($\sum_{d \in PP} P(d)$) adjusts so that full robustness, i.e. full defect coverage for all PVCs, has the value of one.

As can be seen from Equation 5.2, the robustness for a test T on a given bridge b, is determined by the defect coverage and the probability for each PVC in PP. In general, the more PVCs considered, the more accurate the estimate of the robustness (Section 5.3.3). The robustness is a probabilistic metric, combining the outcome of several PVCs. Therefore, it does not give the details of particular PVCs.

PVCs are generated by assigning a random number to each logic threshold voltage parameter and gate drive strength parameter using a random number generator that follows the Gaussian distribution and the mean and standard deviation values for each specific parameter. These mean and standard deviation values are determined as described in Section 5.3.2. The mean μ and standard deviation σ values are also used to estimate the probability $P(c)$ of each PVC c. The probability value $P(c)$ is the product of each probability P_{val} for the parameter values in c. This parameter value probability P_{val} is in turn computed by Equation 5.4, which is the probability density function for Gaussian distribution when $\mu = 0$ and $\sigma = 1$. The probability of the parameter value is taken from the probability density function according to Equation 5.4, where $x = (y - \mu)/\sigma$ and y is the value of the IC parameter.

$$P_{val}(x) = \frac{1}{\sqrt{2 \cdot \pi}} \cdot e^{-\frac{x^2}{2}} \tag{5.4}$$

Using the circuit in Figure 5.1, Table 5.1 shows how the robustness is calculated. There are ten PVCs c0 to c9. The values for the logic thresholds (columns Th1, Th2 and Th3) and for the gate drive strength balance (here represented by V(A,R=0Ω) for simplicity) are taken from Gaussian distributions according to the mean (μ) and standard deviation (σ) given at the bottom of each column. The mean and standard deviation values are assumed for demonstration purposes but could be obtained by Monte-Carlo simulation as was discussed in Section 5.2.1. More detail on the Monte-Carlo simulation is given in Section 5.3.2. Column $P(c)$ gives the probability of the PVCs corresponding to a product of the P_{val} probability for each parameter value. For ex-

TABLE 5.1: Example robustness calculation

	Th1	Th2	Th3	V(A,R=0Ω)	$P(c)$	DC	$P(c) \cdot DC$
c0	0.440	0.420	0.380	0.400	0.0253	1	0.0253
c1	0.440	0.420	0.380	0.425	0.0195	0.40	0.0078
c2	0.440	0.455	0.380	0.400	0.0177	0.75	0.0133
c3	0.430	0.444	0.321	0.422	0.0081	0.77	0.0063
c4	0.488	0.351	0.437	0.456	0.0009	1	0.0009
c5	0.507	0.410	0.431	0.392	0.0048	1	0.0048
c6	0.387	0.463	0.352	0.341	0.0031	0.84	0.0026
c7	0.501	0.475	0.313	0.319	0.0004	1	0.0004
c8	0.369	0.359	0.441	0.469	0.0004	0.94	0.0004
c9	0.394	0.401	0.368	0.368	0.0112	0.92	0.0102
μ	0.440	0.420	0.380	0.400	Sum		Sum
σ	0.048	0.042	0.045	0.048	0.0914		0.0720
						Robustness	0.788

ample, by applying Equation 5.4 concerning PVC c3 in Table 5.1, the following values are found: $P_{val}(Th1 = 0.430)$=0.3904, $P_{val}(Th2 = 0.444)$=0.3388, $P_{val}(Th3 = 0.321)$=0.1689 and $P_{val}(V(A,R = 0\Omega)$=0.422)=0.3591. This means that the product P(c3)=$0.3904 \cdot 0.3388 \cdot 0.1689 \cdot 0.3591 = 0.0081$, which is the value in the $P(c)$ column. I.e. $P(c)$ shows the product of the probabilities for the values shown in the columns Th1, Th2, Th3 and V(A,R=0Ω). In PVC c0, all parameters have the mean value, and this PVC has the highest probability (0.0253). PVC c0 causes the nominal behaviour shown in Figure 5.2. Similarly, PVC c1 is the gate drive strength shift scenario of Figure 5.4. The drop in defect coverage in PVC c1 is due to the escaping logic fault LF4 (Figure 5.7). PVC c2 is the logic threshold shift scenario of Figure 5.6 with the escaping logic fault LF6 (Figure 5.7). DC is the defect coverage $DC(b, c, T1) = \|CADI(b, c, T1)\|/\|GADI(b, c)\|$, for test $T1$ (Section 5.2.2). The DC values shown in Table 5.1 were calculated for each of the PVCs c0-c9 using Equation 5.3. DC=1 means that there is no test escape. DC<1 means that test escapes occur with the probability shown in column $P(c)$. Column $P(c) \cdot DC$ shows the product of the probability of the PVC (from column $P(c)$) and the defect coverage for the PVC (from column DC). Summing column $P(c)$ gives the denominator in Equation 5.2 and summing the column $P(c) \cdot DC$ gives the numerator. The robustness calculated in Table 5.1 is 0.788. The fact that the robustness is <1 means that for some PVCs, there are test escapes that reduces the defect coverage. This happens for PVC c1, c2, c3, c6, c8 and c9. It should be noted that the PVC probabilities have the role of giving appropriate weight to each PVC. Even though there is test escape for c8, it has little impact on the robustness, because of the low probability (0.0004). PVC c1 is more probable (0.0195) and reduces the defect coverage, which affects the robustness. It can be seen in Table 5.1 how the probability $P(c)$ and defect coverage DC for each PVC c influence the test robustness.

Equation 5.2 calculates the robustness for a bridge b. To calculate the test robustness for a design with a set of bridges B, Equation 5.5 is provided, showing the weighted average test robustness ($WA(T)$). Here $w(b)$ is a weight that corresponds to the bridge defect probability for bridge b, defined so that $b \in B$ and $\sum_{b \in B} w(b) = 1$. The bridge defect probabilities should be considered as shown in Equation 5.5 to give appropriate weight to bridge locations that are more probable to have a defect than others. If the bridge defect probability data is not available, then for each bridge b, $w(b) = \frac{1}{\|B\|}$ which means that the overall robustness is the average over the bridge locations.

$$WA(T) = \sum_{b \in B} Robustness(b, T) \cdot w(b) \tag{5.5}$$

5.3.2 Preparation of Test Robustness Calculation

In preparation for calculating the test robustness, the distribution of logic threshold voltage values and gate drive strength values should be computed. This is done by performing Monte-Carlo simulation, while varying IC parameters with μ and σ values according to Table 5.2. Table 5.2 is based on variation data for relevant process parameters based on [32,23,21] and 45nm technology transistor models from [160]. To account for voltage drop in Vdd in practise, Vdd is varied by 2.5% (0.022V) around 0.878V for a 0.9V nominal Vdd. 5nm standard deviation is assumed for the transistor length (L) and width (W) due to line edge roughness affecting the effective geometry of fabricated transistors [23,22]. For the thickness of the gate dielectric/oxide, 1.5Å standard deviation reflects the thickness of one atom layer [22]. For the transistor threshold voltage, 0.045V standard deviation was chosen for random dopant fluctuations based on observations reported in [21] for a minimal size transistor. For these parameters, a Gaussian distribution is assumed. It should be noted that in practise, the distributions of these parameters are limited, because otherwise, there would be unreasonable parameter values such as a negative transistor length. The distributions used in the experiments in this chapter use the Gaussian distribution truncated to three standard deviations.

From the Monte-Carlo simulation described above, calculation of the mean (μ(Th)) and standard deviation (σ(Th)) values was performed for each considered logic threshold voltage and gate drive strength parameter. These μ and σ values were saved for processing in the test robustness calculation (Equation 5.2). For the logic threshold voltages, it was found that there was 11% to 15% standard deviation (σ(Th)).

In this work, the gate drive strength g (gate output conductance) is represented by its reciprocal, the resistance R_H (R_L) between the Vdd (ground) node and the gate output for the gate that is driving high (low). This resistance depends on current I_{ds} through the transistors in the gate according to Equation 5.6 and the voltage V_{ds} between the corresponding source and drain nodes.

TABLE 5.2: Varied process parameters

	μ	σ
Vdd	0.878V	0.022V
W	from gate lib.	5nm
L	45nm	5nm
TOXN	17.5Å	1.5Å
TOXP	18.5Å	1.5Å
VTHN	0.471V	0.045V
VTHP	-0.423V	0.045V

$$I_{ds} = \mu_n \cdot C_{ox} \cdot \frac{W}{L} \cdot \left((V_{GS} - V_{th}) \cdot V_{ds} - \frac{V_{ds}^2}{2} \right) \tag{5.6}$$

Equation 5.6 describes the current I_{ds} through a transistor operating in the linear region ($V_{ds} < V_{gs} - V_{th}$). This is used to show how the resistance of a gate driving high, R_H, is calculated for an inverter. V_{ds} is the voltage between the power rail and the gate output (this applies to an inverter). For an inverter driving high, $R_H = \frac{V_{ds}}{I_{ds}}$. From this it can be seen that R_H depends on V_{ds}, the voltage between the power rail and the gate output. The V_{ds} in turn depends on the resistance R of the bridge defect and the resistance R_L of the gate that is driving low, according to $V_{ds} = Vdd \cdot \frac{R_H}{R_H+R+R_L}$. Thus, R_H (and R_L) can only be measured in the context of the bridge location. Because of this, the Monte-Carlo simulation described above is performed for each pair of gates in the gate library, for each input assignment to them and for a range of bridge resistance values.

The mean $\mu(R_H)$ ($\mu(R_L)$) and standard deviation $\sigma(R_H)$ ($\sigma(R_L)$) for the resistance that represents the gate drive strength are calculated from the outcome of the Monte-Carlo simulation and saved for processing in the test robustness calculation. It was found that a typical R_H or R_L value lies in the range 1kΩ to 10kΩ and the standard deviation for R_H and R_L is in the range 29% to 38% when the bridge resistance R=0Ω, depending on the type of gates involved in driving the bridged nets. Furthermore, it was found that the standard deviation for R_H and R_L decreases for increasing values of bridge resistance, as can be seen in the effect on the voltage on the bridged nets in Figure 5.3.

Once the means and standard deviations for the logic threshold voltages and gate drive strengths have been obtained, these mean values and standard deviations are used in the further processing to calculate the test robustness with Equation 5.2.

5.3.3 Considerations for Accuracy in the Test Robustness Metric

The accuracy of the test robustness metric (Equation 5.2) depends on the number of parameter value configurations (PVCs) considered in its calculation. In general it can be said that more PVCs mean a higher accuracy. The level of accuracy can be seen by the number of unique logic faults that are encountered while simulating the set of PVCs. To determine an appropriate number of PVCs to consider in the test robustness metric, a series of experiments were performed. In each experiment, a different number of randomly generated PVCs were simulated (for more detail about how PVCs are generated, see Section 5.3.2). Four observations were made. Firstly, it was seen that the number of encountered logic faults follows the law of diminishing returns, i.e. the majority of logic faults were encountered already using the first few PVCs and to find further logic faults, many further PVCs were required. Secondly, it was seen that the number of encountered logic faults increases step-wise, i.e. after the first few PVCs have been simulated, it often occurs that many (>100) randomly generated PVCs are simulated without finding more logic faults. Thirdly, after simulating a large set of PVCs, the set of encountered logic faults stops growing altogether because there are no more possible logic faults to find. In other words, the set of encountered logic faults saturate. Lastly, a correlation was found between the number of required PVCs and the number of fan-ins and fan-outs concerning the bridge location. All of the bridges that required >20 PVCs to saturate the set of logic faults had either a combined fan-in >2 or a combined fan-out >4. The combined fan-in (fan-out) is the sum of the fan-in (fan-out) for the gates that drive the bridged nets. The typical relation between the number of simulated PVCs and the number of identified logic faults is illustrated in Figure 5.8.

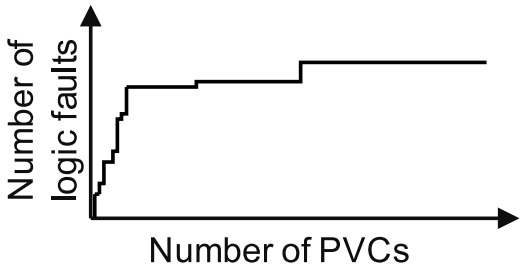


FIGURE 5.8: Typical scenario when identifying variation induced faults by simulating PVCs

In deciding the number of PVCs to consider in the experiments, it was taken into account that many PVCs also mean a long computation time, as every PVC corresponds to simulation of the bridge fault site (Appendix A). For the experiments presented in this work, a single-pattern, single-fault implementation of a bridge resistance-aware fault simulator (it is presented in Section 3.2.3) was used to calculate the $CADI$ and $GADI$ values used for the defect coverage (Equation 5.3). With a simulator that employs parallelism for speed-up [108] it is expected that more PVCs could be considered within the same CPU time. With the computation time in mind, it was decided that the

analysis should be performed with 500 PVCs. To validate the decision of using 500 PVCs, a few experiments were performed with larger sets of PVCs and it was found that the test robustness results did not change significantly due to this decision. This can be explained by noting that the majority of process variation induced logic faults are identified with the first few PVCs. Because of the fact that the PVCs are randomly generated, these first logic faults are among the ones that have the highest likelihood of occurring.

5.3.4 Estimating the Resistance Range for a Logic Fault Under Process Variation

The test robustness metric (Equation 5.2) requires the defect coverage to be calculated for a given PVC (Equation 5.3). This involves determining the resistance range for each logic fault based on the parameter values that are specified in the PVC. The resistance ranges can be determined by simulation using Cadence Spectre, to get the voltages for the bridged nets and compare them with the logic threshold voltages as in the example in Figure 5.2 and Figure 5.7. However, Spectre simulations are time consuming and that may make it infeasible to process many PVCs. Instead, the approach employed for the experimental analysis performed in this chapter is based on a database of pre-calculated functions for the drive strength $g_H = 1/R_H(R_{Sh})$ and $g_L = 1/R_L(R_{Sh})$ of the gate that drives high and the gate that drives low. Each database entry is given as a function of the bridge resistance R_{Sh}. The database describes the nominal scenario and the functions for the gate drive strengths are modified according to the PVCs. The PVC c leads to the modified gate drive strength functions $g_H(c) = 1/R_{H,c}(R_{Sh})$ and $g_L(c) = 1/R_{L,c}(R_{Sh})$. These modified functions represent the effect of process variation on the gates that drive the bridged nets. The resulting voltage on the bridged nets is modelled by Equation 5.7 and Equation 5.8 for the net that is driven high and the net that is driven low respectively.

$$V_H(R_{Sh}, c) = Vdd - Vdd \cdot \frac{R_{H,c}(R_{Sh})}{R_{H,c}(R_{Sh}) + R_{L,c}(R_{Sh}) + R_{Sh}} \tag{5.7}$$

$$V_L(R_{Sh}, c) = Vdd \cdot \frac{R_{L,c}(R_{Sh})}{R_{H,c}(R_{Sh}) + R_{L,c}(R_{Sh}) + R_{Sh}} \tag{5.8}$$

The logic threshold voltage $Th(c)$ for each input that is driven by the bridged nets are modified in a similar way according to the PVC c. The V_H and V_L functions are compared with the corresponding Th values to find the resistance ranges and logic behaviours, as is illustrated by the example of Figure 5.2 and Figure 5.7.

5.4 Process Variation Aware Test Generation

The proposed process variation-aware test generation method is called PVAA (Process Variation-Aware ATPG) and is presented in Figure 5.9 and Figure 5.10. The key observation for process variation-aware test generation is that a test escape that has a high probability of occurrence and corresponds to a large amount of undetected bridge resistance has a larger impact on test robustness than a test escape with low probability or a small amount of undetected bridge resistance. This observation is supported by experimental results in Section 5.5.2. Thus, it is preferable to target the logic faults that have high probability of occurrence and correspond to a large amount of otherwise undetected bridge resistance. This observation is exploited in the method by two features: (1) logging of the probability and bridge resistance range for each process variation induced logic fault and (2) selection of logic faults to be targeted by test generation based on the logged values.

The inputs to the PVAA method (Figure 5.9) are: a design C with a set of bridge locations B (each with a bridge defect probability $w(b)$ where $b \in B$) and probability distribution data for the logic threshold voltages and gate drive strengths (as discussed in Section 5.3.2). An optional test set T can be given to the test generator, which will make it add test patterns to T until T has acquired the required weighted average test robustness, WA_{target}. If no test set is supplied, the test generator will generate an entirely new test set to meet the same target. The bridge locations are identified using the bridge location list generation tool presented in Section 3.2.1 which has been extended to include the capability of determining the bridge defect probabilities using the coupling capacitance values that are extracted from the circuit layout using Cadence Encounter. A set of parameter value configurations (PVCs) called PP is generated corresponding to the probability distribution data as discussed in Section 5.3.1. Each PVC $c \in PP$ has a probability $P(c)$.

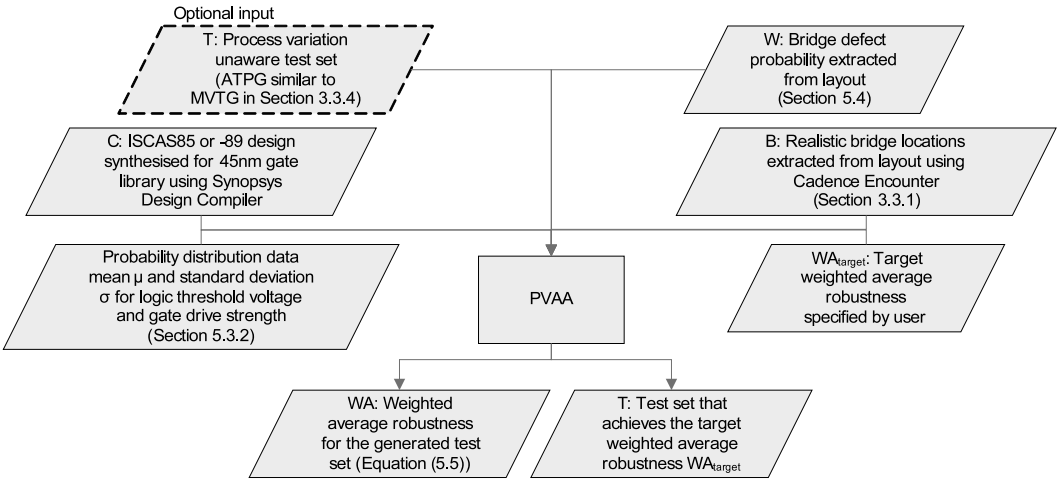


FIGURE 5.9: PVAA top level flow

The method PVAA (Figure 5.10) has two steps, corresponding to the above mentioned features: (1) identification of process variation induced logic faults F (the fault domain) and the corresponding resistance intervals RR for a set of PVCs (parameter value configurations) PP, and (2) iterative selection of test patterns for the faults in F until a user-specified weighted average test robustness (Equation 5.5) target WA_{target} is achieved. The method contains an algorithm RRC (Robustness ReCalculation) in step 3C, which is shown in Figure 5.11 and is used to recalculate the achieved weighted average test robustness whenever a new test pattern is added to the test set.

The PVAA method starts with step 1A, by identifying process variation induced logic faults for each bridge b and PVC c. The process variation induced logic faults are identified in step 1A by applying the parametric bridge fault model for each PVC. Employing the parametric bridging fault model provides estimates of the resistance ranges for each logic fault in Step 1B. The resistance intervals are at this stage of the method called *remaining resistance* ($RR(b, f, c)$) because these are resistances that are not yet covered by test set T, as no test patterns have been added to T yet. The logic faults that were identified in step 1A may be undetectable. To determine if the logic faults are detectable or not, test pattern generation is performed on each logic fault in step 1C using a solver for the Boolean Satisfiability problem [44]. For more details on how test pattern generation can be performed using this type of solver, see Appendix B. The test pattern generation succeeds only if the logic fault is detectable and only detectable logic faults are considered in the subsequent processing. The complete set of detectable logic faults is called "the fault domain". From this point on, each considered logic fault f is associated with a test pattern tp that detects f. Some of the test patterns generated in step 1C are later included in the test set T. In step 1D, the resistance intervals for the detectable logic faults are logged in $GADI(b, c)$, the set of detectable bridge resistances, which forms the basis for the defect coverage calculation using Equation 5.3.

As an optional step 2A it is possible to fault simulate (Section 1.4.3 and Section 3.2.3) an existing test set T for each PVC and bridge and so determine the logic faults and bridge resistance ranges that are already detected and covered. If this option was employed, step 2B would calculate the weighted average test robustness achieved by the test, to evaluate if further test generation is required.

Once the process variation induced logic faults have been identified as detailed above, the method enters a loop of steps 3A, 3B and 3C, which continues until the user-specified weighted average robustness (WA_{target}) is achieved. In step 3B, the method selects a logic fault f that has the highest *incremental robustness contribution* (IRC, Equation 5.9) among all the logic faults for all the bridges. The incremental robustness contribution is the value that would increment the weighted average robustness if a test pattern tp that detects the logic fault f was added to the test set. This means that the highest incremental robustness contribution marks the logic fault that is most probable to cover the most of previously undetected defect resistance. The incremental robustness

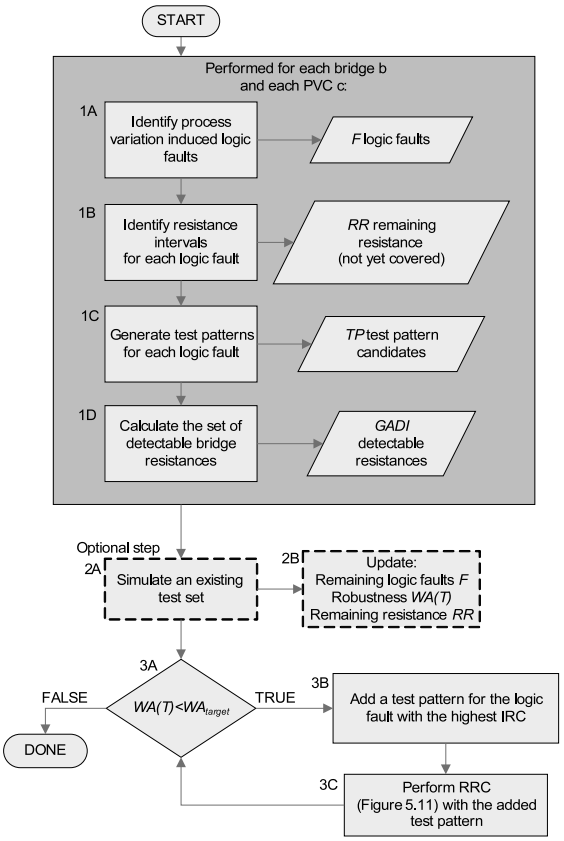


FIGURE 5.10: The flow of the process variation-aware test generation method (PVAA)

contribution (IRC) is defined in Equation 5.9.

$$IRC(f,b) = \sum_{c \in PP} \frac{w(b) \cdot P(c) \cdot \frac{\|RR(b,f,c)\|}{\|GADI(b,c)\|}}{\sum_{d \in PP} P(d)} \tag{5.9}$$

The logic fault f that is selected in step 3B is associated with a test pattern tp which was generated in step 1C while confirming that f is detectable. This test pattern tp is added to the final test set.

In method PVAA, step 3B requires that the incremental robustness contribution, IRC, is kept up-to-date. Each time step 3B is performed and a test pattern is added to the test set, this test pattern will cover some of the remaining bridge resistances ($RR(b,f,c)$). These bridge resistances must be removed from RR and IRC must be recomputed before step 3B is performed next time. To update the set of remaining bridge resistances RR for the selected test pattern tp, algorithm RRC (Robustness ReCalculation, Figure 5.11) is performed in step 3C. Furthermore, algorithm RRC updates the weighted average robustness $WA(T)$. If $WA(T) > WA_{target}$ (the user specified target), then the PVAA method (Figure 5.10) terminates, otherwise it will continue to increase the weighted average test robustness by adding another test pattern as discussed above.

FIGURE 5.11: Algorithm RRC - Robustness ReCalculation

Input:

Test pattern tp just added to the test set
Fault domain F
Bridge locations B
Bridge weights $w(b)$ for $b \in B$
Set of PVCs PP
each with probability $P(c)$, $c \in PP$
Remaining (not yet covered) resistance ranges
$RR(b, f, c)$ for all
$b \in B$, $f \in F$ and $c \in PP$
Set of detectable defect resistance $GADI(b, c)$
for $b \in B$ and $c \in PP$
Weighted average test robustness $WA(T)$

Output:

Updated $WA(T)$
Updated $RR(b, f, c)$ for all
$b \in B$, $f \in F$ and $c \in PP$

1: **for all** faults $f \in F$ detected by tp **do**
2: *// b is the bridge location of fault f*
3: **for all** PVCs $c \in PP$ **do**
4: *// update WA(T) one PVC at a time*
5: $WA(T) := WA(T) + \frac{w(b) \cdot P(c) \cdot \frac{\|RR(b,f,c)\|}{\|GADI(b,c)\|}}{\sum_{d \in PP} P(d)}$
6: *// mark the resistance as covered*
7: **for all** faults g for bridge b **do**
8: $RR(b, g, c) := RR(b, g, c) \backslash RR(b, f, c)$
9: **end for**
10: **end for**
11: **end for**

Algorithm RRC (Figure 5.11) shows how the weighted average test robustness $WA(T)$ is incrementally calculated for each newly selected test pattern tp (from step 3B of PVAA, Figure 5.10). Algorithm RRC keeps the remaining resistance ranges RR up-to-date as test patterns are added to the test set, because keeping RR up-to-date is required for incremental robustness calculation (IRC, Equation 5.9). The algorithm operates as follows: For each fault f that has been detected by the new test pattern tp (line 1), $WA(T)$ is increased (line 5) with the incremental robustness contribution, IRC (Equation 5.9), for one PVC at the time (line 3). Each PVC $c \in PP$ is considered in Algorithm RRC to account for the fact that the detected logic fault f may occur for several different PVCs. Test pattern tp covers some of the remaining resistance ranges RR, which are then removed from the set of remaining resistances RR as shown on lines 7-9.

To summarise, the process variation-aware test generation method is able to produce

test sets that achieve a user-defined weighted average test robustness target. This way it is possible to trade off test set size (which is an important factor in the total cost of test) to test quality in the presence of process variation.

5.5 Experimental Results

The experiments presented in this section were performed on ISCAS85 and ISCAS89 benchmark circuits, synthesised with Synopsys Design Compiler to a 45nm gate library [15] using transistor models from [160]. Cadence Encounter was used to generate layouts for the designs and possible bridge locations were identified in the layouts. For the weighted average test robustness calculations, bridge weights w were computed using the coupling capacitance between the bridged nets. A bridge location with a higher coupling capacitance has higher defect probability than a bridge location with a lesser coupling capacitance. The test sets used as input in the experiments were generated using a single-Vdd version of the test generator presented in Chapter 3. This test generator is not process variation-aware and assumes nominal values for all process parameters. These test sets are used in the experiments to show the impact of process variation on tests that are generated without consideration of process variation. The ATPG engine used in the test generation method is a solver [44] for the Boolean Satisfiability problem (Section 1.4.4 and Appendix B).

The results of three experiments are discussed next to analyse different aspects of testing for resistive bridging faults in the presence of process variation. Firstly, it is shown how the proposed test robustness metric reflects test quality (Section 5.5.1). Secondly, (Section 5.5.2) the quantitative impact of process variation on test quality is measured. Finally (Section 5.5.3) experimental results for the proposed PVAA method are presented.

5.5.1 Analysis: Test Robustness Reflects Test Quality

A test set for benchmark S838 which has 28 bridge locations, that has 10 test patterns that achieve full defect coverage when there is no process variation, was simulated using 500 PVCs and the test robustness (Equation 5.2) was calculated to analyse how process variation impacts test quality and to explain how such impact is measured by the test robustness metric (Equation 5.2). Figure 5.12 shows the ratio of detected process variation induced logic faults, i.e. the logic fault coverage. Each bar corresponds to a bridge location as numbered below the graph. The height of the dark bar shows how many percent of the fault domain of process variation induced logic faults that are detected by the test set. The bridges have been enumerated according to the height of the dark bar. Above each bar is the total number of process variation induced logic faults in the fault domain for the bridge location.

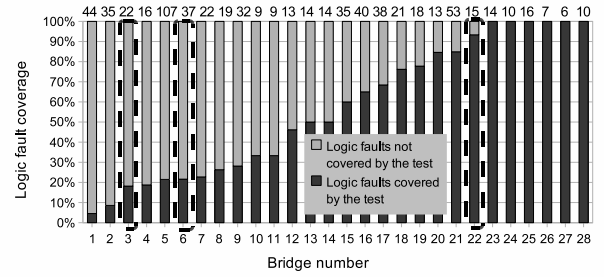


FIGURE 5.12: Detected and undetected logic faults on benchmark circuit S838

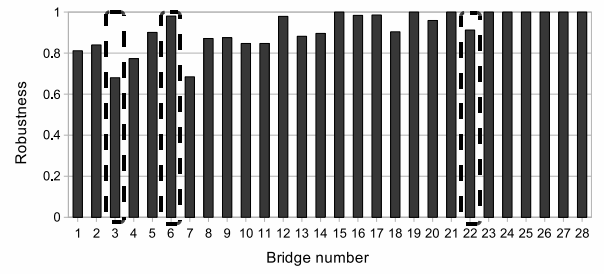


FIGURE 5.13: Robustness for the bridges of benchmark circuit S838

Figure 5.12 shows that some bridge locations are sensitive to process variation in terms of the logic behaviour. In particular bridges 1 to 9 have many logic faults that did not occur for the nominal process parameter values, as can be seen by the low dark bars. Figure 5.13 shows the test robustness for the bridges. It should be noted that the test robustness does not necessarily follow the logic fault coverage (compare Figure 5.12 and Figure 5.13). To explain the test robustness values, three bridge locations (3, 6 and 22) are analysed in more detail (marked in Figure 5.12 and Figure 5.13). Bridge 3 has low fault coverage and low test robustness with the test set, whilst bridge 6 has also low fault coverage but high test robustness. Bridge 22 has higher logic fault coverage than bridge 6 but still lower test robustness.

To explain why the three bridge locations have different statistics in terms of fault coverage and test robustness, consider Figure 5.14, which shows, for the studied PVCs, the probability of the PVC (vertical axis) and the amount of undetected defect resistance for the PVC (horizontal axis). Each dot in the graph corresponds to a test escape. The graph on the top left is an example that illustrates that test escape with both high probability and a large amount of undetected defect resistance has larger negative impact on test robustness than a test escape with a low probability (close to the horizontal axis) or a small amount of undetected defect resistance (close to the vertical axis). The impact of a test escape grows in the direction of the arrow, which means that dots in Figure 5.14 that are close to the axes have small impact and dots that are in the middle of the graphs, away from the axes, have large impact.

As can be seen from Figure 5.14, Bridge 3 with low fault coverage has many test escapes (dots), most of which are in the middle of the graph and thus have high probability of

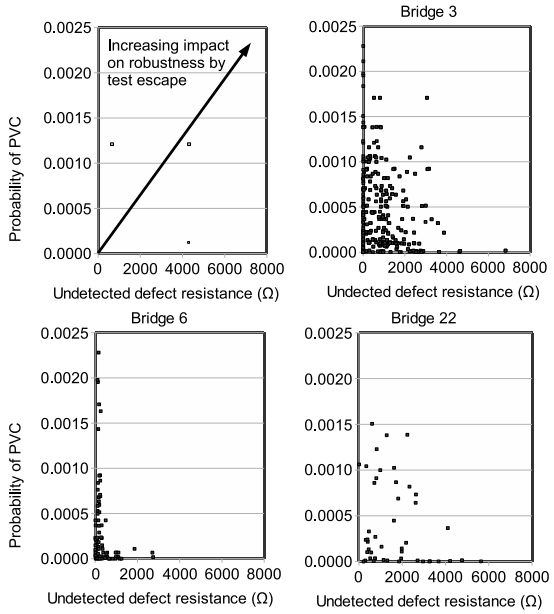


FIGURE 5.14: Test escapes of bridge 3, 6 and 22 of benchmark S838

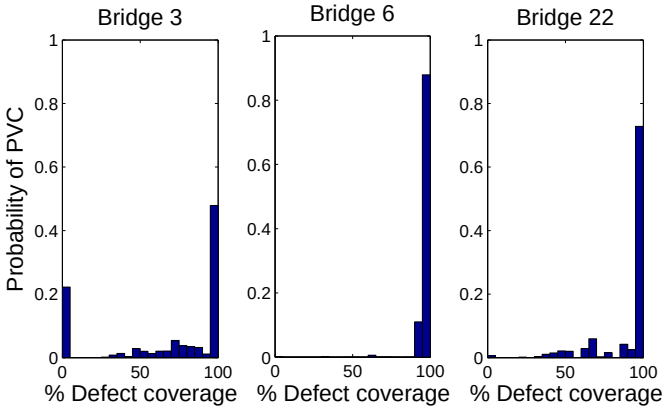


FIGURE 5.15: Probability of defect coverage for bridge 3, 6 and 22 of design S838

leaving large amounts of defect resistance undetected. Therefore, Bridge 3 has low test robustness with the considered test set. On the other hand, Bridge 6 has a similarly low fault coverage but a higher test robustness. The difference is explained by Figure 5.14 as all the dots for Bridge 6 are close to the axes, which means that the test escapes have low impact on test robustness. Compare with Bridge 22, which has a high logic fault coverage but a lower test robustness than Bridge 6. This is explained by the fact that even though there are few test escapes (dots) for Bridge 22 (Figure 5.14), most of them are in the middle of the graph and have high impact on test robustness.

A further illustration of how the considered test set perform for Bridge 3, 6 and 22 is given in Figure 5.15. The horizontal axis shows the defect coverage and is divided into bins, where each bin corresponds to 5% of the range of possible defect coverage values. The height of the bars for each bin indicate the probability that a fabricated unit will have the corresponding defect coverage with the considered test set. For example Bridge 3 has

TABLE 5.3: Test robustness for benchmark circuits and corresponding process variation-unaware tests

Design	Gates	Bridges	TPs	Nominal		With process variation		
				Domain	DFs	Domain	DFs	$WA(T)$
C432	175	36	31	363	118	1719	837	0.916
C499	211	107	37	1560	372	9990	5951	0.958
C880	297	96	45	1981	464	10654	4850	0.978
C1355	307	111	46	2912	668	24008	7993	0.965
C1908	278	154	55	2042	677	10691	6151	0.971
C2670	481	154	68	2434	642	9499	5595	0.943
C3540	1001	695	138	9317	3206	50586	42917	0.992
S641	175	44	20	649	165	4976	2613	0.938
S838	265	28	11	343	105	1357	612	0.899
S1488	704	873	124	9613	3721	53925	48863	0.991
S5378	1365	727	168	10024	2971	50619	39126	0.984
S9234	1015	318	88	4593	1292	28354	19334	0.956

≈0.5 probability of very high defect coverage (95% to 100%). Furthermore, achieving no defect coverage at all for Bridge 3 has ≈0.22 probability. This observation implies that the robustness should be low for Bridge 3 with the considered test set, which is also the case in Figure 5.13. Bridge 6 on the other hand has high probability (≈0.87) of full defect coverage, which explains high test robustness for Bridge 6 as shown in Figure 5.13. Furthermore, Bridge 22 has ≈0.7 probability of full defect coverage with the test set. Thus, the test robustness for Bridge 22 is higher than for Bridge 3 but lower than for Bridge 6. This shows that fault coverage does not necessarily reflect test quality.

The analysis shows that process variation has impact on the test quality and that the test robustness metric reflects it. Furthermore, Bridge 6 shows high test robustness with relatively low logic fault coverage, because all the undetected logic faults have either low probability of occurrence or correspond to small amount of undetected defect resistance. Analysed from a different perspective, this test detects the logic faults that are most probable and corresponds to significant amount of defect resistance and that is why it has high test robustness.

5.5.2 Analysis: Process Variation Impact on Test Quality

To demonstrate and quantify the impact of process variation on test quality, the following experiment was performed. The test robustness was calculated for test sets of 12 ISCAS85 and ISCAS89 benchmark circuits in two scenarios: (1) Nominal and (2) with process variation. In the nominal scenario, test robustness (Equation 5.2) is the same thing as defect coverage (Equation 5.3). The experiment results are given in Table 5.3.

The first column shows the name of the benchmark design. The second and third column show the number of gates and bridge locations in each design. The fourth column, marked "TPs", show the size of the test set. The test sets have full defect coverage in the nominal case which means that any loss of test quality is due to process variation. The next two columns give the outcome of calculating the test robustness for the nominal scenario. The fifth column shows the fault domain (the number of detectable logic faults identified in the experiment) and the sixth column shows the number of logic faults that were detected by the test set to achieve 100% defect coverage. The number of detected faults (DFs) is less than the size of the fault domain in Table 5.3, because a bridge defect can cause several distinctive logic faults and only one out of these logic faults is required to be detected to cover the defect. That is why a test set only targets a subset of the fault domain while achieving 100% defect coverage. The last three columns of Table 5.3 consider the cases when the faulty logic behaviour changes due to process variation. Similarly to the analysis in Section 5.2.1 where the examples of a logic threshold shift and a shift in gate drive strength introduced three new logic faults (LF4, LF5 and LF6), the fault domain of all the benchmarks has increased due to process variation induced faults. Some of these faults are detected by original test sets, just as LF5 in the example in Section 5.2.2, as shown by the DFs column. The number of detected faults has increased as some logic faults are detected by the original test sets. This is called accidental detection, i.e. test patterns that were generated targeting a particular set of logic faults are found to be effective in detecting other logic faults as well. The remaining faults cannot be detected by the original test sets, just as LF4 and LF6 in the example in Section 5.2.2, which means that if some bridge defects cause malfunction only as undetected logic faults, these defects are test escapes, leading to lower weighted average robustness. The process variation induced test escape occurs for all the benchmarks, as demonstrated in the last column. For example, the test set for design C432 has 31 test patterns that detect 118 logic faults in the nominal scenario. These particular 118 logic faults covers all the detectable defect resistance (full defect coverage) in the nominal scenario. There are 363 detectable logic faults in total, but detecting any of the remaining 245 would not lead to increased defect coverage. When process variation is considered by simulating 500 PVCs, there are in total 1719 detectable logic faults and 837 of them are detected by the test set, but some of the remaining 882 logic faults are test escapes, which explains why the weighted average test robustness is <1. From Table 5.3, it can be seen that process variation has a negative impact on test quality, because the weighted average test robustness is <1, indicating that there are escapes for the tests for all benchmarks.

5.5.3 Test Quality Improvement from Variation-Aware Test Generation

To mitigate the test escapes and reduced test robustness, it is possible to employ the proposed process variation-aware test generation method (PVAA, Section 5.4). Two experiments were performed. In the first experiment (Table 5.4) the original test set of each circuit was used. These test sets were augmented with additional test patterns to reduce test escapes for different weighted average test robustness targets (WA_{target} 0.95 to 0.9999). The column DFs shows the number of detected faults and the column TPs represents the total number of test patterns (original test set plus added test pattern).

As can be seen from Table 5.4, it is possible to achieve higher WA_{target} in comparison to Table 5.3 by including additional test patterns. For example, PVAA for design S838 has increased $WA(T)$ to 0.9999 compared with 0.899 (Table 5.3) but at the expense of 36 additional test patterns. The number of additional test patterns differs according to the required test robustness, but in general significant increase in test patterns is required to achieve the WA_{target}=0.9999 when compared with WA_{target}=0.95, in particular for the larger benchmarks. The method targets the logic fault with the highest incremental robustness contribution, as discussed in Section 5.4, which means that the test robustness increases rapidly with the first few test patterns and then increases more slowly as test patterns are added, because the remaining logic faults have less incremental impact on test robustness. Typically, each added test pattern increases the weighted average robustness by a lesser amount than the previously added test pattern. In other words, the cost of adding to the weighted average test robustness increases in terms of test patterns. It should be noted that the number of detected faults (columns marked DFs) increase because of the increase in the number of test patterns. It should be noted in Table 5.4, that test patterns were added only when the test robustness of the original test set was below the target. No test patterns were removed from the original test set. This is the reason why for example C1908 has 55 test patterns for WA_{target} 0.95, 0.96 and 0.97. These 55 test patterns are from the original test set, which achieved 0.971 test robustness, as can be seen in Table 5.3.

In the second experiment, the starting point was an empty test set for each circuit and used the proposed process variation-aware test method PVAA to generate the required test set for a given test robustness target. The results are shown in Table 5.5. As can be seen, the target test robustness for a given design can be met using smaller test set (TPs) when compared with the test set shown in Table 5.4. For example, design S9234 needs 48 test patterns to achieve test robustness of 0.95 compared to 88 test patterns (Table 5.4).

TABLE 5.4: Results for the original test set augmented with test patterns generated by PVAA

Target	$WA_{target} = 0.95$		$WA_{target} = 0.96$		$WA_{target} = 0.97$		$WA_{target} = 0.98$		$WA_{target} = 0.99$		$WA_{target} = 0.995$		$WA_{target} = 0.9999$	
Design	DFs	TPs	DFs	TPs	DFs	TPs	DFs	TPs	DFs	TPs	DFs	TPs	DFs	TPs
C432	860	32	867	33	902	36	978	39	1069	46	1143	53	1357	76
C499	5951	37	5971	38	5971	38	6009	40	6323	45	6675	54	8267	113
C880	4850	45	4850	45	4850	45	4960	47	5318	58	5801	73	7571	163
C1355	7993	46	7993	46	8023	47	8361	52	9413	63	10432	74	15569	153
C1908	6151	55	6151	55	6151	55	6268	59	6577	67	6899	76	8763	181
C2670	5625	71	5637	74	5740	80	6118	92	6498	116	6859	136	7775	251
C3540	42917	138	42917	138	42917	138	42917	138	42917	138	43588	156	47287	301
S641	2731	21	2750	22	2801	24	2892	27	3094	36	3218	43	3694	78
S838	672	13	695	14	718	16	736	19	754	23	837	28	958	47
S1488	48863	124	48863	124	48863	124	48863	124	48863	124	49186	129	51444	226
S5378	39126	168	39126	168	39126	168	39126	168	40431	187	42452	228	43609	602
S9234	19334	88	19406	91	19456	94	19824	101	20862	121	21776	143	24552	295

TABLE 5.5: Results achieved by the process variation-aware test generation method

Target	$WA_{target} = 0.95$		$WA_{target} = 0.96$		$WA_{target} = 0.97$		$WA_{target} = 0.98$		$WA_{target} = 0.99$		$WA_{target} = 0.995$		$WA_{target} = 0.9999$	
Design	DFs	TPs	DFs	TPs	DFs	TPs	DFs	TPs	DFs	TPs	DFs	TPs	DFs	TPs
C432	741	25	807	27	809	28	907	33	970	41	1035	49	1353	75
C499	4581	23	4802	26	5001	30	5316	37	5659	46	6180	53	8190	116
C880	3434	39	3584	43	4133	48	4503	59	5208	81	5938	97	7583	180
C1355	6045	29	6634	34	7012	37	7967	44	8963	58	10293	71	15518	144
C1908	4741	36	5306	42	5513	49	6042	58	6461	68	6835	78	8781	187
C2670	4677	57	4872	62	5263	73	5689	85	6224	112	6629	134	7708	258
C3540	31649	53	33064	62	34925	76	37058	89	40063	117	42568	147	47309	307
S641	2528	25	2568	27	2636	30	2789	36	3067	43	3230	50	3730	85
S838	644	12	657	13	695	15	730	18	785	22	809	27	958	47
S1488	39332	49	40800	56	42418	62	44321	76	46880	97	48259	119	51364	243
S5378	31468	70	32875	82	33273	85	35286	103	38377	145	40401	191	45509	458
S9234	15497	48	16374	56	17562	67	18742	84	20416	113	21327	131	24405	296

In the second experiment PVAA achieves the same robustness targets as in the first experiment with less test patterns because PVAA targets logic faults that (in being detected) has the largest contribution to robustness. Since the original test set used as a starting point in the first experiment (Table 5.4) is not generated with robustness as the guiding metric, it is likely that the original test patterns are less effective in achieving high robustness, leading to a test set that has a higher number of test patterns in Table 5.4. This is true for a large majority of cases with a few exceptions (for example C499, WA_{target}=0.99 and C880, WA_{target} 0.97-0.9999), which are due to accidental detection. Accidental detection means that a test pattern is generated for a particular logic fault, but is found to be effective at detecting other logic faults as well. It should be noted that the number of detected faults (columns marked DFs) increase with the number of test patterns just as for Table 5.4.

To give an insight into how the proposed process variation-aware test generation method PVAA improves test quality, compare Table 5.3 with Table 5.5. For all designs but C880, PVAA achieves higher weighted average test robustness than the original test sets with a smaller test set size. For example, the original test for S9234 achieved $WA(T)$=0.956 with 88 test patterns and PVAA achieved WA_{target}=0.96 with 56 test patterns (improvement of 36%). In the case of C880, the test set size is increased by PVAA, but higher $WA(T)$ is still achieved.

A consistent trend can be seen in Table 5.5, to achieve a higher robustness, more test patterns are required, which cover more of the logic faults. It should be noted in Table 5.5 that typically, there are more test patterns required in order to go from 0.98 robustness to 0.99 robustness than to go from 0.95 robustness to 0.98 robustness. This trend continues as the number of test patterns increases rapidly for robustness targets above 0.99. Indeed, plotting the number of test patterns versus the weighted average test robustness target shows exponential behaviour as is shown by the example of design C499, C1355, S641 and S838 in Figure 5.16.

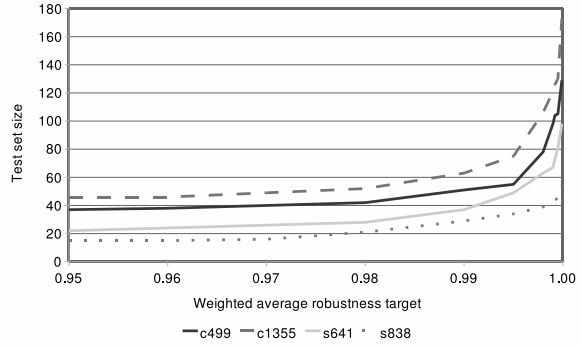


FIGURE 5.16: The number of test patterns required to achieve a given weighted average robustness

5.6 Concluding Remarks

This chapter presented an investigation into the impact of process variation on test quality in the context of logic testing for resistive bridge defects. Resistive bridge defects constitute an important type of defects for deep submicron designs. Process variation, which makes IC parameters deviate from their intended values, is particularly important to consider in such designs because the small transistors in deep submicron designs are sensitive to process variation. Resistive bridge defects connect nets that should not be connected, with a fixed but unknown resistance. In this context, the only previous work [99] presented a process variation-aware fault model that was abstract from the IC parameters by considering the set of possible logic faults for bridge defects. The reason why that study used abstraction was to enable fast fault simulation. The level of abstraction used in [99] made the proposed fault model inappropriate for a detailed study on how process variation impacts testing. As a contrast, this chapter considered in detail how variation of a range of IC parameters influences the behaviour of resistive bridge defects. The considered IC parameters are the transistor threshold voltage VT, the thickness of transistor gate oxide TOX and the transistor length L. Through Monte-Carlo simulation it was found that variation on these parameters lead to varied behaviour in resistive bridges because the gate drive strength of gates involved in a bridge is affected by the variation and so is the logic threshold voltage of gate inputs that are driven by the bridged nets.

The analysis conducted in this chapter shows how process variation leads to additional logic faults in the presence of a resistive bridge. It is shown that some of the additional logic faults lead to test escape (i.e. some defects pass the test undetected) for a test that is generated without consideration of process variation. This was demonstrated by fault simulating a test that had full bridge defect coverage for nominal values of the IC parameters for the set of process variation induced logic faults.

This chapter proposed a new metric to quantify the impact of process variation on test quality. The metric is called test robustness and records the probability of logic faults to occur and the range of bridge resistance that can be covered by detecting them. To calculate the test robustness, the parametric bridging fault model is applied on a large set of configurations of the parameter values. This process emulates the behaviour of corresponding manufactured ICs that are influenced by process variation.

An important observation was made while calculating the robustness for tests that were generated without consideration of process variation. It was found that tests with high robustness do not necessarily have to detect all process variation-induced logic faults, but should detect the logic faults that combine high probability to occur with large sets of bridge resistance that would otherwise not be covered. This observation was used to develop a process variation-aware test generation method that produce test sets with high test quality, i.e. few test escape. The proposed process variation-aware test

generation method achieves a user-specified test robustness with a small number of test patterns. The only previous work on test generation for resistive bridge defects under process variation used a fault model that was too abstract to make use of the above mentioned observation [155].

This chapter describes the proposed process variation-aware test generation tool along with the method for calculating the test robustness. Analysis and experimentation using these tools was presented using ISCAS benchmark designs that were synthesised and placed-and-routed for a 45nm gate library. The experiments used realistic bridge locations and bridge defect probabilities identified from the layout of the designs. The results show that test sets that are generated without consideration of process variation are inadequate in terms of test quality, particularly for small test sets. On the other hand, the proposed test generation method achieves high robustness with up to 36% smaller test sets compared with test sets that were generated without consideration of process variation.

Chapter 6

Conclusions and Future Work

This thesis addressed the impact of supply voltage and process variation on manufacturing testing in the context of low-power designs that often employ either multiple supply voltages or deep submicron technologies to save power. Testing multi-voltage designs provides a challenge because of supply voltage dependent detectability of some physical defects. Two such defects were addressed in the thesis: resistive bridges and full opens. Another test challenge is to cope with process variation, which affects the parameter values in deep submicron designs and cause unpredicted circuit behaviour in the presence of a defect. This thesis contributed by addressing these challenges as follows:

6.1 Contributions in this Thesis

This report provides contributions as follows:

- Chapter 2 - **An overview of test methods for supply voltage-dependent defects and process variation-aware testing**
  The literature review has shown that, while many defects are best detectable using delay fault testing at a very low supply voltage, other defects require static fault testing or testing at an elevated supply voltage. From this it was concluded that more than one supply voltage setting is required for testing designs that operate using multiple supply voltages. The literature can be seen as useful advice for how such tests should be conducted. However, no study has so far showed how to generate tests for designs with regard to supply voltage variation. On the topic of testing for resistive bridges in the presence of process variation, the existing test generation methods do not consider the probability of different logic faults in test generation.
- Chapter 3 - **Effective test generation for resistive bridge defects in designs with multiple supply voltages**

Resistive bridging faults cause a logic behaviour that depends on the bridge resistance and the supply voltage. The state-of-the-art test generation methods for resistive bridging faults use interval algebra to reason about the bridge resistance ranges that are covered when logic faults are detected. The interval algebra approach was extended in Chapter 3 to include the influence of supply voltage. To target the resistive bridging defects at the supply voltage setting where they manifest themselves as faults, a multi-voltage test generation method was presented, which achieves full defect coverage using supply voltage specific test sets. In general, resistive bridges are better detectable at a low supply voltage setting and more bridge resistance can be exposed by testing using such low voltage, but there are some bridge defects that require a higher supply voltage setting. This was reflected in the experimental results for the method, which showed a majority of test patterns assigned to the lowest available supply voltage setting and some additional test patterns assigned to one or more of the other supply voltage settings. This served as a proof-of-concept, that in testing under supply voltage variation, it is possible to generate supply voltage-specific test sets. The experiments were performed for synthesised benchmark circuits and realistic bridge locations using a tool flow including commercial tools and purpose-built ATPG and fault simulation software. It was shown that the proposed multi-voltage test generation method achieves full bridge defect coverage over the considered supply voltage settings and can be combined with a commercial ATPG.

- Chapter 4 - **An analysis of supply voltage-dependent detectability of full open defects**
  The full open defects are similar to the resistive bridging faults as another defect type with static behaviour, but full opens behave differently with supply voltage variation. The analysis used two different models for the voltage on the net that is affected by the open defect. The first model determined the voltage by the influence of capacitive coupling to neighbouring nodes and the second model determined the voltage based on gate tunnelling leakage currents. Experimental results were presented using a purpose-built simulation tool based on the two models and synthesised benchmark circuits. The results showed that, for both the considered models, there are many full opens with supply voltage-dependent behaviour. However the results also showed very few defects with supply voltage-dependent detectability. Thus, the vast majority of full open defects can be detected using any supply voltage setting. It was previously not known what supply voltage setting to use when testing for full opens under supply voltage variation, but now such tests can be confidently applied using any supply voltage setting.

- Chapter 5 - **Analysis of the impact of process variation on test quality and variation-aware test generation for resistive bridge defects**
  Unpredictable and unavoidable variations in the manufacturing process influence the behaviour of resistive bridging faults through two parameters, the logic thresh-

old voltage and the gate drive strength. The influence of process variation on the behaviour of resistive bridges was thoroughly analysed using extensive simulations using Cadence Spectre and recently reported data for the variation of the process parameters. A metric for quantifying the impact of process variation on test quality was presented, called test robustness, and it was shown that process variation lead to logic faults that would not occur for nominal IC parameters. Such process variation induced logic faults are potential test escapes, meaning that some defective ICs pass the test. To reduce test escape, a process variation-aware test generation tool was presented, which is guided by the test robustness metric and produces test sets that achieve a user-specified test quality target. The method improved on previously proposed approaches, because it explicitly considered the values of the IC parameters and the probability for different logic faults to occur. The method generates test sets with user-specified test quality and makes it possible to trade-off test quality and test set size, which was demonstrated in experimental results on synthesised benchmark circuits and realistic bridge locations. The experiments compared the result of a test that is generated without consideration of process variation with the result of applying the proposed method. It was found that the proposed method achieves equally high or higher test robustness with up to 36% smaller test sets. Furthermore, a combined test generation flow is discussed which would allow the proposed method to add test patterns to an existing test set to increase the test robustness.

The above contributions provide new, relevant and useful knowledge on how to test designs that operate using multiple supply voltages and ICs that are influenced by process variation. This knowledge is supported by extensive and realistic simulations of two defect types and software tools developed specifically for the purpose of the presented studies. This thesis has contributed low-cost and effective test methods in the context of logic testing of designs that use more than one supply voltage. Furthermore, the thesis has shown that it is necessary to consider process variation during test generation, otherwise test escape occur which leads to reliability problems. The future of testing low-power ICs will very likely combine several different test methods to cover a range of defect types while addressing a multitude of challenges. It is hoped that what has been proposed in this thesis will make useful contributions towards the development of solutions regarding the challenges of process variation and designs that operate using multiple supply voltages. Some of the software tools developed during the course of this Ph.D. project have already been used in studies that were not conducted by the author of this thesis. Such studies include research on diagnosis for resistive bridges and methods to reduce the number of supply voltage settings used while testing [92, 93, 94, 95].

6.2 Future Work

Based on the research presented in this thesis, a number of directions for future studies have been identified and are outlined in the following.

Development of Process Variation-Aware Fault Models

There is a need for process variation-aware fault models for bridges and opens. These should be abstract enough to allow time-efficient computation in terms of test generation and fault simulation, but as shown in Chapter 5 such fault models should also consider the probabilities of different logic faults. In this sense, the approach in Chapter 5 requires significant computation and is therefore limited to small designs. Further research should be conducted to develop process variation-aware fault models for bridges and opens that find the appropriate balance between attention to detail and computational efficiency. Research on the impact of process-variation on manufacturing testing should also include consideration of delay fault testing. In this context recent work has been reported addressing the problem that the critical path is no longer unique because of process variation influencing gate delay [151]. Other studies have addressed the problem of avoiding false delay fault failures that are caused by process variation [150]. These problems and other challenges raised by process variation should be considered in the development of process variation-aware fault models.

Development of Test Solutions for DVFS Designs

Developing test methods for devices that have Dynamic Voltage and Frequency Scaling (DVFS) involves considering the scaling of the clock frequency. The research presented in this thesis has considered logic testing with test sets applied at different supply voltage settings to achieve high test quality. Test cost has only been considered in terms of reducing the number of test patterns. In fact, the cost of test application varies with the supply voltage setting. While test application at a low supply voltage has been found to be effective for many defects, the cost of testing may increase at low supply voltages. Depending on the available test equipment and the prevailing test application procedures, testing using a supply voltage and clock frequency pair $< v, f >$ that implements a low-power mode may be costly in terms of test application time because the scan chain operation on a scan-enabled design has to be conducted slower than at a higher supply voltage. This cost needs to be considered as tests for DVFS designs are generated. Further research should be conducted to find tests for DVFS designs that achieve high test quality at a low total test cost. On a related topic, studies have been performed on the cost of very-low-voltage testing [114] and on selecting a supply voltage and clock frequency setting for saving power during test [91], but no study has addressed

the trade-off between test application time and test quality for DVFS designs. Scaling the clock frequency may also influence how delay fault testing should be performed. Some indications are given in the literature review in Section 2.3 for a selection of defect types. Further research should also include other defect types, for example feedback bridge defects.

Appendix A

Fault Site Schematic

The research presented in this thesis relies on experimentation using benchmark designs and examples that involve simulation of small circuits which include the defect. To enable such simulation it is useful to define what circuitry should be processed using analog simulation such as Spice and what circuitry can be simulated using gate-level logic simulation. This section defines the concept of a fault site, for which analog simulation is necessary. The concept of a fault site is used in Section 2.1.2, Section 3.1, Section 3.1.2, Section 3.1.5, Section 3.2.3, Section 4.1, Section 4.2.2, Section 5.2 and Section 5.3.3.

When considering a defect, the fault site is the circuitry immediately involved in activating the defect or the circuitry that is first affected by any faulty behaviour due to the defect. The fault site is the circuitry for which analog behaviour can be studied in the presence of a defect. Outside the fault site all signals are well defined logic signals. If no faulty behaviour is observed at the fault site, there cannot be any faulty behaviour for the full circuit. Either CPU-intensive analog simulation or a defect model is employed at the fault site to determine the faulty/non-faulty behaviour of the defect. By translating the analog values seen at the fault site into logic level signals, the rest of the circuit is simulated using logic gate-level simulation, which is less compute-intensive. To identify the circuitry that needs to be included in the fault site study, typically the gates and nets immediately before or after the defect needs to be included, and for some defect types also the physically adjacent circuitry. Figure A.1 shows the typical components of a fault site. It should be noted that fault site is defined as the minimum circuitry that can encapsulate the analog behaviour of the defect, while defect location is defined as the defect position in the schematic. The defect location is always a part of the fault site circuitry.

Figure A.1 introduces some concepts that define the fault site. The fault site consists of nets and gates that require analog simulation or a defect model to determine the logic behaviour. Nets that are internal to the fault-site are in the set $N_{internal}$. Some nets are important for the defect behaviour but are themselves not influenced by the defect.

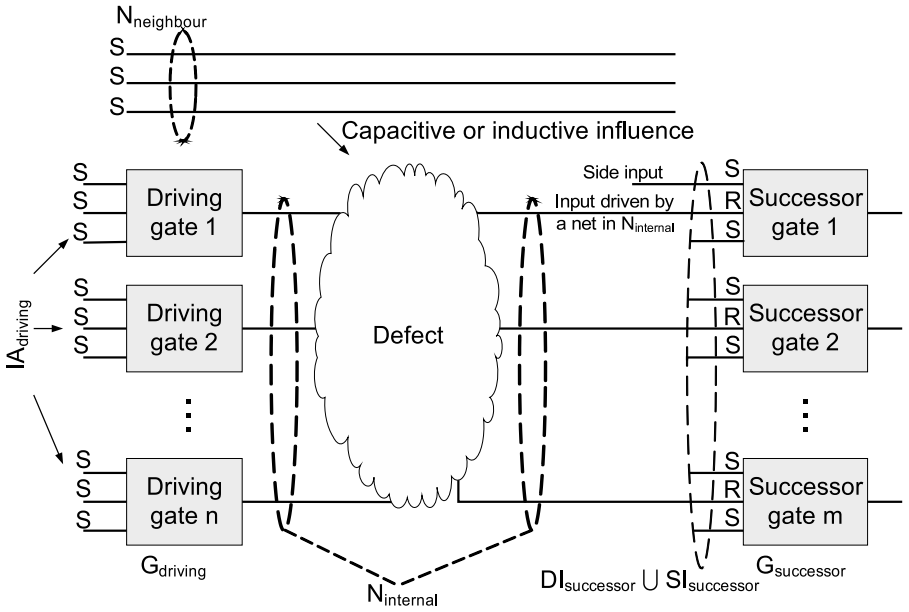


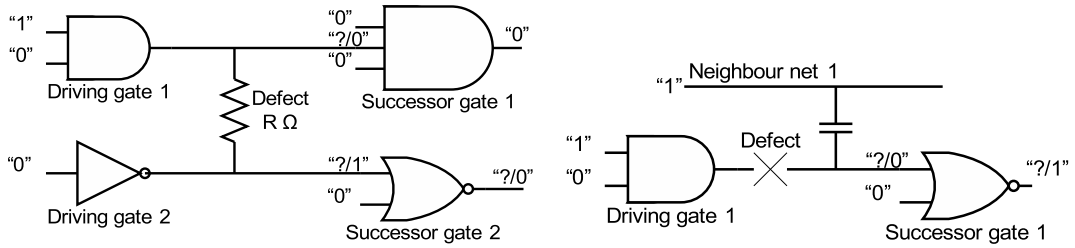
FIGURE A.1: Fault site components

These nets are in the set $N_{neighbour}$ and have capacitive or inductive influence on the defect. Fault-site internal nets are driven by gates, which are called the *driving gates*. The set of driving gates is $G_{driving}$. These driving gates are controlled by an input assignment $IA_{driving}$. The input assignment to a driving gate determine the intended logic value on corresponding net in $N_{internal}$. The gates that are driven by nets in $N_{internal}$ are called *successor gates*. The set of successor gates is $G_{successor}$.

Consider a net $n \in N_{internal}$. The voltage on n depends on the defect and the inputs to the fault site ($IA_{driving}$ and the logic assignment to $N_{neighbour}$) and can be found by analog simulation or by employing a defect model. The logic behaviour due to this voltage depends on how it compares with the logic threshold voltage of each input that is driven by n (Section 3.2.2). These inputs belong to gates in $G_{successor}$. The logic output of the fault site is measured at the output of the successor gate. For simulation that includes timing, i.e. delay related simulation, it is important to include the load capacitance of the successor gate output, as it will significantly influence the delay of the successor gate. For simulation of defects with static behaviour, as the defects considered in this thesis (resistive bridging faults and full open defects on interconnect), it is adequate to record the logic behaviour seen by the successor gate inputs. The successor gate inputs that are driven by nets in $N_{internal}$ are in the set $DI_{successor}$. This method of determining the logic output of the fault site leads to a smaller circuit to consider in analog simulation and leaves the other inputs to the successor gates outside of the fault site. These other inputs are called side inputs and are in the set $SI_{successor}$. Controlling these other inputs becomes part of the problem of propagating faulty signals from the fault site to primary outputs. For a fault-site it is possible to declare every input, belonging to driving gates and successor gates, as either a place for stimulus or

for logic responses. The inputs are marked by "S" for stimulus and "R" for responses in Figure A.1. Together, the stimulus and the responses encapsulate the analog behaviour of the defect with logic values.

The fault site definition given here does not include the possibility of feedback where the faulty signal from a defect affects the inputs of the fault site. In defining what is meant by fault site, it is possible to extend the presented definition to include feedback, but as the research in this thesis only considers non-feedback defects that definition is beyond the scope of the thesis.



(a) Bridge fault site

(b) Full open fault site

FIGURE A.2: Example fault sites

The general fault site definition provided above is used to provide the two examples in Figure A.2. The fault site in Figure A.2(a) shows a bridge fault site, where the defect is modelled by a resistance between two internal nets. An input assignment to the driving gates is given along with an assignment to the side inputs of the successor gates. In this example, the side inputs prevent propagation of a faulty signal through the 3 input AND gate and allow propagation through the NOR gate. The notation "?/0" and "?/1" says that the correct behaviour is Logic-0 and Logic-1 respectively, but the logic value in the presence of the defect is unknown. These logic values, seen by the driven gates are what needs to be defined by analog simulation or a defect model, and will depend on the defect resistance R. Similarly, Figure A.2(b) shows a full open fault site, where the defect is a complete break between the output of the AND gate and the input of the NOR gate, marked by X. As the upper input of the NOR gate in Figure A.2(b) is left without a driver, it may be influenced by neighbouring nets, as shown by "neighbour net 1" in the example. The logic values that encapsulate the analog behaviour of the defect are the input assignment to the AND gate, the logic assignment to the neighbouring net and the logic behaviour seen on the input of the NOR gate. The side input of the NOR gate is at Logic-0, which means that the fault signal from the defect is allowed to propagate to the NOR gate output.

Appendix B

SAT-Based ATPG

In the tools that have been developed during this Ph.D. project (Section 3.2 and Section 5.4) a solver for the Boolean Satisfiability problem [44] has been used to implement an ATPG-engine. This appendix explains how this type of solver can be used for ATPG.

The Boolean Satisfiability problem is the problem of determining if the variables of a given Boolean formula can be assigned in such a way as to make the formula evaluate to **true**. The problem also involves determining if the formula will always evaluate to **false** independent of the variable assignments. For example Equation B.1 shows a function h that evaluates to **true** for the assignment A=**true**, B=**false** and C=**true** and Equation B.2 shows a function g that is always false.

$$h = A \wedge -B \wedge C \tag{B.1}$$

$$g = A \wedge -A \tag{B.2}$$

Digital circuits can be seen as Boolean formula of the logic values on the primary inputs. Each output of the circuit is the result of a Boolean formula. The gates perform logic functions and Logic-1 maps to **true** and Logic-0 maps to **false**. For example, the Boolean formula in Equation B.1 corresponds to the circuit in Figure B.1.

To see how a Boolean formula can be constructed to use a Boolean Satisfiability solver for ATPG, consider the logic fault F that sets the signal on net X to Logic-0. To detect

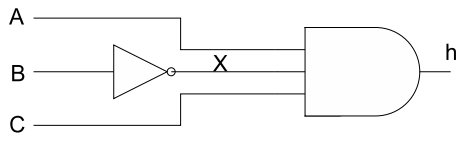


FIGURE B.1: A circuit that implements the Boolean formula in Equation B.1

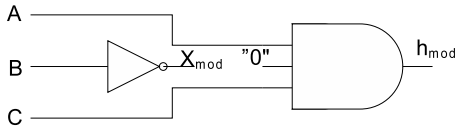


FIGURE B.2: The circuit in Equation B.1 modified by fault F

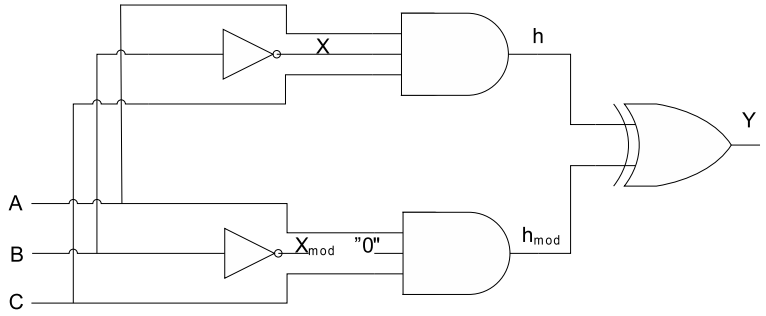


FIGURE B.3: A circuit for the Satisfiability problem of fault F in Figure B.1

this fault, there is one possible test pattern, A=1, B=0, C=1, which is expected to put Logic-1 on net h. This is the result that is anticipated for an ATPG. To construct a Boolean Satisfiability problem for this fault the circuit is duplicated so that one instance performs the fault-free function h and the other instance perform the function h_{mod} in Equation B.3 which is the result of the fault F modifying h.

$$h_{mod} = A \wedge 0 \wedge C \tag{B.3}$$

The two instances of the circuit, h in Equation B.1 and h_{mod} in Equation B.3, are compared with an XOR gate as shown in Figure B.3, which means that the output of the XOR gate is Logic-1 if the output of Equation B.1 and the output of Equation B.3 are different. It should be noted that the inputs corresponding to the variables A, B and C are given to both instances. The circuit in Figure B.3 implements the Boolean formula required to get the test pattern from the solver.

Even as Figure B.3 shows the digital circuit that implements the Boolean formula that should be given to the solver, there is one remaining obstacle, which is that there is no symbol for Logic-0 in the programming constructs used to interface with the solver [44]. Instead, Figure B.4 shows how adding an inverter and an AND gate tells the solver that Logic-0 is required for the input that is influenced by fault F. The AND gate requires the circuit in Figure B.3 to output Logic-1 if the final output of the circuit is to evaluate to Logic-1. It also requires the inversion of the Logic-0 (from fault F) to be Logic-1. The net that is marked *Constraint* is an additional variable that the solver will have to set to Logic-0 in order to make the final output Logic-1.

Figure B.4 concludes the search for the Boolean formula that can be used in a solver for the Boolean Satisfiability problem to implement an ATPG. The solver will find the

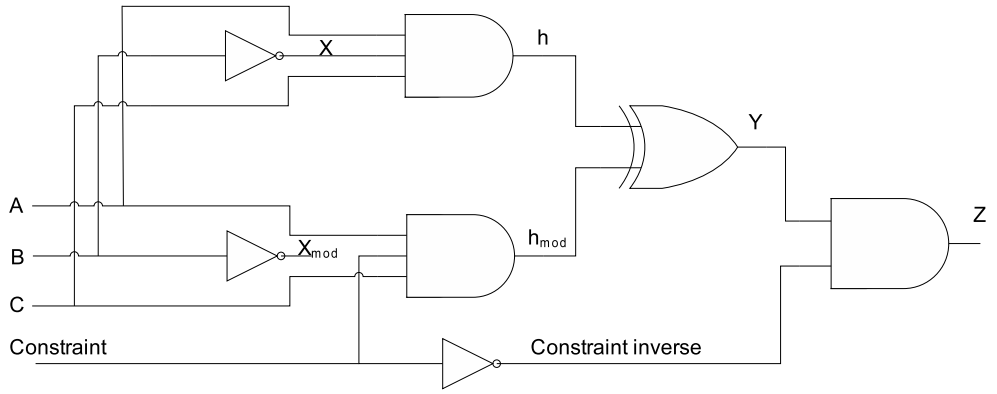


FIGURE B.4: A circuit that forces a Logic-0 on the input that is influenced by fault F

solution A=1, B=0, C=1 and these values represent the stimuli of the test pattern.

The solver from [44] uses conjunctive normal form to represent Boolean formula, but provides an interface of program constructs that allows the user of the solver to abstract from the particular form of the formula. These program constructs represent the generic logic gates, such as AND, OR, NAND, NOR, XOR and inverter. However compound gates like the AND-OR-INVERT (AOI) type of gates and multiplexers have to be translated for the solver in terms of the generic gates.

Bibliography

[1] G.E. Moore. Cramming more components onto integrated circuits. *Proceedings of the IEEE*, 86(1):82–85, 1998.

[2] Y. Cai, S.M. Reddy, I. Pomeranz, and B.M. Al-Hashimi. Battery-aware dynamic voltage scaling in multiprocessor embedded system. In *Proceedings of International Symposium on Circuits and Systems (ISCAS)*, pages 616–619, May 2005.

[3] K. Lahiri, A. Raghunathan, S. Dey, and D. Panigrahi. Battery driven system design: a new frontier in low power design. In *Proceedings of the International Conference on VLSI Design*, pages 261–267, January 2002.

[4] G. Theocharous, S. Mannor, N. Shah, P. Gandhi, B. Kveton, S. Siddiqi, and C.-H. Yu. Machine learning for adaptive power management. *Intel Technology Journal*, 10(4):299–310, November 2006.

[5] A. Andrei, M. Schmitz, P. Eles, Z. Peng, and B. M. Al-Hashimi. Overhead-conscious voltage selection for dynamic and leakage energy reduction of time-constrained systems. In *Proceedings Conference on Design, Automation and Test in Europe (DATE)*, pages 518–525, Paris, France, February 2004. IEEE Computer Society.

[6] ARM IEM. ARM1176JZF-S r0p2 technical reference manual. Documentation on ARM web site at www.arm.com, March 2006. http://www.arm.com/pdfs/DDI0301D_arm1176jzfs_r0p2_trm.pdf.

[7] Intel. Intel PXA270 processor electrical, mechanical and thermal specification data sheet. Data Sheet of Intel, January 2008. http://www.phytec.com/pdf/datasheets/PXA270_DS.pdf.

[8] AMD. AMD athlon 64 processor power and thermal data sheet. Data Sheet of AMD, January 2008. http://www.amd.com/us-en/assets/content_type/white_papers_and_tech_docs/30430.pdf.

[9] D.R. Ditzel. Power reduction using longrun2 in transmeta's efficeon processor. Conference presentation, January 2008. http://www.transmeta.com/pdfs/presentations/060517_mpf_spring_2006_tmta_longrun2.pdf.

[10] K. Flautner, S.K. Reinhardt, and T.N. Mudge. Automatic performance setting for dynamic voltage scaling. In *Mobile Computing and Networking*, pages 210–271, 2001.

[11] Q.A. Khan, S.K. Wadhwa, and K. Misri. A single supply level shifter for multi-voltage systems. In *Proceedings of the International Conference on VLSI Design*, pages 4–7, January 2006.

[12] D. Ernst, N.S. Kim, S. Das, R. Rao, T. Pham, C. Ziesler, D. Blaauw, T. Austin, K. Flautner, and T. Mudge. Razor: a low-power pipeline based on circuit-level timing speculation. In *Proceedings of the International Symposium on Microarchitecture (MICRO)*, pages 7–18, December 2003.

[13] 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.

[14] 0.12μm vlsi design kit. ST Microelectronics product, December 2005. http://cmp.imag.fr/products/ic/?p=STHCMOS9.

[15] OSU FreePDK. Oklahoma State University, 2008. http://avatar.ecen.okstate.edu/projects/scells/OSUFreePDK.php.

[16] M. Keating, D. Flynn, R. Aitken, A. Gibbons, and K. Shi. *Low power methodology manual: for system-on-chip design*. Springer, 2007.

[17] E. Macii. *Ultra low-power electronics and design*. Springer, 2004.

[18] H. J. M. Veendrick. *Deep-submicron CMOS ICs: from basics to ASICs*. Springer, 2000.

[19] International technology roadmap for semiconductors. ITRS website, March 2009. http://www.itrs.net/Links/2008ITRS/Home2008.htm.

[20] K. Bernstein, D. J. Frank, A. E. Gattiker, W. Haench, 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):433–450, 2006.

[21] A. Asenov. Random dopant induced threshold voltage lowering and fluctuations in sub-0.1 μm MOSFETs. *Transactions on Electron Devices (TED)*, 45(12):2505–2513, December 1998.

[22] S. R. Nassif. Modeling and analysis of manufacturing variations. In *Proceedings of the IEEE custom integrated circuits Conference*, pages 223–228, May 2001.

[23] O. Oldiges, Q. Lin, K. Petrillo, M. Sanchez, M. Ieong, and M. Hargrove. Modeling line edge roughness effects in sub 100 nanometer gate length devices. In *Proceedings of the International Conference on simulation of semiconductor processes and devices (SISPAD)*, pages 131–134, September 2002.

[24] S. Bhunia, S. Mukhopadhyay, and K. Roy. Process variations and process-tolerant design. In *Proceedings of the International Conference on VLSI design (VLSID)*, pages 699–704, January 2007.

[25] M. Bühler, J. Koehl, J. Bickford, J. Hibbeler, U. Schlichtmann, R. Sommer, M. Pronath, and A. Ripp. DFM/DFY design for manufacturability and yield - influence of process variations in digital, analog and mixed-signal circuit design. In *Proceedings of the Conference on Design, Automation and Test in Europe (DATE)*, pages 387–392, March 2006.

[26] G. Roy, F. Adamu-Lema, A. R. Brown, S. Roy, and A. Asenov. Intrinsic parameter fluctuations in conventional MOSFETs at the scaling limit: a statistical study. *Journal of Physics: Conference Series*, 38(1):188–191, 2006.

[27] S. Kundu, A. Sreedhar, and A. Sanyal. Forbidden pitches in sub-wavelength lithography and their implications on design. *Journal of Computer-Aided Materials Design*, 14(1), April 2007.

[28] L. H. A. Leunissen, W. G. Lawrence, and M. Erkcen. Line edge roughness: experimental results related to a two-parameter model. *Microelectronic Engineering*, 73–74:265–270, 2004.

[29] A. Sreedhar, A. Sanyal, and S. Kundu. On modeling and testing of lithography related open faults in nano-cmos circuits. In *Proceedings of the European Conference on Design and Test (DATE)*, pages 616–621, March 2008.

[30] T. Chen and S. Naffziger. Comparison of adaptive body bias (abb) and adaptive supply voltage (asv) for improving delay and leakage under the presence of process variation. *Transactions on Very Large Scale Integration (VLSI) Systems*, 11(5):888–899, 2003.

[31] S. H. Choi, B. C. Paul, and K. Roy. Novel sizing algorithm for yield improvement under process variation in nanometer technology. In *Proceedings of design automation Conference (DAC)*, pages 454–459, June 2004.

[32] K.J. Kuhn. Reducing variation in advanced logic technologies: approaches to process and design for manufacturability of nanoscale CMOS. In *IEEE International Electron Devices Meeting (IEDM)*, pages 471–474, December 2007.

[33] N. H. E. Weste and K. Eshraghian. *Principles of CMOS VLSI design*. Addison-Wesley Pub. Co., 1993.

[34] M. Abramovici, M. A. Breuer, and A. D. Friedman. *Digital systems testing and testable design*. IEEE press, 1994.

[35] R. J. Lipp. Limitations of the stuck-at fault model as an accurate measure of CMOS IC quality and a proposed schematic level fault model. In *Proceedings of the Custom Integrated Circuits Conference*, May 1989.

[36] J.H. Patel. Stuck-at fault: a fault model for the next millenium. In *Proceedings of IEEE International Test Conference (ITC)*, October 1998.

[37] W. Needham, C. Prunty, and E. H. Yeoh. High volume microprocessor test escapes, an analysis of defects our tests are missing. In *Proceedings of the International Test Conference (ITC)*, pages 25–34, October 1998.

[38] O. V. Maiuri and W. R. Moore. Implications of voltage and dimension scaling on CMOS testing: the multidimensional testing paradigm. In *Proceedings of VLSI test Symposium (VTS)*, pages 22–27, Princeton, NJ, USA, May 1998.

[39] M. Renovell, F. Azais, and Y. Bertrand. Test escapes: analysis of short defects. In *Proceedings of IEEE Brazilian Symposium on Integrated Circuit Design*, pages 477–486, September 1999.

[40] E. J. McCluskey, A. A. Al-Yamani, C.-M. J. Li, C.-W. Tseng, E. H. Volkerink, F.-F. Ferhani, E. Li, and S. Mitra. ELF-Murphy data on defects and test sets. In *Proceedings of VLSI test Symposium (VTS)*, pages 16–22, April 2004.

[41] M. E. Amyeen, S. Venkataraman, A. Ojha, and S Lee. Evaluation of the quality of n-detect scan ATPG patterns on a processor. In *Proceedings of the International Test Conference (ITC)*, pages 669–678, October 2004.

[42] H. Tang, G. Chen, S. M. Reddy, C. Wang, J. Rajski, and I. Pomeranz. Defect aware test patterns. In *Proceedings of the Conference on Design, Automation and Test in Europe (DATE)*, pages 450–455, March 2005.

[43] S. Biswas, P. Srikanth, R. Jha, S. Mukhopadhyay, A. Patra, and D. Sarkar. On-line testing of digital circuits for n-detect and bridging fault models. In *Proceedings of the Asian test Symposium (ATS)*, pages 88–93, December 2005.

[44] zChaff Boolean Satisfiability problem solver, March 2007. http://www.princeton.edu/~chaff/zchaff.html.

[45] IEEE Standards Association. IEEE Standards Description: 1149.1-1990. On the IEEE web site, July 2009. http://standards.ieee.org/reading/ieee/std_public/description/testtech/1149.1-1990_desc.html.

[46] J. Savir and S. Patil. On broad-side delay test. *IEEE Transactions on VLSI Systems*, 2(3):368–372, 1994.

[47] J. Savir. Skewed-load transition test: Part 1, calculus. In *Proceedings IEEE International Test Conference (ITC)*, pages 705–713, September 1992.

[48] R. Rodríguez-Montañés and J. Figueras. Estimation of the defective IDDQ caused by shorts in deep-submicron CMOS ICs. In *Proceedings of the Conference on Design, Automation and Test in Europe (DATE)*, pages 490–494, Paris, France, February 1998.

[49] M. Sachdev. Deep sub-micron IDDQ testing: Issues and solutions. In *Proceedings of the European Conference on Design and Test (DATE)*, pages 271–278, March 1997.

[50] B. Kruseman, S. vand den Oetelaar, and J. Rius. Comparison of IDDQ testing and very-low voltage testing. In *Proceedings of the International Test Conference (ITC)*, pages 964–973, October 2002.

[51] T. S. Kim, S. H. Hong, and J. B. Kim. A current sensing circuit for IDDQ testing. In *Proceedings of the International Conference on ASIC*, pages 666–669, October 2005.

[52] S. Kim, S. Chakravarty, and B. Vinnakota. An analysis of the delay defect detection capability of the ECR test method. In *Proceedings of the International Test Conference (ITC)*, pages 1060–1069, October 2000.

[53] W. Jiang and E. Peterson. Performance comparison of VLV, ULV and ECR tests. *Journal of Electronic Testing: Theory and Applications*, 19(2):137–147.

[54] H. Hao and E.J. McCluskey. "Resistive shorts" within CMOS gates. In *Proceedings of the International Test Conference (ITC)*, Nashville, TN, USA, October 1991.

[55] Y. Miura and S. Seno. Behaviour analysis of internal feedback bridging faults in CMOS circuits. *Journal of Electronic Testing: Theory and Applications (JETTA)*, 18(2):109–120, 2002.

[56] B. K. Koch and K. D. Müller-Glaser. An examination of feedback bridging faults in digital CMOS circuits. In *Proceedings of International Symposium on Circuits and Systems (ISCAS)*, pages 1527–1530, May 1993.

[57] B. Chess and T. Larrabee. Generating test patterns for bridge faults in CMOS ICs. In *Proceedings of the European Conference on Design Automation*, pages 165–170, March 1994.

[58] M. Roca and A. Rubio. Current testability analysis feedback bridging faults in CMOS circuits. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, (TCAD)*, 14(10):1299–1305, 1995.

[59] P. Dahlgren. Switch-level bridging fault simulation in the presence of feedback. In *Proceeding of the International Test Conference (ITC)*, pages 363–371, October 1998.

[60] A. Keshk, Y. Miura, and K. Kinoshita. Simulation of resistive bridging fault to minimize the presence of intermediate voltage and oscillation in CMOS circuits. In *Proceedings of the Asian test Symposium (ATS)*, pages 120–124, December 2000.

[61] I. Polian, P. Engelke, M. Renovell, and B. Becker. Modeling feedback bridging faults with non-zero resistance. In *Proceedings of the European Test Workshop (ETW)*, pages 91–96, May 2003.

[62] M. Roca and A. Rubio. Current testability analysis of feedback bridging faults in CMOS circuits. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, (TCAD)*, 14(10):1299–1305, October 1995.

[63] E. Isern and J. Figueras. Test generation with high coverages for quiescent current test of bridging faults in combinational circuits. In *Proceedings of the International Test Conference (ITC)*, pages 73–82, October 1993.

[64] R. Rodríguez-Montañés, E.M.J.G. Bruls, and J. Figueras. Bridging defects resistance measurements in a CMOS process. In *Proceeding of the International Test Conference (ITC)*, pages 892–899, September 1992.

[65] V. Krishnaswamy, A.B. Ma, and P. Vishakantaiah. A study of bridging defect probabilities on a pentium (tm) 4 cpu. In *Proceedings of the International Test Conference (ITC)*, pages 688–695, Baltimore, MD, USA, October 2001.

[66] Z. Li, X. Lu, W. Qiu, W. Shi, and D.M.H. Walker. A circuit level fault model for resistive opens and bridges. In *Proceedings of VLSI Test Symposium (VTS)*, pages 379–384, April 2003.

[67] W. Moore, G. Gronthoud, K. Baker, and M. Lousberg. Delay-fault testing and defects in deep sub-micron ICs - Does critical resistance really mean anything? In *Proceedings of the International Test Conference (ITC)*, pages 95–104, Atlantic City, NJ, USA, October 2000.

[68] A. Keshk, K. Kinoshita, and Y. Miura. IDDQ current dependency on test vectors and bridging resistance. In *Proceedings of the Asian test Symposium (ATS)*, pages 158–163, November 1999.

[69] R. Rodríguez-Montañés, P. Volf, and J. Pineda de Gyvez. Resistance characterization for weak open defects. *IEEE Design & Test of Computers*, 19(5):18–26, 2002.

[70] J.C.-M. Li, C.-W. Tseng, and E.J. McCluskey. Testing for resistive opens and stuck opens. In *Proceedings of the International Test Conference (ITC)*, pages 1049–1058, October 2001.

[71] J.C.-M. Li and E.J. McCluskey. Diagnosis of resistive-open and stuck-open defects in digital CMOS ICs. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, (TCAD)*, 24(11):1748–1759, 2005.

[72] R. Rodríguez-Montañés, D. Arumi, J. Figueras, S. Einchenberger, C. Hora, B. Kruseman, M. Lousberg, and A.K. Majhi. Diagnosis of full open defects in interconnecting lines. In *Proceedings IEEE VLSI Test Symposium (VTS)*, pages 158–166, May 2007.

[73] Y. Miura. Detection of interconnect open faults with unknown values by ramp voltage application. In *Proceedings of the Asian test Symposium (ATS)*, pages 55–62, November 2006.

[74] M. Renovell, M. Comte, I. Polian, P. Engelke, and B. Becker. A specific ATPG technique for resistive open with sequence recursive dependency. In *Proceedings of the Asian test Symposium (ATS)*, pages 273–278, November 2006.

[75] W. Maly, P.K. Nag, and P. Nigh. Testing oriented analysis of CMOS ICs with opens. In *Proceedings IEEE International Conference on Computer-Aided Design (ICCAD)*, pages 344–347, November 1988.

[76] V. Champac and J. Figueras. Testability of floating gate defects in sequential circuits. In *Proceedings IEEE VLSI Test Symposium (VTS)*, pages 202–207, May 1995.

[77] S. Rafiq, A. Ivanov, S. Tabatabaei, and M. Renovell. Testing for floating gates defects in CMOS circuits. In *Proceedings IEEE Asian Test Symposium (ATS)*, pages 228–236, December 1998.

[78] M. Pronath, H. Graeb, and K. Antreich. A test design method for floating gate defects (FGD) in analog integrated circuits. In *Proceedings of Conference on Design, Automation and Test in Europe (DATE)*, pages 78–83, March 2002.

[79] 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.

[80] H. Konuk. Voltage- and current-based fault simulation for interconnect open defects. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, (TCAD)*, 18(12):1768–1779, 1999.

[81] J.C.-M. Li and E.J. McCluskey. Testing for tunneling opens. In *Proceedings of the International Test Conference (ITC)*, pages 85–94, October 2000.

[82] D. Arumi, R. Rodríguez-Montañés, and J. Figueras. Experimental characterization of CMOS interconnect open defects. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, (TCAD)*, 27(1):123–136, January 2008.

[83] R. Rodríguez-Montañés, D. Arumí, J. Figueras, S. Eichenberger, C Hora, and B. Kruseman. Impact of gate tunneling leakage on cmos circuits with full open defects. *Electronics letters*, 43, October 2007.

[84] D. Arumí, R. Rodríguez-Montañés, J. Figueras, S. Eichenberger, C. Hora, and B. Kruseman. Full open defects in nanometric CMOS. In *Proceedings of VLSI test Symposium (VTS)*, pages 119–124, April 2008.

[85] S.-Y. Huang. Diagnosis of byzantine open-segment faults. In *Proceedings IEEE Asian Test Symposium (ATS)*, pages 248–253, November 2002.

[86] P.M. Rosinger, B.M. Al-Hashimi, and N. Nicolici. Power profile manipulation: a new approach for reducing test application time under power constraints. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, (TCAD)*, 21(10):1217–1225, October 2002.

[87] N. Nicolici and B. M. Al-Hashimi. *Power-constrained testing of VLSI circuits.* Springer, 2003.

[88] P. Girard. Survey of low-power testing of VLSI circuits. *IEEE Design & test of computers*, 19(3):82–92, 2002.

[89] P. M. Rosinger, B. M. Al-Hashimi, and K. Chakrabarty. Thermal-safe test scheduling for core-based system-on-chip. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, (TCAD)*, 25(11):2502–2512, 2006.

[90] P.M. Rosinger, P.T. Gonciari, B.M. Al-Hashimi, and N. Nicolici. Analysing trade-offs in scan power and test data compression for systems-on-a-chip. *IEE Proceedings of Computers and Digital Techniques*, 149(4):188–196, July 2002.

[91] 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. In *Proceedings of IEEE International Test Conference (ITC)*, pages 1–9, October 2007.

[92] S. Khursheed, P. Rosinger, B.M. Al-Hashimi, S.M. Reddy, and P. Harrod. Bridge defect diagnosis for multiple-voltage design. In *Proceedings of the IEEE European Test Symposium (ETS)*, pages 99–104, May 2008.

[93] S. Khursheed, U. Ingelsson, P. Rosinger, B.M. Al-Hashimi, and P. Harrod. Bridging fault test method with adaptive power management awareness. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, June 2008.

[94] S. S. Khursheed, B. Al-Hashimi, S. M. Reddy, and P. Harrod. Diagnosis of multiple-voltage design with bridge defect. *(In press) IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, 2009.

[95] S. S. Khursheed, B. Al-Hashimi, and P. Harrod. Test cost reduction for multiple-voltage designs with bridge defects through gate-sizing. In *(In press) Proceedings of the Conference on Design, Automation and Test in Europe (DATE)*, April 2009.

[96] I. Polian, S. Kundu, J.-M. Galliere, P. Engelke, M. Renovell, and B. Becker. Resistive bridging fault model evolution from conventional to ultra deep submicron technologies. In *Proceedings of the VLSI Test Symposium (VTS)*, pages 343–348, May 2005.

[97] 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 of the IEEE European Test Symposium (ETS)*, pages 22–27, May 2005.

[98] S. Chakravarty, Y. Chang, H. Hoang, S. Jayaraman, S. Picano, C. Prunty, E. W. Savage, R. Sheikh, E. N. Tran, and K. Wee. Experimental evaluation of bridge patterns for a high performance microprocessor. In *Proceedings of the VLSI Test Symposium (VTS)*, pages 337–342, May 2005.

[99] M. Favalli and M. Dalpasso. Bridging fault modeling and simulation for deep submicron CMOS ICs. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, (TCAD)*, 21(8):941–953, August 2002.

[100] B. Chess, C. Roth, and T. Larabee. On evaluating competing bridge fault models for CMOS ICs. In *Proceedings of VLSI test Symposium (VTS)*, pages 446–451, April 1994.

[101] B. Chess and T. Larrabee. Bridge fault simulation strategies for CMOS integrated circuits. In *Proceedings of the International Conference on design automation (DAC)*, pages 458–462, 1993.

[102] P.C. Maxwell and R.C. Aitken. Biased voting: A method for simulating CMOS bridging faults in the prescence of variable gate logic thresholds. In *Proceedings of the International Test Conference (ITC)*, pages 64–72, October 1993.

[103] M. Dalpasso, M. Favalli, P. Olivo, and B Riccó. Fault simulation of parametric bridging faults in CMOS IC's. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, (TCAD)*, 12(9):1403–1410, 1993.

[104] 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.

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

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

[107] P. Engelke, I. Polian, M. Renovell, and B. Becker. Simulating resistive bridging and stuck-at faults. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, (TCAD)*, 25:2181–2192, October 2006.

[108] P. Engelke, B. Braitling, I. Polian, M. Renovell, and B. Becker. SUPERB: simulator utilizing parallel evaluation of resistive bridges. In *Proceedings of the Asian test Symposium (ATS)*, pages 433–438, October 2007.

[109] 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.

[110] 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.

[111] 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 2006.

[112] 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.

[113] Y. Liao and D. M. H. Walker. Fault coverage analysis for physically-based CMOS bridging faults at different power supply voltages. In *Proceedings of the International Test Conference (ITC)*, pages 767–775, October 1996.

[114] 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.

[115] R. Rodríguez-Montañés, D. Arumi, and J. Figueras. Effectiveness of very low voltage testing of bridging defects. *Electronic letters*, 42(19), 2006.

[116] P. Engelke, I. Polian, M. Renovell, S. Kundu, B. Seshadri, and B. Becker. On detection of resistive bridging defects by low-temperature or low-voltage testing. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, (TCAD)*, 27(2):327–338, February 2008.

[117] C.Y. Lee and D.M.H. Walker. PROBE: a PPSFP simulator for resistive bridging faults. In *Proceedings of VLSI test Symposium (VTS)*, pages 105–110, May 2000.

[118] W. Zou, W.-T. Cheng, and S.M. Reddy. Interconnect open defect diagnosis with physical information. In *Proceedings of the Asian test Symposium (ATS)*, pages 203–208, November 2006.

[119] Y. Sato, H. Takahashi, Y. Higami, and Y. Takamatsu. Failure analyssis of open faults by using detecting/un-detecting information on tests. In *Proceedings of the Asian test Symposium (ATS)*, pages 222–227, November 2004.

[120] Y. Zhao and M. H. White. Modeling of direct tunneling current through interfacial oxide and high-k gate stacks. *Solid-State Electronics*, 48(10–11):1801–1807, 2003.

[121] C.L. Henderson, J.M. Soden, and C.F. Hawkins. The behavior and testing implications of CMOS IC logic gate open circuits. In *Proceedings IEEE International Test Conference (ITC)*, pages 302–310, October 1991.

[122] V.H. Campac and A. Zenteno. Detectability conditions for interconnection open defects. In *Proceedings IEEE VLSI Test Symposium (VTS)*, pages 305–311, May 2000.

[123] D. Arumi, R. Rodríguez-Montañés, and J. Figueras. Defective behaviours of resistive opens in interconnect lines. In *Proceedings of the IEEE European Test Symposium (ETS)*, pages 28–33, May 2005.

[124] M. Renovell and Y. Bertrand. Test strategy sensitivity to defect parameters. In *Proceedings of the International test Conference (ITC)*, pages 607–616, November 1997.

[125] H. Konuk and F.J. Ferguson. Oscillation and sequential behaviour caused by interconnect opens in digital CMOS circuits. In *Proceedings IEEE International Test Conference (ITC)*, pages 597–606, 1997.

[126] R. Gomez, A. Giron, and V. Champac. Test of interconnection opens considering coupling signals. In *Proceedings IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems (DFT)*, pages 247–255, October 2005.

[127] S. Spinner, I. Polian, P. Engelke, B. Becker, M. Keim, and W.-T. Cheng. Automatic test pattern generation for interconnect open defects. In *Proceedings of VLSI test Symposium (VTS)*, pages 181–186, April 2008.

[128] S. M. Reddy, I. Pomeranz, H. Tang, S. Kajihara, and K. Konishita. On testing of interconnect open defects in combinational logic circuits with stems of large fanout. In *Proceedings of the International test Conference (ITC)*, pages 83–89, October 2002.

[129] S. Ghosh and F.J. Ferguson. Estimating detection probability of interconnect opens using stuck-at tests. In *Proceedings of the Great Lakes Symposium on VLSI*, pages 254–259, April 2004.

[130] 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 1996.

[131] 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.

[132] H. Hao and E.J. McCluskey. Analysis of gate oxide shorts in CMOS circuits. *IEEE Transactions on Computers*, 42(12):1510–1516, December 1993.

[133] H. Yan and A.D. Singh. A delay test to differentiate resistive interconnect faults from weak transistor defects. In *Proceedings of the International Conference on VLSI Design (VLSID)*, pages 47–52, Kolkata, India, January 2005.

[134] 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.

[135] M. Rodriguez-Irago, J.J. Rodriguez Andina, F. Vargas, J. Semião, I.C. Teixeira, and J.P. Teixeira. On the use of multi-clock, multi-V_{dd} and multi-temperature schemes to improve dynamic fault detection in digital systems. In *Informal Digest of the IEEE European Test Symposium (ETS)*, Southampton, UK, May 2006.

[136] N.B. Zain Ali, M. Zwolinski, and B.M. Al-Hashimi. Testing of level shifters in multiple voltage designs. In *Proceedings of the International Conference on Electronics, Circuits and Systems*, December 2007.

[137] J.T.-Y. Chang and E.J. McCluskey. Short voltage elevation (SHOVE) testfor weak CMOS ICs. In *Proceedings of the VLSI Test Symposium (VTS)*, pages 446–451, April 1997.

[138] C.-W. Tseng, R. Chen, P. Nigh, and E.J. McCluskey. MINVDD testing for weak CMOS ICs. In *Proceedings of the VLSI Test Symposium (VTS)*, pages 339–344, April 2001.

[139] A. K. B. A'ain, A. H. Bratt, and A. P. Dorey. Exposing floating gate defects in analogue CMOS circuits by power supply voltage control testing technique. In *Proceedings of the International Conference on VLSI design*, pages 239–242, January 1995.

[140] D. Ager and J. Henderson. The use of marginal voltage mesurements to detect and locate defects in digital microcircuits. In *Proceedings of the Reliability Physics Symposium*, pages 139–148, 1981.

[141] R. Madge, B.H. Goh, and V. Rajagopalan. Screening minVDD outliers using feed-forward voltage testing. In *Proceedings of the International Test Conference (ITC)*, pages 673–682, October 2002.

[142] S. Kundu, S. Sengupta, and R. Galivanche. Test challenges in nanometer technologies. In *Proceedings of the European Test Workshop (ETW)*, pages 83–90, 2000.

[143] K.-T. Cheng, S. Dey, M. Rodgers, and K. Roy. Test challenges for deep submicron technologies. In *Proceedings of Design Automation Conference (DAC)*, pages 142–149, June 2000.

[144] M. Nourani and A. Radhakrishnan. Modeling and testing process variation in nanometer CMOS. In *Proceedings of the International Test Conference (ITC)*, pages 1–10, October 2006.

[145] D. Acharyya and J. Plusquellic. Hardware results demonstrating defect detection using power supply signal measurements. In *Proceedings of the VLSI test Symposium (VTS)*, pages 433–438, May 2005.

[146] K. Baker, G. Gronthoud, M. Lousberg, I. Schanstra, and C. Hawkins. Defect-based delay testing of resistive vias-contacts a critical evaluation. In *Proceedings of the International Test Conference (ITC)*, pages 467–476, September 1999.

[147] G.M. Luong and D.M.H. Walker. Test generation for global delay faults. In *Proceedings of the International Test Conference (ITC)*, pages 433–442, October 1996.

[148] X. Lu, Z. Li, W. Qui, D. M. H. Walker, and W. Shi. PARADE: parametric delay evaluation under process variation. In *Proceedings of the International Symposium on quality electronic design (ISQED)*, pages 276–280, March 2004.

[149] X. Lu, Z. Li, W. Qiu, D.M.H. Walker, and W. Shi. Longest-path selection for delay test under process variation. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, (TCAD)*, 24(12):1924–1929, December 2005.

[150] V. R. Devanathan, C. P. Ravikumar, and V. Kamakoti. Variation-tolerant, power-safe pattern generation. *IEEE design & test of computers*, 24(4):374–384, July 2007.

[151] V. Iyengar, J. Xiong, S. Venkatesan, V. Zolotov, D. Lackey, P. Habitz, and C. Visweswariah. Variation-aware performance verification using at-speed structural test and statistical timing. In *Proceedings of the International Conference on computer-aided design (ICCAD)*, pages 405–412, November 2007.

[152] M. Hashizume, H. Yotsuyanagi, and M. Ichimiya. High speed IDDQ test and its testability for process variation. In *Proceedings of the Asian test Symposium (ATS)*, pages 344–349, December 2000.

[153] A. Keshavarzi, J.W. Tschanz, S. Narendra, W.R. De, V. annd Daasch, K. Roy, M. Sachdev, and C.F. Hawkins. Leakage and process variation effects in current

testing on future CMOS circuits. *IEEE Design & Test of Computers*, 19(5):36–43, 2002.

[154] G. Devarayanadurg and M. Soma. Analytical fault modeling and static test generation for analog ICs. In *Proceedings of the International Conference on Computer Aided Design*, pages 44–47, 1994.

[155] M. Favalli and M. Dalpasso. High quality test vectors for bridging faults in the presences of IC's parameters variations. In *Proceedings of IEEE International Symposium on Defect and Fault-Tolerance in VLSI systems (DFT)*, pages 448–456, September 2007.

[156] Lp solver, August 2008. http://lpsolve.sourceforge.net/.

[157] Y. Sato, K. Sugiura, R. Shimoda, Y. Yoshizawa, K. Norimatsu, and M. Sanada. Defect diagnosis - reasoning methodology. In *Proceedings of the Asian test Symposium (ATS)*, pages 209–214, November 2006.

[158] R.S. Guindi and F.N. Najm. Design techniques for gate-leakage reduction in cmos circuits. In *Int'l Symp. on Quality Electronic Design*, pages 61–65, March 2003.

[159] 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. *IEEE Electron Device Letters*, 18(5):209–211, May 1997.

[160] Predictive technology model. Released by Arizona State University, April 2008. http://www.eas.asu.edu/˜ptm.

[161] Star-Hspice Manual. LEVEL 54 BSIM4.0 Model. On University of California web site, July 2009. http://www.ece.uci.edu/docs/hspice/hspice_2001_2-173.html.

[162] I. D. Dinov, N. Christou, and J. Sanchez. Central limit theorem: New SOCR applet and demonstration activity. *Journal of Statistics Education*, 16(2), 2008. http://www.amstat.org/publications/jse/v16n2/dinov.html.

[163] P. Nigh and A.E. Gattiker. Test method evaluation experiments and data. In *Proceeding of the International Test Conference (ITC)*, pages 454–463, October 2000.

[164] F. Fink, K. Fuchs, and M. H. Schulz. Robust and nonrobust path delay fault simulation by parallelprocessing of patterns. *IEEE Transactions on Computers*, 41(12), December 1992.