UNIVERSITY OF SOUTHAMPTON

An Investigation of Delay Fault Testing for Multi Voltage Design

by

Noohul Basheer Zain Ali

A thesis submitted in partial fulfillment for the degree of Doctor of Philosophy

in the

Faculty of Engineering, Science and Mathematics

School of Electronics and Computer Science

January 2009

UNIVERSITY OF SOUTHAMPTON

ABSTRACT

FACULTY OF ENGINEERING, SCIENCE AND MATHEMATICS

SCHOOL OF ELECTRONICS AND COMPUTER SCIENCE

Doctor of Philosophy

by Noohul Basheer Zain Ali

Multi Voltage Design(MVD) has been successfully applied in contemporary processors as a technique to reduce energy consumption. This work is aimed at finding a generalised delay testing method for MVD. There has been little work to date on testing such systems, but testing the smallest number of operating voltages reduces testing costs. In the initial stage, the impact of varying supply voltage on different types of physical defects is analysed. Simulation results indicate that it is necessary to conduct test at more than one operating voltage and the lowest operating voltage does not necessarily give the best fault coverage. The second part of this work is related to the issues in the testing of level shifters in a MVD environment. The testing of level shifters was analysed to determine if high test coverage can be achieved at a single supply voltage. Resistive opens and shorts were considered and it was shown that, for testing purposes, consideration of purely digital fault effects is sufficient. Multiple faults were also considered. In all cases, it can be concluded that a single supply voltage is sufficient to test the level shifters. To further enhance the quality of test, we have proposed fault modelling and simulations using VHDL-AMS. Our simulation results show that the model derived using simplified VHDL-AMS gives acceptable results and significantly reduces the fault simulations time.

Contents

List of Figures

List of Tables

Nomenclature

$ASIC$	Application-Specific Integrated Circuit
ATE	Automated Test Equipment
$ATPG$	Automatic Test Pattern Generation
AVS	Adaptive Voltage Scaling
$BIST$	Built-In Self Test
$CMLS$	Contention Mitigated Level Shifter
$CMOS$	Complementary metaloxidesemiconductor
$CVLS$	Conventional Level Shifter
DFT	Design for Testability
DSM	Deep Sub Micron
$DVFS$	Dynamic Voltage and Frequency Scaling
DVS	Dynamic Voltage Scaling
IC	Integrated Circuits
$ITRS$	International Roadmap for Semiconductor
LOC	Launch-On-Capture
LOS	Launch-On-Shift
MUX	Multiplexer
MVD	Multi Voltage Design
MVS	Multi-level Voltage Scaling
NVC	Non Controlling Value
PI	Primary Input

PO	Primary Output
RBF	Resistive Bridging Fault
SFA	Single Fault Assumption
$SISO$	Scan-in, Scan-Out
$SPICE$	Simulation Program with Integrated Circuit Emphasis
$SSFM$	Single Stuck-at Fault Model
SVS	Static Voltage Scaling
$TCAD$	Technology Computer Aided Design
TPI	Test Point Insertion
VLV	Very Low Voltage

Acknowledgements

This work was funded by Universiti Teknologi Petronas (UTP), Malaysia. I am very much indebt to UTP as well as its parent company PETRONAS for their financial support. I am grateful to my supervisor, Professor Mark Zwolinski for his guidance on conducting the research and showing how to clearly present my work. I also would like to acknowledge Professor Bashir Al-Hashimi, who served as the internal examiner of both my nine month progress report and my transfer viva. His feedback on these occasions as well as many other occasions has been very useful for my research progress. My appreciation to Professor Sandip Kundu, Professor Sudhakar Reddy and Professor David Flynn for fruitful discussions. I am thankful to my fellow friends: Arash, Biswajit, Karthik, Sawal, Urban, Bakar, Sankalp, Xiangqiang, Anton, Eduardo, Saqib, Aisha, Li Ke, Dirk, Razol, Chowdry and many others. My life in Southampton has been more meaningful with their friendship. To my parents: no words to say how grateful am I: May Allah bless you wherever you are. Finally to my beloved wife, Jerina Banu...thanks for your motivating words, your patience, your sacrifices and also for taking care of my little sweethearts Irfan and Imran.

Chapter 1

Introduction

One of the fastest growing segments of the electronics market is battery-powered devices. As the demand for portable and mobile real-time embedded systems increases, energy efficient design is becoming very important. Minimising the energy consumption of systems has become major design consideration.

Among the most promising power management policies is Multi Voltage Design (MVD). In modern System-on-Chip (SoC) design, not all parts of the design require similar performance objectives. Lowering the supply voltage on selected parts of the circuit helps reduce power significantly. The supply voltage can be assigned on fixed basis such as in Static Voltage Scaling(SVS) designs or in real-time according to the performance requirement such as Dynamic Voltage and Frequency Scaling (DVFS). Dynamic voltage adjustment methods such as in DVFS has gained its popularity due to its efficiency in power management.

In DVFS, the functional dependence of Voltage, Frequency and Energy is exploited. The clock frequency of a processor changes proportionally with the supply voltage, while the dynamic energy is proportional to the square of the processor's supply voltage.

Energy saving is achieved by adjusting dynamically the supply voltage and the clock frequency to the workload demand of the system. With the expense of increased execution time, the processor can be operated at lower speed. This enables the voltage to be lowered yielding a quadratic reduction in the energy consumption.

As for any electronics systems, it is necessary to ensure the system is tested and can function correctly to all design specification. The primary goal of testing any integrated circuits (ICs) is to ensure that the least number of defective ICs are shipped to customers. In order to achieve the targeted defective parts per million (DPM), an efficient test must be able to detect a very high percentage of the circuits' defects.

During normal operation, an MVD-enabled system can run at several different Voltage. In some cases such as in DVFS, an MVD-enabled system operates in several different Voltage/Frequency (V/F) pair settings. Therefore it is necessary to ensure that the system will function correctly at each possible voltage setting. Previous studies on low voltage testing has shown that while some faults cannot be observed at the nominal power supply voltage, they become apparent at lower supply voltage [2],[3], [4]. This raises a question about the validity of traditional test methodologies assuming a fixed/nominal power supply voltage and clock frequency. In a traditional method, the systems will be tested such that it will function correctly at one voltage and frequency setting. It is norm for these non-MVD systems to be tested at +-10% of the operating voltage, giving space for small variations.

Exhaustive test at all operating condition is a naive and straight forward method of testing MVD systems. However, cost of testing is counted by pence/second which involves the cost of Automated Test Equipment(ATE) and man-hours involved. Obviously the straight forward method will dramatically increase the cost of testing per chip. Therefore, an efficient method to test this multi voltage is essential to achieve better test quality.

The most commonly accepted test method is using stuck at fault model. This is used when the defect causes the output of a gate stuck at 1(HIGH) or 0(LOW). The limitation of this method is that, it cannot be used to detect fault that does not cause logical error. Shrinking transistor geometries have resulted in an increasing frequency of operation. With the ever increasing speed of ICs, violations of the performance specifications will affect the quality of the products. As a result, at-speed delay fault testing has become a necessity for high performance circuits.

This work mainly focus on delay fault testing. Delay fault can be modelled in a number of different ways. Gate delay fault model, path delay fault model and transition model are some of the commonly used models. The delay testing is normally applied using 2 vector test models, V1 and V2. V1 is used to initialise the target node and once it stabilises, the second vectors are injected and response are measured after the intended delay timing.

1.1 Testing for Multi Voltage Design

Even though Multi Voltage Design has been implemented in several contemporary embedded microprocessors such as Intel XScale [5] and Transmeta Crusoe [6], and in ARMs IEM [7], there are very limited publications addressing the issues with regards to testing these systems. Realising this, ARM Incorporation has initiated a comprehensive study to improve test quality for MVD. This part of our work has looked into dynamic fault in detail and reported its findings.

The problem we are addressing is finding the necessary voltage settings to detect all the known delay faults in a circuit or a design that can operate at more than one voltage. There are two main constraints in deciding the testing voltages. The first constraint is the fault coverage has to be maximised and ideally achieve 100%. The second constraint is the cost of the testing should be kept at the lowest possible.

The fault coverage is measured by taking the ratio of the number of fault detected to the number of total faults in the circuit under test. Fault coverage will depend on the known faults that we are trying to detect. The unknown and unmodelled faults will not have any impact on our test coverage matrix. In practical cases, 100% coverage might not be possible due to redundancy in the circuit architecture [8]. However the detectable physical defect range has to be maximised. This is applicable for defects that are modelled with resistive ranges such as resistive opens, resistive shorts and resistive bridging faults.

The cost of the testing will mainly depend on test data size and test time [9]. The test time will have direct impact on the test cost since the tester throughput is a function of time. The test data size will determine the size of memory required on the Automatic Test Equipment (ATE). Subsequently the memory capacity is one of the main factors that determines the cost of the ATE. For cases where the data size cannot fit in the tester's memory, multi-pass testing which requires memory update i.e. reloading of a subset of the test patterns during test application becomes necessary. This has to be done even though multi-pass test is time consuming [9]. Therefore it is necessary to keep the test data size and test time requirements as low as possible.

Khursheed et al. report that more than one voltage settings for testing multi-voltage design [10]. Their study on the impact of varying voltage on resistive bridging faults proves that testing at a single voltage is insufficient. However, their studies are limited to defects that cause stuck-at-fault only. On the other hand, exhaustive testing at all voltage settings will be too expensive due to time and resources such as ATE memory size. Finding a better trade-off is one of the objectives of this study.

In theory, once the appropriate voltage is known, testing process should be able to be commenced. However, components such as level shifters needs special attention due to their design characteristics. Level shifters is not a purely digital circuit. Furthermore the number of the level shifters in MVD circuit is large. We need to indicate if level shifters can be regarded as digital circuits for testing purposes.

1.2 Fault Modelling for Multi Voltage Design

As the design of VLSI circuits moves towards deep sub-micron (DSM) geometries, the process of testing becomes more complex. Fault simulation is an important step in generating efficient test patterns. The circuit is injected with known faults and then simulated using specific test patterns. The responses of the faulty circuits are compared with the response from fault-free simulation. This gives a list of faults that can be detected by each pattern. Fault simulation in multi-Vdd systems is further complicated by the need to test the circuit at more than a single voltage level to achieve higher fault coverage [11]. With increasing demand for power-aware consumer products, energy-efficient operation has become an important design objective. In multiple voltage design techniques, such as Dynamic Voltage Scaling, the system's voltage/frequency (V/F) setting may be dynamically varied according to the performance requirements.

Fault simulation is conventionally done at the gate level. Faults in digital circuits can be modelled as stuck-at, delay fault, bridging and open faults [8].

To speed up simulation, behavioural fault simulation has been looked at as a way to abstract gate level faulty behaviour, [12], [13], [14]. Since these simulations were conducted at Register Transfer Level (RTL), the type of fault is restricted to logical errors, as there is no information about the circuit structure.

SPICE-level simulation is commonly used to study the effect of faults at transistor level [11], [15],[16]. Fault simulations at circuit level will give more detailed results than gate level simulations [16]. Detailed power consumption and delays are some of the data that can be readily obtained from circuit level simulations. Even though circuit level simulation has the advantage of accuracy, it has the drawback of lower processing speed.

On the other hand, mixed-signal simulation using languages such as VHDL-AMS and Verilog-AMS tend to be faster than circuit level simulation. At this level of simulation, either some parts of the circuit or the whole circuit are described behaviorally. The main challenge in mixed-mode simulation is to produce meaningful results using the simplest behavioural models. There has been some prior work on behavioural fault simulation, [15], [17], but these works have concentrated on analogue circuits and systems. Figure 1.1 shows the relations between six different levels of hierarchy in terms of speed, accuracy and complexity. The behavioural/-functional models using high level languages such as C and Matlab will generally have the highest processing speed. However, this model will have a penalty in terms of accuracy. On the other hand, the device level models using Technology Computer Aided Design (TCAD) will be advantageous in terms of accuracy and complexity. However, the processing speed will be low. Previous behavioural models such as in [12], [13], [14] have used the RTL in VHDL or Verilog.

In this work, behavioural language VHDL-AMS has been used for mixed-signal fault simulations. At first, circuit level simulation using SPICE was used on basic cells such as NAND and NOR gates. Faults were injected and the responses were observed. The results were then used to generate the delay curve with respect to the value of fault resistance and supply voltage. The delay is then modelled at behavioural level using VHDL-AMS. The remaining parts of the circuit, which are fault free, are written at gate level with basic nominal delay values.

Finally, fault simulations at mixed signals are conducted to get the best trade-off between circuit level accuracy and gate level speed. Resistive open and resistive short defects are the two main classes of defects studied in this work.

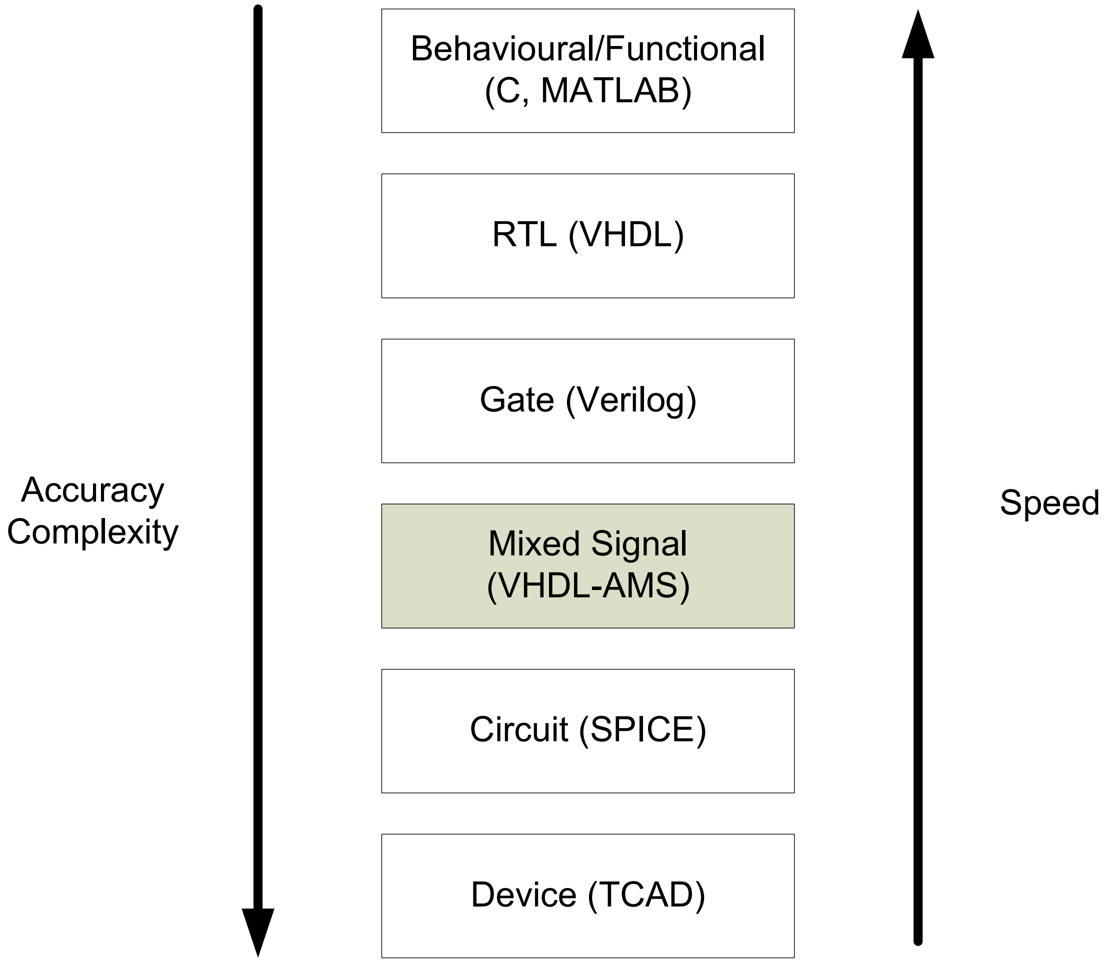


FIGURE 1.1: Speed versus accuracy in simulation

1.3 Contributions and Thesis Overview

Three main areas have been explored. The areas have been chosen to complement each other.

- the behaviour of defects at different operating conditions has been observed and studied in detail. To achieve the most in a short period, more pronounced faults such as resistive shorts and opens are used as case studies.

Appropriate voltage setting to achieve maximum fault coverage has been identified and suggested.

- the impact of including the level shifters in the digital Design-for-Testing (DFT) has been studied. Since MVD system works by varying the voltage, a defective level shifter can cause performance degradations as well as other critical faults.

- A novel method to develop behavioural fault model for MVD has been suggested. Fault simulation is one of the important step in generating quality test pattern. Since MVD circuits deal with more than one voltage, extensive fault simulation at transistor level will be time consuming. On the other hand, gate level fault simulations might not be able to expose the true behaviour of the fault. As a trade-off, a mixed-mode simulation will be something acceptable. The accuracy from the transistor level is used to model the defect and used at gate level for faster simulations.

This work presents an efficient testing methodology for dynamic fault detection. The findings have resulted in a better understanding of dynamic defect behaviours at different voltage levels. In the absence of our findings, there will be a significant increment in test application time to achieve required fault coverage. This is true if a naive approach of using all voltage levels is used to achieve required fault coverage. Our approach will shorten the overall test application time. In addition, the behavioural fault model can be used in complex design to expedite the fault simulation process.

1.3.1 Thesis Outline

The organisation of the thesis is as follows: Chapter 2 provides introduction to electronic testing with much emphasise on delay fault testing. Chapter 3 analyses the defect behaviour of bridging fault and open defect. Resistive defects which are more prone to cause delay faults have been explored in detail. Issues related to testing of low power design have been highlighted in Chapter 4. Very low voltage testing and level shifters are among the two studied in this chapter.

Chapter 5 is about delay fault testing for MVD. This chapter focuses on investigating the relationship between delay due to defects and the supply voltage. Detailed simulation results using simple and more complex circuit are presented in this chapter. Through extensive simulation using ST 0.12μm technology, we have looked into possibilities of using minimum set of voltages to ensure correct operation of the circuit.

In chapter 6, we have looked at method for testing level shifters. Contention mitigated level shifters were used to study the impact of defects on level shifters. This chapter investigates the need for testing methodology for level shifters in Multi Voltage Design environment. The impact of having bridging fault in the level shifter design and how it amplifies the fault effect were shown. Two different test conditions - PASSIVE and ACTIVE - were investigated and detailed results have been presented.

Chapter 7 presents work on fault modelling and simulation using mixed-mode language such as VHDL-AMS. Delay due to different defects are modelled at transistor level and used in behavioural fault simulations. Finally, Chapter 8 concludes this thesis by summarising its most important contributions. It also provides recommendation for future work.

Three appendices are also included in the thesis. Appendix A gives results and discussion for delay fault testing simulations conducted using ST 0.35μm technology. Appendix B gives the spice netlist for the multiplier circuit used in Chapter 5. Appendix C gives the VHDL-AMS code used for one of the simulation in Chapter 7.

List of Publication

The research work in this thesis were presented and published in official proceedings of rigorously refereed conferences through the following research papers:

- Zain Ali, N. B., Zwolinski, M., Al-Hashimi, B. M. and Harrod, P. ***"Dynamic Voltage Scaling Aware Delay Fault Testing"***. In Proceedings of 11th IEEE European Test Symposium, Southampton, May 2006.
- Zain Ali, N. B., Zwolinski, M., Al-Hashimi, B. M, ***"Testing of Level Shifters in Multiple Voltage Designs"***, In Proceedings 14th IEEE International Conference on Electronics, Circuits and Systems, Morocco, December 2007
- Zain Ali, N. B., Zwolinski, M. and Ahmadi, A. ***"Delay Fault Modelling/Simulation using VHDL-AMS in Multi-Vdd Systems"***, In 26th IEEE International Conference on Microelectronics, Nis, Serbia,May 2008

The following informal presentation were also given:

- Zain Ali, N.B. , Zwolinski, M, ***"Dynamic Voltage Scale Aware Delay fault Testing"***, 13th International Test Synthesis Workshop, Santa Barbara, California, USA, April 2006.
- Zain Ali, N.B. , Zwolinski,M. ***"Testing of Level Shifters in DVS Enabled Chip"***, Postgraduate Workshop on Microelectronics and Embedded Systems, Birmingham, UK, October 2006
- Zain Ali, N.B. , "Delay Fault Testing and Modelling in Multi Voltage Design Systems", 11th Annual ACM/SIGDA Ph.D. Forum at DAC, Anaheim, California, USA, June 2008

Chapter 2

Electronic Testing

Testing of VLSI circuits is an essential technology to realise dependable systems. In this chapter, we will discuss electronic testing methodology as has been addressed by the research community. The importance of testing is introduced in Section 2.1. Fault models and how it can help in testing process is introduced in Section 2.2. This is followed by Automatic Test Pattern Generation(ATPG) in Section 2.3 and Fault Simulation in Section 2.4. Section 2.5 discusses issues in relation to testability of the digital circuit. Finally, detail discussion on delay fault testing is given in Section 2.6.

2.1 Introduction to Digital Testing

The goal of testing is to determine if a manufactured circuit contains any defects. Testability of a circuit is measured in terms of controllability and observability of a defect. Controllability is a metric used to measure the difficulty in driving a node of a circuit to a specific value. Similarly, observability is a metric used to measure the difficulty in propagating the value on the node to a primary output of a circuit.

In a more general term, testability can be reflected as the ability of detect failures causing malfunctioning of the circuits. With the increasing design complexity and reduced error margins in semiconductor manufacturing, testability of the circuits becomes one of the major requirements for circuit designers. Rapidly shrinking feature sizes widen the spectrum of new types of defects, and increasing gate counts have increased the number of locations where such defects can occur.

The International Roadmap for Semiconductor(ITRS) reported that even though significant progress continues in the reduction of manufacturing test cost, much work remains ahead. Even though the cost of Automatic Test Equipments (ATE) has dropped around 40% recently, the demands generated by increasing design complexity has quickly offset the improvement [18].

At the same time, while tester accuracy for timing-signal resolution has improved at a rate of 12% per year, semiconductor speeds have increased at 30% per year [19]. As a result, the “National Technology Road-Map for Semiconductors” has taken the view that testing is one of the six ”Grand Challenges for the Semiconductor Industry” [20].

Testing a system is an exercise where the resulting response is analysed to ascertain whether it has behaved correctly. In the case of the incorrect responses, the second goal of testing is to diagnose or locate the cause of the misbehaviour [8], [21], [22]. To conduct a diagnosis, the internal structure has to be understood very well.

In simple terms, the manufacturer of a product requires the product to function without any unwanted behaviour for the period for which the product was designed to last. This is important since the cost to repair the product if it is reported to be malfunctioning is high. Therefore, it is necessary that all the testing is performed before the product is shipped to the end user.

In critical systems such as safety related and aviation, another step to ensure the function of the system even with the presence of fault is necessary. This is called fault tolerance [23], [24], [25]. The most basic method of fault tolerance is having redundancy in the circuit. The redundancy can be in terms of space, time or information [26].

In electronic testing, each component of a system has to be tested for fabrication defects. Ideally, the defects have to be detected at the earliest point possible. Before any component is mounted on a board, it has to be tested extensively since board level testing is much more expensive and complex [27]. However, it is important to note that most industries still conduct board level test such as boundary-scan at systems level to further ensure that the system is fault free [28].

Faults detected at the systems level, i.e in an assembled board, will cost 10 times more than the cost of detecting faulty chips [29]. The cost increases dramatically since it is more complicated and expensive to diagnose and locate the fault. Board level testing must not only account for the functionality and performance of all the devices placed on the board, but it must also account for how the devices are assembled on to the board and how the devices interact with one another [30].

2.1.1 Functional and Structural Testing

The general spectrum of test process can be divided into two main categories

- Functional Testing - does the system work correctly ?
- Structural Testing - does the system contain a fault ?

Functional testing exercises the chip's intended functionality. It verifies the input-to-output behaviour of the circuit. This method is advantageous for testing off the shelf components and cores. Example of such tests are memory test by reading and writing as well as communication interface tests.

As the functional test moves from individual IC level to assembled board level, the total number of defects that can be detected will decrease significantly. This is resulted from reduced controllability and observability which potentially affects the resulting product quality [31].

In addition, detailed knowledge of the functionality of the circuit is required to derive the test set [32]. This will be a difficult and time consuming task if the test engineer is not involved in the design process.

On the other hand, a structural test assumes that if the circuit has been manufactured correctly it will function accordingly. It tests for any mismatch between the intended structure of the circuit and the manufactured circuit [33].

Structural test involves strategies to verify the individual elements of the design. These elements include logic gates, transistors, and interconnects. Such tests are called 'structural' as the tests depend on the specific structures of the design. Since the internal structure of the circuit is known, this class of testing facilitates diagnostics of failing devices. The work in this thesis is based on structural test.

2.1.2 Manufacturing Defects

Hardware defect can be defined as an unintended difference between the implemented hardware and the intended design. The defects can be caused by design errors, fabrication errors, fabrication defects and physical failures [8]. Examples of design errors are incomplete or inconsistence specifications, incorrect mappings between different levels of designs and violation of design rules. Wrong components, incorrect wiring and shorts caused by improper soldering are examples of fabrication errors.

While design errors and fabrication errors are directly attributable to human errors, fabrication defects are due to imperfect manufacturing processes [34], [35].

Among the common fabrication defects are shorts, opens, improper doping profiles, mask alignment errors and poor encapsulation.

Defects due to component wear-out and/or environmental factors during the lifetime of a system are categorised as physical failures. Overstress, electromigration, corrosion and cosmic radiation are examples of conditions that can cause physical failures [8].

A fault is an abstracted representation of defect. The term *physical fault* are commonly used for defects due to fabrication errors, fabrication defects and physical failures. These faults need to be detected through testing. Modelling this fault in an abstract level will assist the testing process. The next section describes Fault Models.

2.2 Fault Models

Fault model serves as an abstraction of silicon defects to aid test generation and fault simulation. The subject of test generation and fault simulation are discussed in Sections 2.3 and Section 2.4 respectively.

By modelling the physical faults as fault models, the complexity of fault analysis is greatly reduced. For example, when a defect is modelled as logical fault, the analysis of the defect can be explained in logical terms. In addition, many physical faults can be modelled by the same fault model thus reducing the number of individual defects that have to be considered.

Faults can be modelled at switch-level, gate-level as well as Register Transfer Level (RTL). The gate-level fault model is widely accepted as the best compromise between abstraction and the ability to represent most of the defects in the device-under-test (DUT)[36].

2.2.1 Single Stuck-at Fault Model (SSFM)

One of the most mature test strategies is to apply a logical test based on the Single Stuck-at Fault Model (SSFM). In the SSFM model, a wire or node in a system is considered to retain a logical value ("0" or "1") regardless of the value driving it. Each node can have two types of fault: stuck-at-1 and stuck-at-0, commonly referred as *s-a-1* and *s-a-0* respectively. A node will produce a logical error whenever the driving line assumes the opposite value ("1" or "0" respectively).

The assumptions in SSFM are that only one line in the circuit is faulty at a time, the fault is permanent and the fault can be at an input or output of a gate [37].

Stuck-at fault testing is static voltage based testing [38]. Test patterns for this test are usually applied at lower than normal operating speed using a scan chain. Constraint in the power and constraint on the scan chain routing are the main reasons for the lower speed testing [39].

2.2.2 Limitation of SSFM and its alternatives

Even though stuck-at fault is the *de facto* test technique, it has been reported that single stuck-at fault model is not a realistic description of faults.

In the 1999 ITRS, it has been reported that SSFM covers only around 70% of the possible manufacturing defects in CMOS circuits. That leaves 30% of the possible defects potentially undetected. The test quality will be strongly dependent on the number of un-targeted fault using the SSFM model. Test quality is measured by taking the probability of all detectable faults detected by the fault model. In the above case, the test quality cannot exceed 70% since 30% of the faults will never be detected by SSFM.

The test quality can be improved by using more accurate fault models in combination with stuck-at fault test methods. In general, there are four classes of test methods [38]:

- Static voltage based tests such as stuck-at fault model
- Dynamic voltage based tests, such as delay fault testing, "at-speed" testing and functional testing
- Static current based tests, such as conventional I_{DDQ}.
- Dynamic current based tests or transient testing I_{DDT}.

A well accepted test complement for SSFM test technique is delay fault testing [40],[41],[42]. Delay fault testing which is also known as at-speed testing can detect a fault that might not cause logical error but will result in performance degradation. As geometries continue to shrink, manufacturing tests based on the static stuck-at fault models are becoming less effective in detecting defects which are typically resistive opens and shorts. Resistive shorts and opens can cause logical errors as well as performance degradation. The SSFM can only be used to detect logical error whereas timing related performance degradation needs to be detected by use of delay fault testing. Delay fault models will be discussed in detail in Section 2.6.

In I_{DDQ} testing [39], the leakage current of the power supply is observed between clock edges when there is no switching activity. Short circuit defects cause abnormal current flow and can be detected by monitoring the *quiescent* supply current of the device, which is normally due to the leakage current. Defects will introduce abnormal currents which are typically one or more orders of magnitude larger than the fault free leakage current.

In CMOS technologies the leakage current used to be very low and an elevated current indicated the presence of defects. Besides resistive short, current based tests are also capable of detecting a wide range of other defects such as gate-oxide shorts and stuck opens. Furthermore I_{DDQ} testing is capable of catching defects, which do not cause logic faults but nevertheless make the device unacceptable for the customer or form a reliability risk, as for example excessive power consumption owing to shorts between VDD and VSS line. I_{DDQ} test is still accepted as high quality supplement test for its effectiveness in the detection of bridging and gate-oxide defects [43]. However measurement of extremely low currents requires a large settling time, which results in a slow measurement process [44].

I_{DDT} testing is the counterpart of I_{DDQ} testing. I_{DDT} testing method is built on the observation that a fault free circuit will draw a significantly large amount of current while changing from one state to another. The transient current is used as criterion to differentiate between good and defective circuits [45]. While the circuit is actively switching, the measurement of the peak value of the transient current and shape/duration of the transient pulse are taken. Any large mismatch from the pre-recorded value signifies a faulty circuit.

With the increase of sub-threshold leakage in smaller geometry designs, current based testing effectiveness will be hampered [38]. As CMOS technology moves toward smaller features, the mean I_{DDQ} current for fault free circuit will increase dramatically. This will result in a smaller difference between mean I_{DDQ} currents of fault free and faulty circuits. Once the feature size moves beyond 130 nm, I_{DDQ} testing will be ineffective as it will be difficult to distinguish a defective circuit from a fault free circuit [46].

2.3 Automatic Test Pattern Generation

In the previous sections, it has been shown that a defect in a manufactured circuit can be modelled as a fault. Given a circuit, we then need to determine if the circuit contains any faults. Automatic Test Pattern Generation (ATPG) is the process of generating patterns or input sequences to test the targeted faults in a circuit [47]. The patterns are also known as test vectors and the targeted faults are kept in a fault list.

The two main phases in ATPG are fault activation and fault propagation. During fault activation, ATPG generates an appropriate subset of all input combinations at primary inputs, such that a desired percentage of faults is activated and observed at the primary outputs. The output signal of the circuit changes from the value expected from a fault free circuit. Next, in fault propagation step, fault propagated to the primary outputs will allow the fault to be detected. Among the commonly used ATPG algorithms are the Sensitive path algorithm, D Algorithm and PODEM [48].

The effectiveness of ATPG is measured by the number of detected faults and the number of generated patterns. A detected fault is a fault for which a valid test vector has been generated. Fault coverage is a metric used to measure the effectiveness of the generated test patterns. Equation 2.1 shows how the fault coverage is calculated.

$$Fault\ Coverage = 100 * \frac{number\ of\ detected\ faults}{total\ number\ of\ faults\ in\ the\ circuit\ undertest} \tag{2.1}$$

Fault coverage can be improved by improving design such that it has improved observability and controllability.

The second metric to measure the effectiveness of the ATPG algorithm is the number of generated pattern within a specific time. The number of generated patterns has an impact on the test application time. Test application time is defined as the time taken to apply a certain number of test vectors. An increase in test application time will have a detrimental effect on test quality [4], [2]. Issues in relation to test application time and test quality is discussed in Chapter 3.

2.4 Fault Simulation

Fault simulation consists of simulating a circuit in the presence of faults [8]. The goal of fault simulation is to determine the list of faults in a device-under-test (DUT) that are detected by a specific test vector. The general procedure is to simulate the good and faulty circuits and determine if they produce different outputs.

As a large portion of this work involves fault insertion and fault simulations, we will briefly explain the issues. Before presenting details on fault simulations, some fundamental definitions are given [8].

Definition of Fault Equivalence: *Two faults f_i and f_j are equivalent if there is no test that will distinguish between them.*

Figure 2.1 shows an example of fault equivalence. When input A of the NAND gate is stuck-at-0 and input B is stuck-at-0 it is equivalent to output C stuck-at-1. This is true since only the input combination of AB=11 can detect this. Therefore, the fault condition of A is stuck-at-0 and B is stuck-at-0 can be removed from the fault list. In general any of the equivalent faults can be removed.

Definition of Dominant Fault: *A fault f_i dominates f_j if every test that detects f_i also detects f_j.*

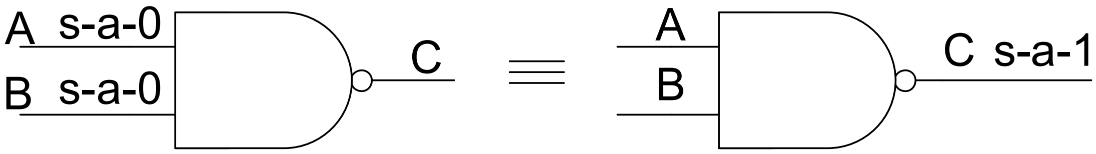


Figure 2.1: Example of Fault Equivalent

Consider a two input NAND gate. The output C stuck-at-0 can be detected by the input combination AB=00, or 10, or 01. Similarly, input A stuck-at-1 can be detected by input combination of AB=10 and for fault for input B stuck-at-1 can only be by AB=10. Therefore, fault A stuck-at-0 and fault B stuck-at-1 dominate fault C stuck-at-0. As a result, fault C stuck-at-0 can be removed from the fault list.

Definition of Fault Collapsing : *The process of reducing the fault set in a fault list by removing equivalent and dominated fault.*

Definition of Fault Insertion : *Selecting a subset of faults to be simulated and creating the data structures to indicate presence of faults.* Fault insertion is also known as fault injection.

Definition of Simulation Based Fault Injection : *Done at the pre-manufacturing design stage. Typically, the circuit is described either in a hardware description language (HDL) or at the transistor level. Fault injection is done by perturbing the fault free descriptions so that the resulting system emulates the faulty circuit.*

Fault simulation algorithms consist of five specific tasks:

- The good circuit is simulated and responses are recorded. The responses from the good circuit are used to compare responses from faulty circuit.
- Fault specification: First, the fault list is generated. Then, Fault Collapsing is used to reduce the number of faults in the fault list.
- Fault Insertion.

- Fault-effect generation and propagation. Fault effects are generated by fault insertion and propagated to primary outputs.

- Fault detection and discarding. All the detected faults are discarded from the set of fault list to indicate that the fault can be detected by the specific fault simulation. Remaining undetected faults are targeted for the next round of fault simulation.

Among the uses of the fault simulations is the measurement of the effectiveness of a sequence of test vectors in detecting manufacturing faults in integrated circuits.

Given a test set T, by conducting a fault simulation, we can observe how many of the faults can be detected by T. A fault is considered detected if the response of the faulty circuit is different from the response from the fault free circuit.

A good set T should give a good fault coverage. In the context of Fault Simulation, Fault Coverage is defined as the ratio of the number of faults detected to the total number of faults simulated [49].

A low fault coverage indicates that there are parts of the circuits that are not being tested by the vector set. In order to increase the fault coverage to an acceptable level, additional test vectors are written - targeting the untested area of the circuit. However, there will be areas which are inherently untestable [50]. The main reason for the area to be inherently undetected is the redundancy structure of the circuit. If testability needs to be increased, the circuit might need to be redesigned.

Fault simulators based on gate level models [51],[52], [53] can only model static voltage based faults such as stuck-at-0 and stuck-at-1. However, CMOS circuits also exhibit dynamic properties that cannot be modelled at the gate level. Only fault simulations at transistor or switch-level can incorporate these dynamic behaviours such as bidirectionality of signal flow, dynamic charge storage, charge sharing and ratioed circuits [49].

For complex VLSI circuits, serial fault simulations where the fault free circuits and faulty circuits are simulated separately would be computationally very expensive. There is a number of methods used to reduce the computation. Parallel fault simulation, deductive fault simulation and concurrent fault simulation are amongst the well accepted fault simulation techniques. These techniques are discussed in detail in [8].

Simulation based fault injection have been used throughout this study. The circuit is described in transistor level and faults were injected in chapter 5 and chapter 6. In chapter 7 fault simulations were conducted in transistor level as well as in behavioural level using VHDL-AMS.

2.5 Testability of Digital Circuit

Design for Test which is also known as "Design for Testability" or "DFT" is a name for design techniques that add certain testability features to an electronic hardware product design. Testability can be measured in terms of ease and speed with which a test program with high fault coverage can be developed. Observability and Controllability are two metrics used to measure testability. DFT encompasses a broad range of issues which include product performance, circuit design time, wafer yield impact, test development time, fault coverage, product quality and finally, the overall time to market the new designs.

There are two main categories of DFT techniques: ad-hoc techniques and structured techniques. Among the ad-hoc techniques are Partitioning, Degating and Test Point Insertion [29]. Partitioning use a "Divide and Conquer" rule in which the system is physically divided into multiple chips or boards.

In the Degating technique, the similar "Divide and Conquer" as for Partitioning is used. However, rather than physically dividing the integrated blocks, gates are used to separate the blocks. Test point insertion techniques use additional lines to control and observe internal points.

The structured DFT techniques comprise of Scan Based DFT and Built-In Self Test (BIST)[52], [48].

The general technique in scan based DFT is to make all or some state variables directly controllable and observable. Without the scan based DFT, some faults are untestable since certain states cannot be reached. In scan based DFT, Scan-in, scan-out (SISO) principles provide direct control to all inputs of the combinational logic. It also provides mechanism to control and observe the state variables by connecting all the flip-flop together as shift registers.

Figure 2.2 shows an n-bit scan register. The scan registers are normal flip-flop with an extra control input T. The control input will determine the mode of operating: test mode or normal mode. When T=0, the scan register is in normal mode and the register is loaded with functional inputs through input port D_1 to D_n. When T=1, the scan register is in test mode where the data are shifted into the registers through primary input port S_{in} and later shifted out through primary output port S_{out}

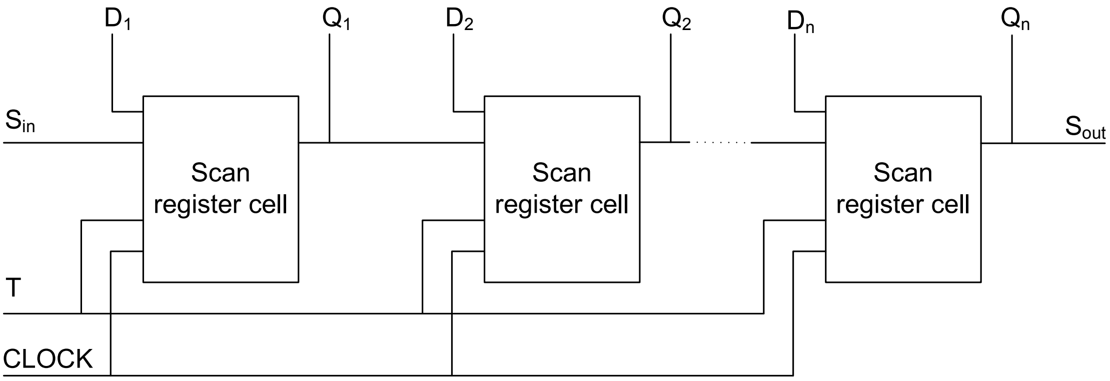


FIGURE 2.2: A scan register

Figure 2.3 shows how a scan register can be used in the scan registers shown in Figure 2.2. Figure 2.3 is showing a general scan based DFT whereby the test vectors are provided externally while the system is in test mode. In normal mode, the multiplexer (MUX) propagates the functional value provided by C1. When the system is in test mode, the value provided by the scan flip-flop is fed to C2. The Scan Register Cell is part of overall scan register or chain of registers. The cell provides test point functionality for the whole design.

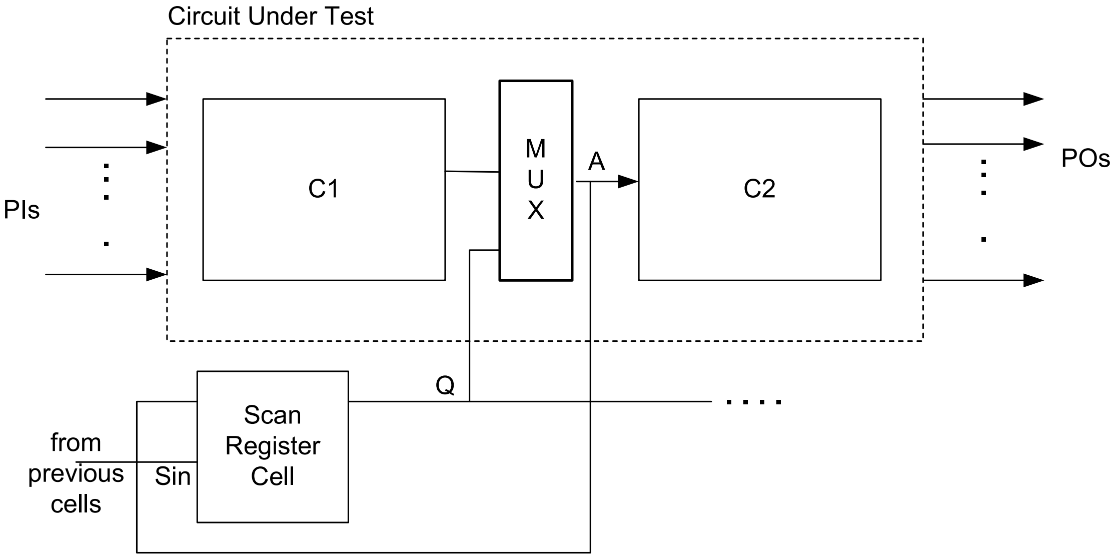


FIGURE 2.3: CUT with DFT employing scan register cell

ATPG algorithms combined with Scan-based DFT methodology automate the generation of test patterns [54]. This combination has the advantage in terms of high efficiency and effectiveness in generating a test set by targeting different fault models, such as SSFM, delay fault model and I_{DDQ} testing.

In BIST based DFT, the goal is to add devices to a design that will allow it to test itself. The whole process of testing which includes the test generation and test application is accomplished through built-in hardware features. By having the test structures within the circuits, the testability can be enhanced and at the same time reduce the cost of test equipment [37], [55], [56]

BIST solution can be implemented either as an off-line or an on-line scheme. In off-line testing, the circuit is placed in special test mode in which the circuit does not carry out normal operation. In on-line BIST, the tests are performed during normal operation of the circuit.

A DFT or test method should be selected to improve the product quality with minimal increase in cost due to area overhead and yield loss. The product quality can be measured by the ratio of total number of non-defective devices to the total shipped devices [54].

Often it is necessary to find the best trade-off between DFT methods and increase in cost. There are also methods that combine ad-hoc techniques such as Test Point Insertion with structured techniques such as scan based design [57].

As we have seen for BIST testing, an electronic test can be conducted either off-line or on-line. In off-line testing, the Circuit Under Test (CUT) is taken off-line. It means that its normal operation is suspended. Then test pattern or test vectors are applied to its inputs and responses are observed at the outputs. These responses are compared to the expected fault-free responses. Any mismatches indicate faulty conditions. On the other hand, the circuit is tested while running its normal operation in on-line testing schemes.

The test vectors or test patterns can be provided either externally through Automated Test Equipment(ATE) such as in scan based DFT or internally by dedicated embedded hardware within the structure of the circuit such as BIST method.

2.6 Delay Fault Models

One method to achieve higher performance of a system is by maximising the frequency of the system clock. By increasing the clock frequency, the number of operations in a given time can be increased. The maximum allowable clock is limited by the two types of delay of the combinational logic block between flip-flops: propagation delay and switching delay. Propagation or interconnect delay is the time a transition takes to travel between gates. The delay will depend on transmission line effects i.e. distributed R,L,C parameters, length and loading of routing paths.

The circuit will also have a switching or inertial delay between an input change and the output change. This interval depends on input capacitance, device or transistor characteristics and output capacitance [37]. Other factors which are generally referred to as second order impact will also have their effects on switching delay. These are rise time, fall time as well as states of other driving inputs. The states of other input will determine the driving strength of the circuit. For an example, a 2 input NAND gate will have higher driving strength when the inputs are 11 compared to when the inputs are 10 or 01. The driving strength of the input will have an impact on the capability in term of load conditions which will be in terms of the number of fan-outs [58].

Failures that cause logic circuits to malfunction at the desired system clock rate thus violates timing specifications are modelled as delay faults. These faults cannot be detected by stuck-at fault models and can only be detected by at-speed testing.

Previous researches have shown that inclusion of delay fault testing is critical to achieve the desired product quality. For an example Kee et al. [59] reported that there will be more than 1,400 slow units for every million manufactured without delay fault testing. These 1,400 units themselves would exceed the total quality requirement for most manufacturers, who often target 500 or fewer total defects per million units shipped (DPM). Gatej et al. [60] have shown that a test program without at-speed test will result in an escape rate of up to 3%. Their analysis was based on a microprocessor designed using 180nm technology.

Stanford University's Murphy and ELF35 experiments have shown that 3 out of 116 defective parts escaped when tested at a slower than functional speed at normal operating voltage [61]. These chips were built using 0.7 and 0.35 micron technology.

2.6.1 Causes of Delay Faults

Delay faults can be caused by physical defects and/or process variations [8]. Among the physical defects that can cause delay faults are low threshold path conductance, narrow interconnect lines, threshold voltage shifts, certain CMOS opens, resistive vias, IR drop on power supply lines and crosstalk. Process variations such as mask misalignment and line registration error can cause devices to switch at a speed lower than the specification. These variations are caused by imperfection in both processing and mask [62], [63]. These imperfections will not make the chip functionally defective. However, the chip may perform slower than the specified speed. These imperfections need to be detected by means of delay fault testing [64].

Chang and McCluskey [40] have looked at timing failures at transistor level that could lead to delay faults. They have shown that transmission gate opens can cause degraded signals at the circuit under test. Other faults reported by Chang and McCluskey are threshold voltage shifts, diminished-drive gates, gate oxide shorts, metal shorts, defective interconnect buffers, high resistance interconnect via defects as well as tunnelling opens. Defects that are more dominant at deep sub micron were reported by Moore et al. [42]. Moore looked at deep sub micron defects such as infinite opens, resistive opens, resistive and zero resistive bridges in combination with signal integrity effects such as crosstalk, extended propagation delays and power rail coupling. Crosstalk has been reported to cause slow-down as well as speed-up in CMOS circuits.

2.6.2 Delay Fault Testing Methodology

A suggested method for detecting delay faults is by using a two pattern test, {V1 and V2}. Figure 2.4 shows the hardware model and the clock timings for the delay test. At time t0, an initializing input vector V1 is applied. After the circuit has stabilised under input V1, the second vector, V2 is applied at time t1. The outputs are sampled at t2. (t2-t1) is the allowable time interval between the input and output clock and is called the rated clock interval, Tc. During normal operations, the input clock C1 and the output clock C2 have the same clock period which is the rated clock. This period should be greater than the maximum propagation delay. Delay fault testing is commonly applied using any of the following three techniques. These are Launch-On-Shift (LOS), Enhanced Scan and Launch-On-Capture(LOC). The main differences between these techniques are the ways the vectors are generated and applied.

2.6.2.1 Launch-On-Shift (LOS)

Launch on Shift(LOS) is also known as Launch from Shift, Skewed Load Test and Scan Shifting [65]. In LOS, the shifted first vector is used as the second vector. As the vectors are fed in serially, the second vectors bit are arranged such that the next vector is just one bit shifted from the first vector. Figure 2.5 shows the waveform of the system clock and corresponding scan enable signal. The first two clocks (Launch 0 and Launch 1) are used to launch the initialise pattern and propagate pattern respectively. The third clock is used to capture. The time between the second launch and the capture is most critical since the capture has to change at the at-speed rate of testing. This implies that the scan enable has to switch exactly between the 2 at-speed system clocks. Due to clock skew problems, this is impractical. Another main disadvantage of this method is that the ability to apply a test is limited by order-dependency of the serial scan path.

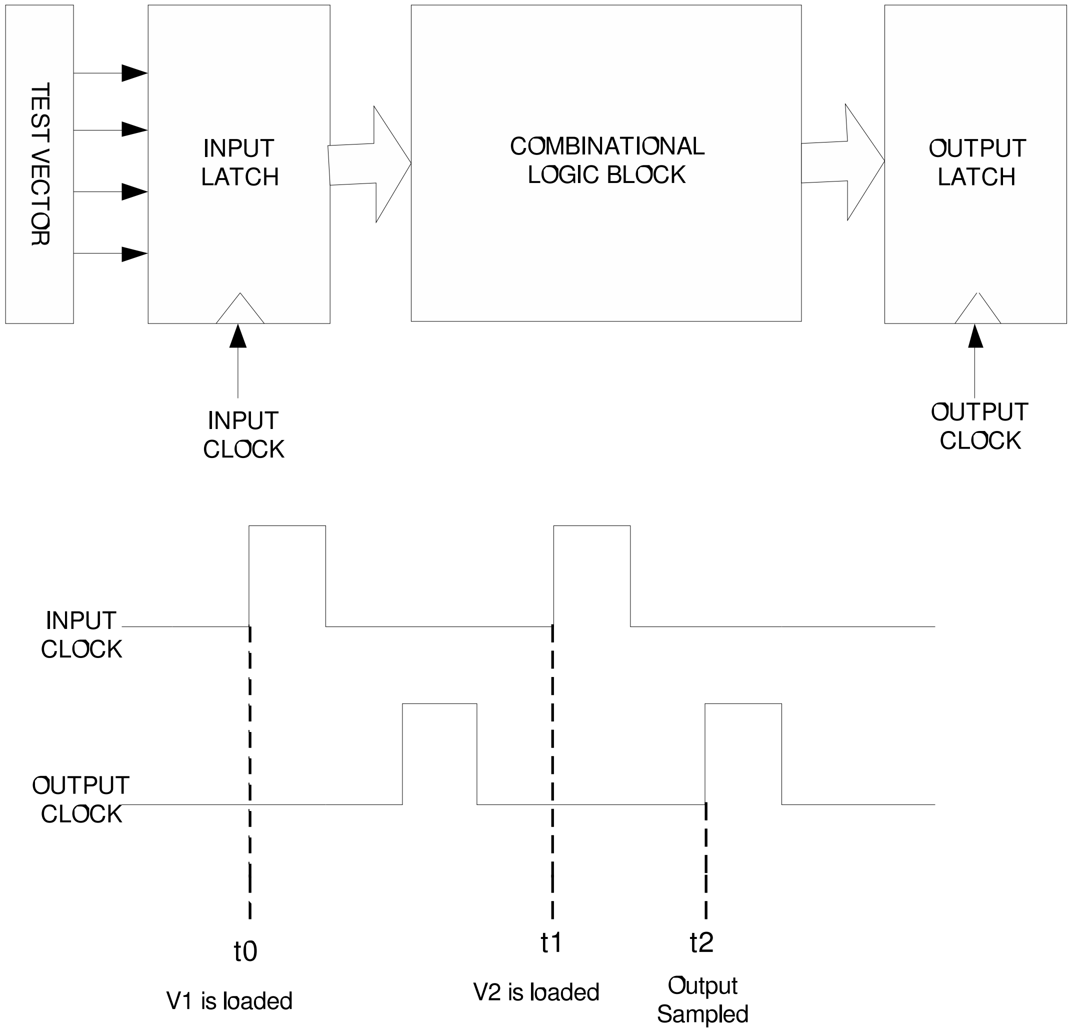


Figure 2.4: Hardware model and clock timing

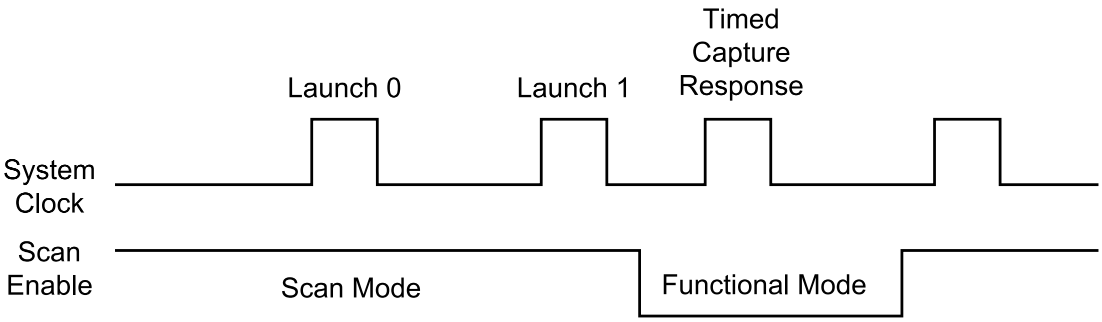


Figure 2.5: Waveform for Launch on Shift method

2.6.2.2 Enhanced Scan

Enhanced scan which is also known as Buffering the Flip-flops is designed to overcome the problem of order-dependency in the LOS method. The advantage of this method is that we can achieve a higher fault coverage since both vectors are controllable. In this method, both the initialise and propagate vectors are shifted in during the shift process to the scan flops. Special scan flops which can hold 2 values at a time are required for this method. Thus, the area overhead for this method is large.

2.6.2.3 Launch-On-Capture (LOC)

Launch from Capture, Launch-Off-Capture, Double Capture Clock and Functional Justification are among the different names used for the Launch-On-Capture (LOC) method. This method only uses one vector during the shift cycle. The first vector V1 is applied and the resulting functional response is used as the second vector V2. Figure 2.6 shows the waveform of the clock and the corresponding scan enable signal. The main difference between this method and the LOS is the critical time between Capture 1 and Capture 2 which is inside the functional mode. This avoids the requirement for the scan enable to switch between two clock cycles. The advantage of this approach is that it does not require the scan enable signal to operate at full speed. In addition, it has a lower area overhead than Enhanced Scan method since no special scan flops are required.

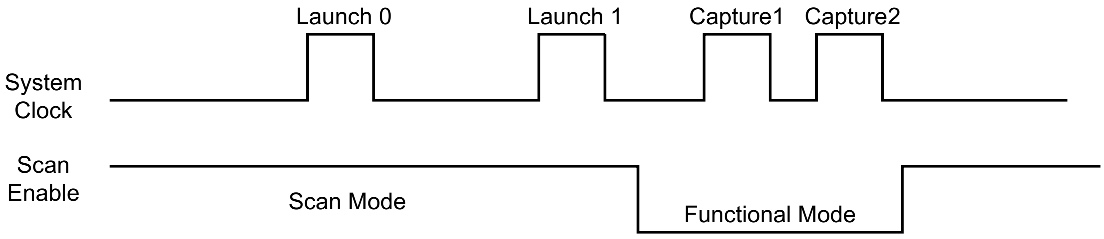


Figure 2.6: Waveform for Launch on Capture method

2.6.3 Classification of Delay Fault Models

Models for delay faults are classified into local and global delay fault models [66]. Figure 2.7 shows the taxonomy of delay faults.

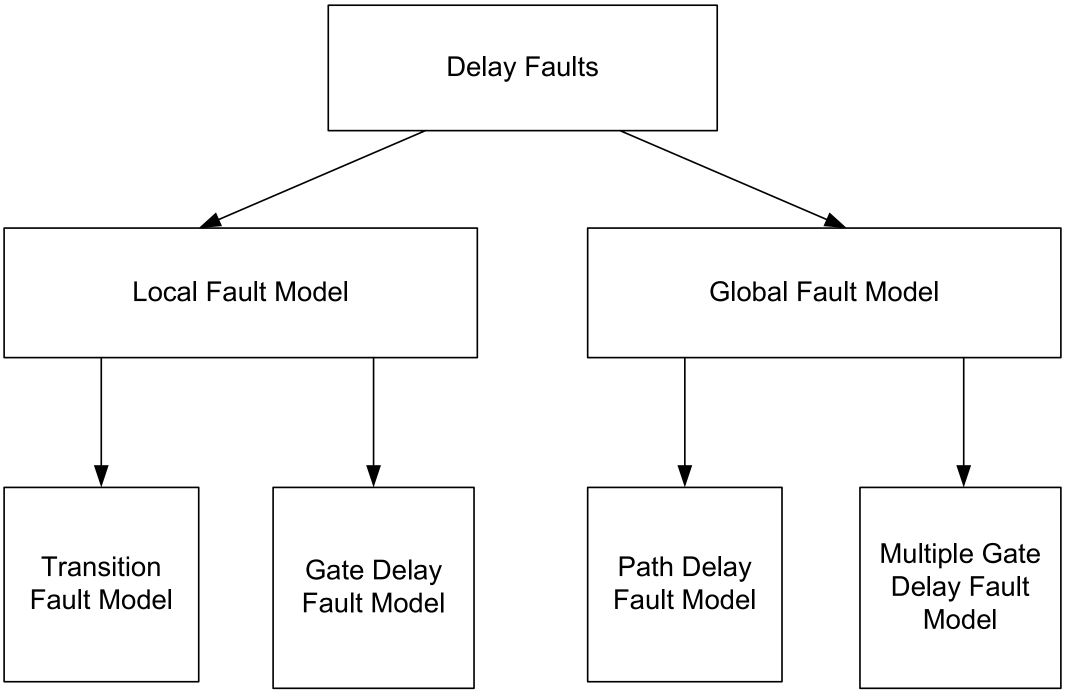


Figure 2.7: Taxanomy of delay faults

The three classical fault models to represent delay defects are transition fault, gate delay fault and path delay fault [41]. Multiple gate delay fault models are a combination of more than one gate delay fault model. Other fault models such as line delay fault model and segment delay fault model are derivation of the classical models.

2.6.3.1 Transition Fault Models

The transition fault model is considered as a logical model for a defect that delays the rising or falling transitions at inputs and outputs of a logic gate. The extra delay caused by the fault is assumed large enough to prevent the transition from reaching any primary output at the time of observation. This implies that a delay fault can be observed independently whether the transition propagates through a long or a short path to any primary output. The transition fault model can also be used as a logic model for transistor stuck-open faults in CMOS circuits [67]. The CMOS transistor stuck-open fault can be treated as a fault that either suppresses or delays the occurrence of certain transitions. The extra delay caused by stuck-open will depend on the electrical characteristic of the defective components [68], [69].

The two kinds of transition fault models are slow-to-rise and slow-to-fall. The slow-to-rise defect will cause the circuit to behave as stuck-at-0 value temporarily. Similarly, the slow-to-fall will behave like stuck-at-1 temporarily.

The main advantage of the transition fault model is that the number of faults in the circuit is linear in terms of the number of gates. In addition to that, the stuck-at-fault test generation and fault simulation tools design can be easily modified for handling transition faults [70]. This is done by adjusting the scan based design for SSFM test.

However, the expectation that the delay fault is large enough for the effect to propagate through any path passing through the fault site might not be realistic because short paths may have a large slack time. Slack time is the difference between task deadline and actual time taken to complete the task. Deadline is a given time to complete a task. Slack time is considered as a result of overperformance [71]. Clearly, slack time is a special case of idle time

Another assumption made in transition fault is the delay fault only affects one gate in the circuit.

2.6.3.2 Gate Delay Fault Models

The gate delay fault model is a quantitative model for delay fault since it takes into account the circuit delay. In this model, two assumptions are made. First, the delays through the logic gates are known. Secondly, the sizes and locations of the likely delay faults are also known.

An added delay of certain magnitude in the propagation of a rising or falling transition from the gate input to output is considered as a fault. In this model, only a delay exceeding a specified delay size can be detected. Methods for computing the smallest delay fault size guaranteed to be detected have been reported in the literature [72].

To determine the ability of a test to detect a gate delay defect is necessary to specify the delay size of the fault. The effectiveness of a test set is limited by the smallest fault size it can detect. A test T, might miss a defect with gate delay size less than the guaranteed size.

2.6.3.3 Path Delay Fault Models

The path delay fault (PDF) model has received greater attention around the mid 1990s [37],[41],[63]. Any path with a delay exceeding the clock interval is said to have a path delay fault. A test set for the path delay fault model can detect both localized and distributed delay defects. For each physical path in a circuit, there are two PDFs, at the rising and falling transitions. The PDF is delay independent since the clock frequency will take into account the switching delay and propagation delay. This feature makes it advantageous to other delay fault model such as gate delay fault model.

A major limitation of this model is that the number of possible path is an exponential function of the size of the circuit [73]. It is impractical to do exhaustive testing since the total number of pattern-pairs required will be $(2^n)(2^{n-1})$, which is in the order of 2^{2n}, for a circuit having n inputs.

There are many techniques used to reduce the number of paths that must be tested in the path delay fault model. The classic way is to test the paths with maximum delays in a circuit. This path is also known as the critical or longest path. However, due to circuit optimisation, the distribution of path delays are compressed. This will results in so many paths close to the longest or maximum delay [74]. As a result, a group of longest paths must be selected and tested.

A simple approach is to choose the path with a delay that exceeds a threshold value. This threshold will depend on the clock frequency. A more advanced path selection procedure involves the process of selecting one of the longest paths through each circuit line. This is to ensure that all local delay faults are covered.

The second problem with path delay fault testing is with regards to the requirement for a robust test [75].

Definition of Non Robust Path : *Two conditions must be fulfilled for a vector pair {V1, V2} to detect a path P non robustly:*

- *The vector pair launches a transition (either rising or falling) at the beginning of the path*
- *All the off-path inputs along the paths have a non-controlling value (NVC) for vector V2*

Definition of Robust Path : *Two conditions must be fulfilled for a vector pair {V1, V2} to detect a path P robustly:*

- *the vector pair must fulfil the condition for Non Robustly testable path*

- *All the off-path inputs of G should be held at a steady state non controlling value whenever the on-path input of a gate G along the path transitions from a NCV to a controlling value.*

A single defect normally effects more than one path. Ideally a robust path is required to detect a path delay fault. However, in practice, a large number of path delay faults are not robust testable. As a result, these path delay faults have to be detected by applying non robust path delay test. Non robust and robust paths are explained hereafter.

Non Robust Path Delay Fault

A Non Robust (NR) path is a statically sensitisable path [75]. A path P is said to be statically sensitisable if there exists at least one input vector which stabilises all side inputs of path P, at non-controlling value (NVC).

A logic value is the controlling value (CV) to a gate if the logic value at an input to the gate independently determines the value at the output of the gate. In the case of an AND gate or NAND gate, the controlling value is logic 0. Similarly the controlling value for an OR gate or NOR gate is logic 1. A non controlling value (NVC) of a gate G is the complementary value of the controlling value.

A non robust test cannot guarantee the detection of a fault in the presence of other faults. A guaranteed test exists only for a subset of these non robustly testable path delay faults. These subsets are called validatable non-robust tests [75].

Robust Path Delay Fault

A robust path is a subset of non robust path. This is true since a robust path fulfils the requirements for the non robust case. In addition, a robust path is able to detect a fault in presence of other faults in the side paths. If all the paths in a circuit are robustly testable, then we will not require any other kind of delay tests. However, this is not a possible case in most circuits.

2.7 Summary

Testing in general and delay fault testing has been discussed specifically in this chapter. Testing was introduced followed by how physical defects can be modelled to enable testing process. One of the *de facto* fault model, the Single Stuck at Fault Model (SSFM), its limitations and alternatives were presented. The limitation is given to justify the need to use other fault models such as the delay fault model. Automatic Test Pattern Generation (ATPG) is presented as a tool to enable test vector generations. Fault simulation and its relation to fault coverage are discussed next.

Methods of implementing Design for Test (DFT) and how it will impact the testability of the circuit were presented. Finally, issues in relation to Delay Fault testing were explained. This includes the causes of delay faults, classification and methods to conduct delay fault testing. Even though the overall discussion in this chapter is broad, particular emphasis was given to issues in relation to delay fault testing.

In the next chapter, analysis of the two main spot defects in interconnect i.e bridging fault and opens are presented.

Chapter 3

Analysis of Bridging Fault and Open Defects

Modern VLSI circuits are interconnect dominant. The two main spot defects in the interconnects are bridging fault and open defects [76]. In this chapter, these two defects are studied in detail. In Section 3.1, the bridging fault, its analysis and different types of bridging faults are presented. A general introduction to open defects and analysis on resistive open is given in Section 3.2.

3.1 Bridging Fault

Shorts between circuit nodes are the predominant types of manufacturing defects [1],[4],[77]. This is proven by studies using Inductive Fault Analysis (IFA), as reported in [78], [79] as well as through experimental analysis in [80].

These shorts can be of two types: intra-gate shorts between nodes within a logic gate and inter-gate or external shorts between outputs of different logic gates [16], [81], [82], [83]. Inter-gate shorts, or bridging faults, account for about 90% of all shorts [83],[84].

Shorts between adjacent line give rise to bridging faults. These shorts are failures resulting from fabrication defects such as introduction of foreign particles, imperfections of masks and imperfections of photo litography [78]. Inductive fault analysis (IFA) gives a systematic method of determining the likelihood of a defect occur in a VLSI circuit. Given a physical implementation of a circuit, IFA tool can generate a list of possible faults.

IFA works by determining the effect of spot defects on the physical circuit. The effects of a defect on various regions of the Integrated Circuit (IC) are simulated in IFA. These different regions include the conducting, insulating and semiconducting region. If the simulation results show a possible fault, it will be reported. Therefore, only the realistically possible faults list are generated [79],[85].

It has been demonstrated that testing for inter-gate bridging faults will have a large impact on the final defective part level since most of the shorts are between the outputs of different logic gates. A non resistive bridge can be detected using static voltage based testing such as single stuck at fault test.

On the other hand, resistive bridging faults result in an intermediate voltage, between 0 and VDD due to voltage divider effect. Depending on the switching threshold voltage of the successive gate, this intermediate voltage will be interpreted as either 1 or 0.

These shorts are not modelled adequately using the traditional stuck at fault model. This is because most of these defects will not affect the logic level of the outputs [86]. The defect will cause a weak HIGH or weak LOW.

By modelling bridging faults as dynamic faults, more defects can be detected since defect not causing logical error will also be detected. As a result, the total fault coverage can be increased significantly. There have been more recent studies showing how bridging faults can cause timing failures [16],[87]. It is reported in

[16] that the delay caused by a bridge resistance can either increase or decrease depending on the input patterns.

In the next section, a detailed analysis of bridging faults is presented.

3.1.1 Bridging Fault Analysis

A bridging fault between two lines in a circuit occurs when the two lines are unintentionally shorted. To detect the bridging fault, the involved shorted nodes need to be set to opposite values. Figure 3.1 shows a bridging fault between the output of 2 NAND gates and its equivalent transistor level circuit.

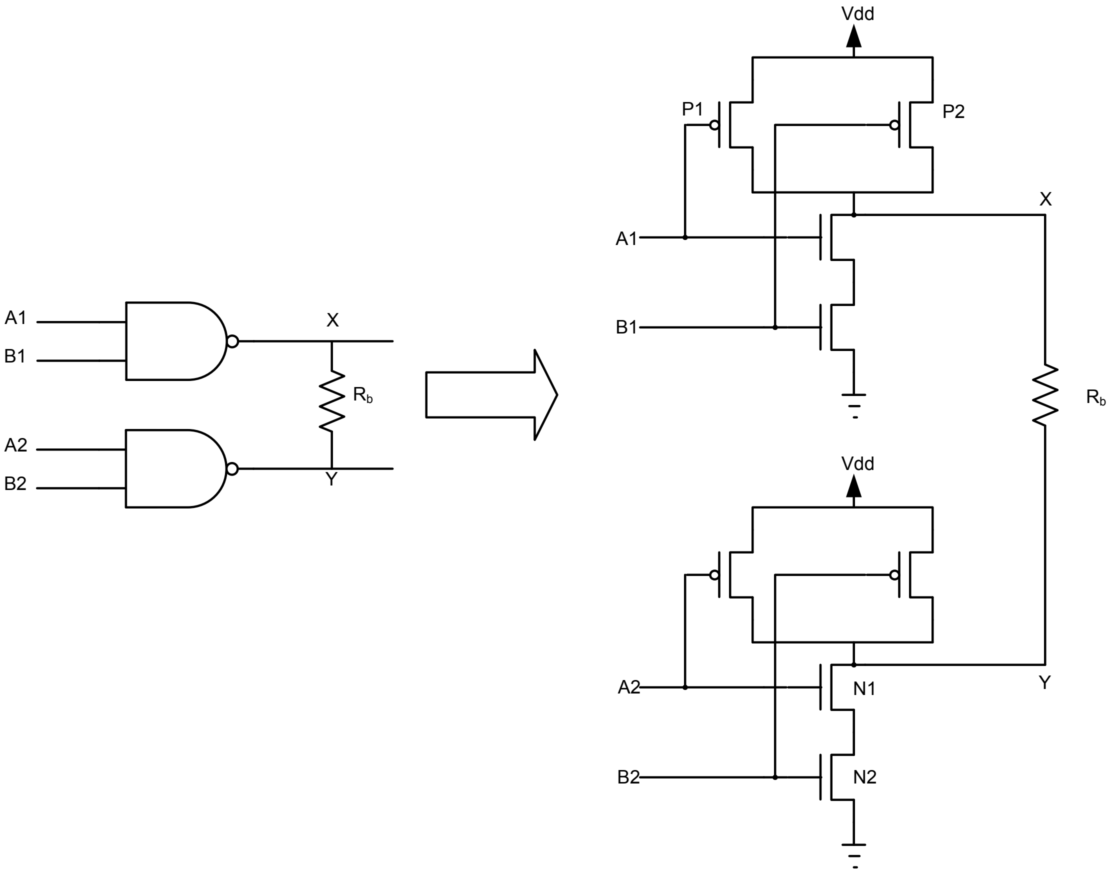


FIGURE 3.1: Mechanism for Bridging Fault

In Figure 3.1, assume that the value of the resistive bridge R_b is small. When the input vector is {A1,B1,A2,B2}={0,1,1,1}, both N1 and N2 will be conducting and only transistor P2 will be conducting. Let us assume the pull-down current is 600μA when both transistor N1 and N2 are conducting and the pull up current is 400μA when only transistor P1 is conducting. In this case, the pull-down current is higher and the output at the bridging node will be low.

If the input vector is {A1,B1,A2,B2}={0,0,1,1}, P1 and P2 will be conducting and the pull-up current will be 800μA. At the same time, pull-down current will be 600μA. Since the pull-up is stronger, the output will be high. In general, if the difference between the pull-up and pull-down is significant, the output can be predicted. However, when the pull-up and and pull-down are equal or even very close, the output voltage will be indeterminate [88].

In order to detect a bridging fault between nodes X and Y, the nodes have to be set to opposite values. Depending on the input vectors and the value of bridging resistance R_b, the bridging lines can have intermediate voltage values. These intermediate values are not well defined logic values of 1 and 0. This intermediate voltage can be interpreted differently by downstream gates. This happens when the downstream gates have different input logic threshold. The misinterpretation is known as the Byzantine General's Problem. Byzantine fault behaviour means that an intermediate value within a certain interval may be interpreted at different gates owing to the variations in threshold voltage between different gate types [89].

Consider Figure 3.2 . The bridged nodes a and b are fed into three different gates. The possible corresponding voltage distribution for intermediate values of R_{sh} is depicted as a solid curve in Figure 3.3[1]. The input vector applied is {A1,B1,A2,B2}={0,0,1,1}. Both the p transistors at gate A and n transistors in gate B are conducting.

As the value of the bridging resistance R_{sh} increases, the voltages at the bridged node, Va and Vb diverge, with Va approaching V_{DD} and Vb approaching 0. The 3 horizontal lines show the threshold voltage for gates C, D and E. The intersection

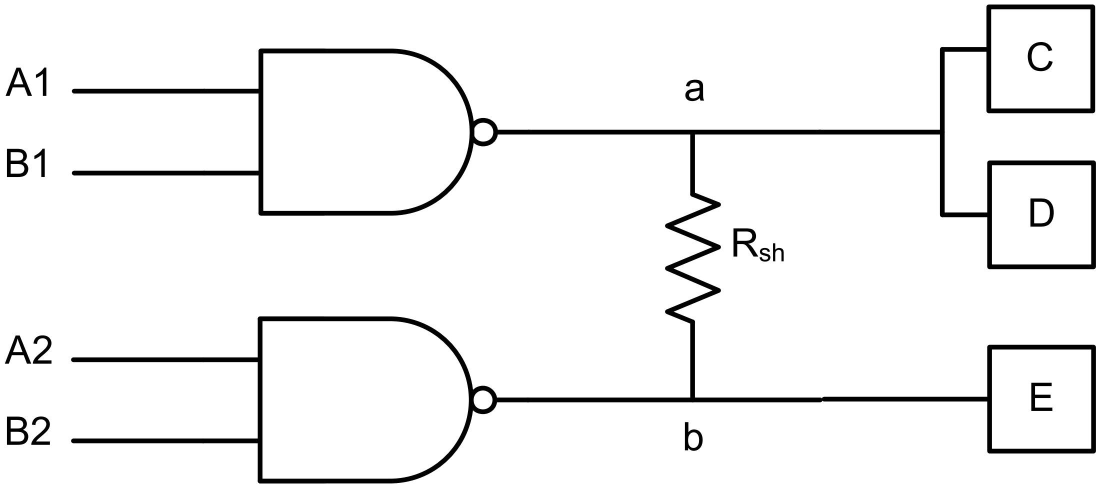


Figure 3.2: Bridging fault with downstream gates

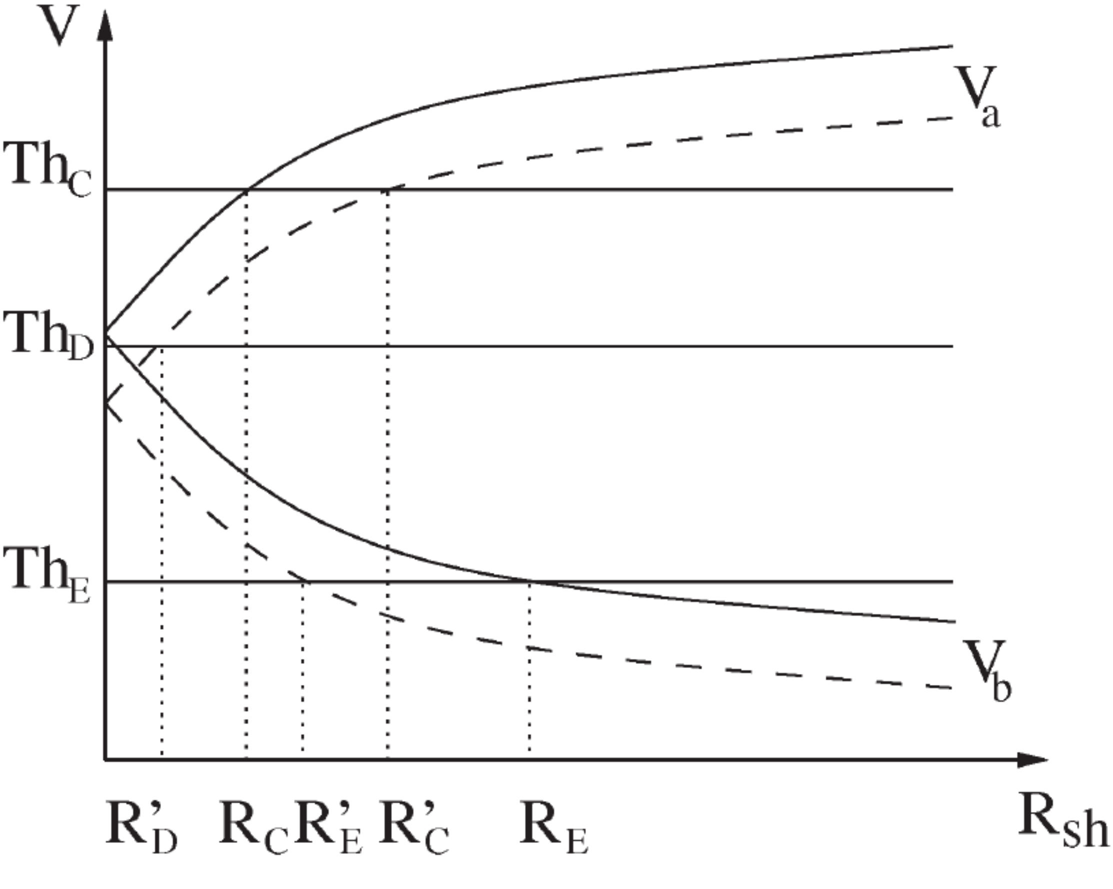


Figure 3.3: R_{sh} - V diagram [1]

between the threshold voltage and the voltage curve gives the range of resistance that can be detected by means of static stuck-at-fault (SAF) testing. For example, for gate C, the resistance value from 0 to R_C can be detected by SAF test. The resistance value above R_C will not cause logical error. However, the value will cause dynamic performance degradation which need to be detected by means of delay fault testing [16],[90],[91]. The value of resistance R_C is known as critical resistance. Methods to determine the critical resistance have been presented in several publications [1], [81],[92].

As it can be observed, the value of Va(Vb) right after R_C is still increasing(decreasing). At some point, when the value of R_{sh} is large enough, the circuit will interpret R_{sh} as open resistance whereby it will not have any impact on the circuit. Until this value of resistance is reached, the circuit may exhibit dynamic faulty behaviour.

For gate D, the threshold value is ThD, below the curve Va. This means that for any value of R_{sh}, the fault effect will not be propagated. There is no critical resistance for gate D. For gate E, R_E is the critical resistance which is relevant to curve Vb. The value of R_{sh} from 0 to R_E will cause gate E to interpret Vb as logic 1. When R_{sh} larger than R_E the circuit might have dynamic performance degradation.

If the input to the NAND gate A is changed to {0,1}, then only one p transistor will pull up the voltage of line a to VDD. This will result in logic 1 with less driving strength. At the same time, since the driving strength on Vb did not change, one possible voltage curve of Va and Vb are shown in Figure 3.3 as dashed line. This has resulted in new critical resistance R'_C and R'_E as well as critical resistance R'_D.

Using Figure 3.2 and assuming line a as victim and line b as aggressor, the bridge slow-down delay can be explained. The aggressor net will dominate the value at the victim net. As the resistance increases more than the critical resistance, the bridge will slow down the switching of the signals. The bridge slow-down delay is a function of the bridge degrading the quiescent voltage levels as well as the capacitance of the victim line. The longest delay can be expected when the victim is rising and the aggressor is low. A similar long delay can be expected when the victim is falling and the aggressor is high. This signifies the importance of the initial or previous condition. However, if the aggressor is much faster than the victim or if the transitions are slightly earlier, the previous level does not matter.

Critical resistance has been widely accepted as indicator or intersection where the value of resistance will determine if the defect will cause logical fault or dynamic fault [1], [81], [93], [86],[92]. However, Moore et. al [42] have raised the concern that there is no clear edge between these two regions. Their argument is based on the fact that signals on both sides involve signals whose quiescent levels are close to the gate threshold and will have small noise margins and unpredictable results. To cater for both arguments, we have run simulations at larger range of resistance such that the grey areas are also covered.

In this section, we have described the bridging fault using simplified circuit analysis. The importance of input vectors and how they affect the critical resistance has also been presented. Our areas of interest are beyond the value of critical resistance where dynamic analysis becomes very important to detect the bridging resistance [91],[93],[94]. Previous work on increasing the range of detectable bridging resistance were conducted by means of delay fault testing at low voltage and high temperature [86],[90],[95].

3.1.2 Different types of Bridging Fault Models

A thorough study on various types of bridging fault has been conducted. This will help in performing suitable fault simulations such that all possible bridging defects scenarios can be covered. An exhaustive description of different types of bridging fault is given in [92], [96]. We have used these classification in all our simulation work. A brief description of each of the models is given in the following subsections.

3.1.2.1 Bridge between two primary inputs

Since primary inputs are sources of infinite current, bridging faults between them are not logic testable. As a result, these types of bridging fault are not modelled.

3.1.2.2 Bridge between a PI and gate output

Figure 3.4 illustrates a bridging fault between a primary input A and the output of 2 INPUT NAND gate. Node X will be feeding two gates having different threshold voltages. The logic threshold of the two driven gates and the test vector at inputs A, B and C will impact the detectable range of bridging resistance. It is important to ensure that the inputs E and F are set to non-controlling values.

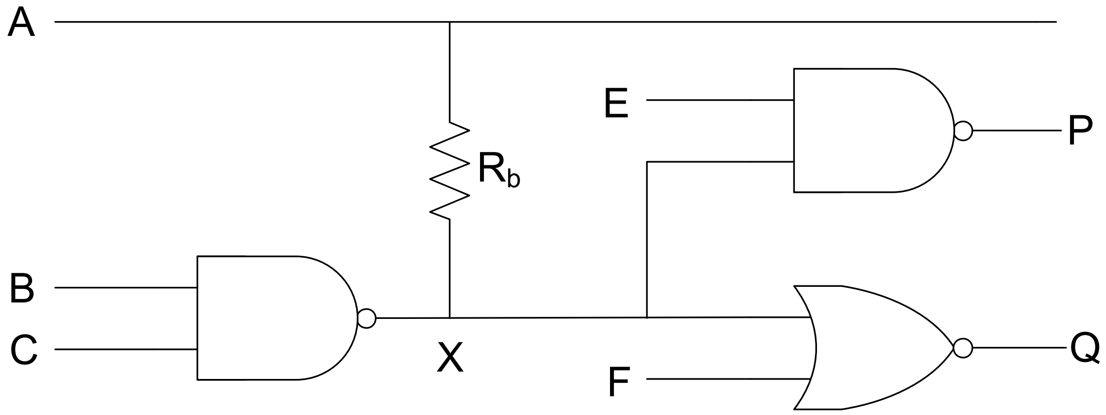


Figure 3.4: Bridging fault between PI and gate output

3.1.2.3 Bridge between two gate outputs (bridged nodes feeding into different gates)

Figure 3.5 shows a bridging fault at the output of a NAND and a NOR gate. The bridged nodes X and Y are feeding to a number of different gates. The detectable resistance range at the outputs depends on the test vectors at input {A1,B1,A2,B2} as well as the logic threshold of the driven gates.

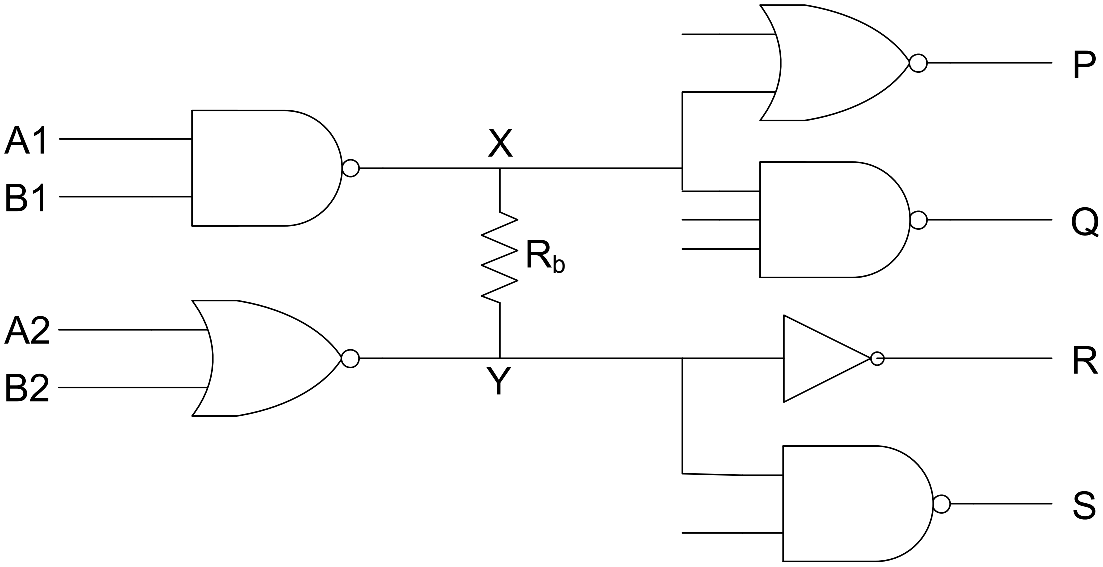


Figure 3.5: Bridging fault between nodes feeding different gates

The fault will propagate either through node X or node Y. If the fault propagates through node X, it can be detected at outputs P and Q. However, the fault will not be detected at outputs R and S. On the other hand, if the fault is propagated through node Y, it will only have an impact at outputs R and S.

3.1.2.4 Bridge between two gate outputs (bridged nodes feeding into the same gates)

Figure 3.6 illustrates an example of a bridging fault between the outputs of two gates, NAND and NOR. The bridged outputs are fed into the same three input NAND gate. The detectable resistance range at the output P will depend on inputs {A1,B1,A2,B2}.

3.1.2.5 Bridge involving two primary outputs

This will be similar to the case of Bridge between a PI and gate output as in Section 3.1.2.2.

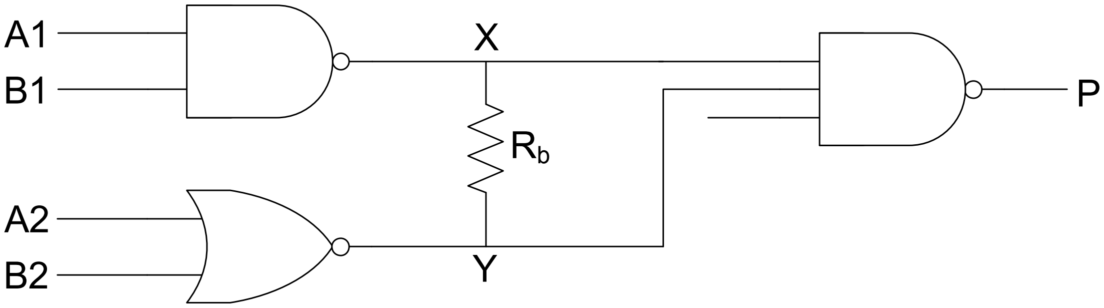


FIGURE 3.6: Bridging fault between nodes feeding different gates

3.2 Open Defects

Open defects have been traditionally defined as unconnected nodes in a manufactured circuit that were connected in the original design [93],[97], [98], [99], [100]. They can cause behaviour that may vary greatly and be difficult to predict. These defects include open contacts (missing metal or unopened oxide), metallisation opens (patterning, improper etching, electromigration, or stress voiding), or opens in diffusion or polysilicon (mask or fabrication errors) and broken vias [93], [99],[101].

If the defect cause strong opens, it will immediately affect the circuit's yield and these opens can be detected using static voltage based stuck-at-fault [102]. Strong opens are resistive opens with resistance value more than 10 Megohms.

Hawkins et. al [93] have graded the opens in CMOS circuits into 6 different classes according to defect properties. The defect properties depend primarily on defect size, defect location, local electrical structure, and process variables.

Even though most of the classes of the opens from [93] can be detected by either stuck-at-fault test or IDDQ test, there is a class of defects which needs to be tested at-speed. This class of open defects is known as weak open.

A weak open defect can still connect the two points in the network but it will be weakly connected. This will introduce higher-than-expected resistance between the linked points [103]. The defect with finite resistance will still allow the circuit to function but it will have performance degradation in terms of circuit delay.

Weak opens are potential hazards since they can escape the traditional stuck-at-fault test. Two pattern delay fault tests are required to detect the weak opens [40], [50],[94].

The resistance distribution of these weak opens is roughly flat if the resistance values are separated by an order of magnitude [102].

3.2.1 Resistive open analysis

Figure 3.7 shows a model of resistive open fault. The resistive open R_{op} is an open in the interconnect at the output of CMOS gate C [104].

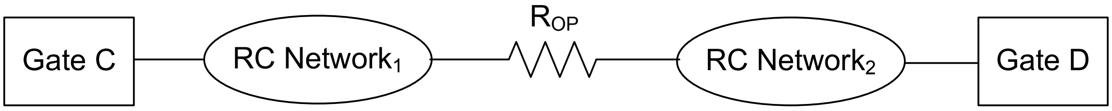


FIGURE 3.7: Resistive open fault model

The nominal delay of the gate C in absence of the defect is given as:

$$D_{nom} = D_{nocharge} + \alpha C_L \tag{3.1}$$

where $D_{nocharge}$ is delay value without considering the load capacitance and α is the constant factor. These values are readily available from gate library. The lumped load capacitance C_L is the sum of input capacitance of all gates driven by gate C and the parasitic capacitance of the interconnect. For an example, the lumped load capacitance for Figure 3.7 given by:

$$C_L = C_{gateD} + C_{line} \tag{3.2}$$

where C_{gateD} is the load capacitance within the CMOS gate D driven by gate C and C_{line} is the capacitance of the interconnect.

Finally, the delay of a gate driving an interconnect with a resistive-open defect R_{OP} is

$$D = D_{nocharge} + \alpha C_L + \beta R_{OP} C_L \tag{3.3}$$

[104]

Factor β depends on the electrical parameters of the driving gate C. β value can be precomputed as its value is constant for a given type of gate and independent of R_{OP} as well as C_L [104].

This signifies that the delay of the circuit with the introduction of the resistive open defect R_{OP} is directly related to value of resistive open as well as the loading capacitance C_L. This will result in delay increased linearly with the open resistance. It has also been shown through simulation that above a certain value, depending on the clock frequency of the circuit, the resistive open becomes stuck-open fault [76]. The value where the delay resistance becomes stuck-at fault is known as the critical resistance.

3.3 Summary

This chapter began by showing the importance to detect the bridging fault. Bridging fault analyse from published works show that resistive bridging fault can cause a logical error as well as timing failure. Similar observations were found for resistive open defects. In the next chapter, issues in relation to testing for multi voltage design are presented.

Chapter 4

Testing for Multi Voltage Design

The need for energy efficient devices has been growing rapidly with the advancement of mobile technology. As the complexity and density of circuits gets higher, new sets of design problems exist. The power density of the highest performance circuit has reached a maximum limit whereby it is no longer possible to increase the clock speed as the geometry of the circuit shrinks. The dynamic power consumptions need to be reduced in active mode and static power needs to be controlled while the system is in standby mode. Adaptive power management techniques which in general scale the voltage supply (VDD) according to the processing load are commonly used.

Effective Design-For-Test (DFT) methods for these multi voltage designs is an essential part of the design and test process. In this chapter, issues in relation to testing these types of systems are reviewed. Section 4.1 looks at available low power design methods with emphasis on Multi Voltage Design. Testing issues for multi voltage design are discussed in Section 4.2. A very mature test method, Very Low Voltage testing, is reviewed in Section 4.3. Finally in Section 4.4, Level shifters, a key component in multi voltage design are discussed.

4.1 Low Power Design

In a SOC design, the total power consumption consists of dynamic and static power. Dynamic power is the power consumed when the device or the circuit is in active mode. This is when the signal is changing value. In addition to switching power, internal power within a CMOS structure also contributes to dynamic power. The dynamic power decreases quadratically with the decrease of supply voltage. On the other hand, static power is the power consumed when there are no active signals switching. The main source of static power is leakage current.

To enable longer battery life for mobile devices, steps are taken during all aspects of designing, from software development up to the hardware implementation [71]. Among the steps taken are power gating and the use of multi-threshold libraries. Power gating techniques use two power modes: a low power mode and an active mode. By switching between these modes at the appropriate time and condition, power savings can be maximised while the impact to the performance can be minimised. Multi-threshold libraries enable the designers to choose from different versions of cell libraries in their design. Many libraries today offer up to three versions of their cells: Low, Standard and High threshold voltage. Where performance is not critical, High VT can be used to decrease the leakage current.

Previous design approaches use a single supply voltage for all gates. However, this has resulted in challenges in handling high total power consumption. To address this problem, new techniques using multiple supply voltages have been introduced. Depending on the temporal requirement, each block will have different supply voltage. In more advanced techniques such as Dynamic Voltage and Frequency Scaling, supply voltage and frequency are dynamically changed depending on the workload and performance required. The aim of our work is to suggest a testing methodology for Multi Voltage Design. Thus, we will concentrate on specific issues related to multi voltage design.

4.1.1 Multi Voltage Design

Multi Voltage Design has evolved from previously used power management technique known as Dynamic Voltage Management (DVM) . The general idea in Dynamic Voltage Management is to shut off the parts of the circuitry that are not active. This is either done by immediately shutting down idle parts or by using time based shutdown, i.e. to shutdown after a certain timeout period. This technique is a very mature technique and used in Advanced Power Management (APM) in notebooks and other mobile devices. The main advantage of this method is its generality whereby it can be used not only on digital circuitry but also on the analogue parts. However, since restarting the system and restoring the states involve a power and time overhead, the advantage of the system has been questioned. It has been reported that this greedy policy might even increase the dissipated power [71].

Almost all the latest low power techniques have been realised with the underlying concept that different parts of the circuitry do not always require the same supply voltage. Consider an example with four blocks of circuit as shown in Figure 4.1. At certain times four different blocks can be operated at different supply voltages. Each multiple voltage region has its own supply voltage. The USB block is running at a lower voltage since it has constraints in terms of protocol. On the other hand, the cache RAMS which are on the critical path are running at maximum voltage. The rest of the chips are running according to the computational requirement at a specific time.

Multi voltage design methods can be divided into four main categories :

- Static Voltage Scaling (SVS): Fixed supply voltages for different blocks of the system

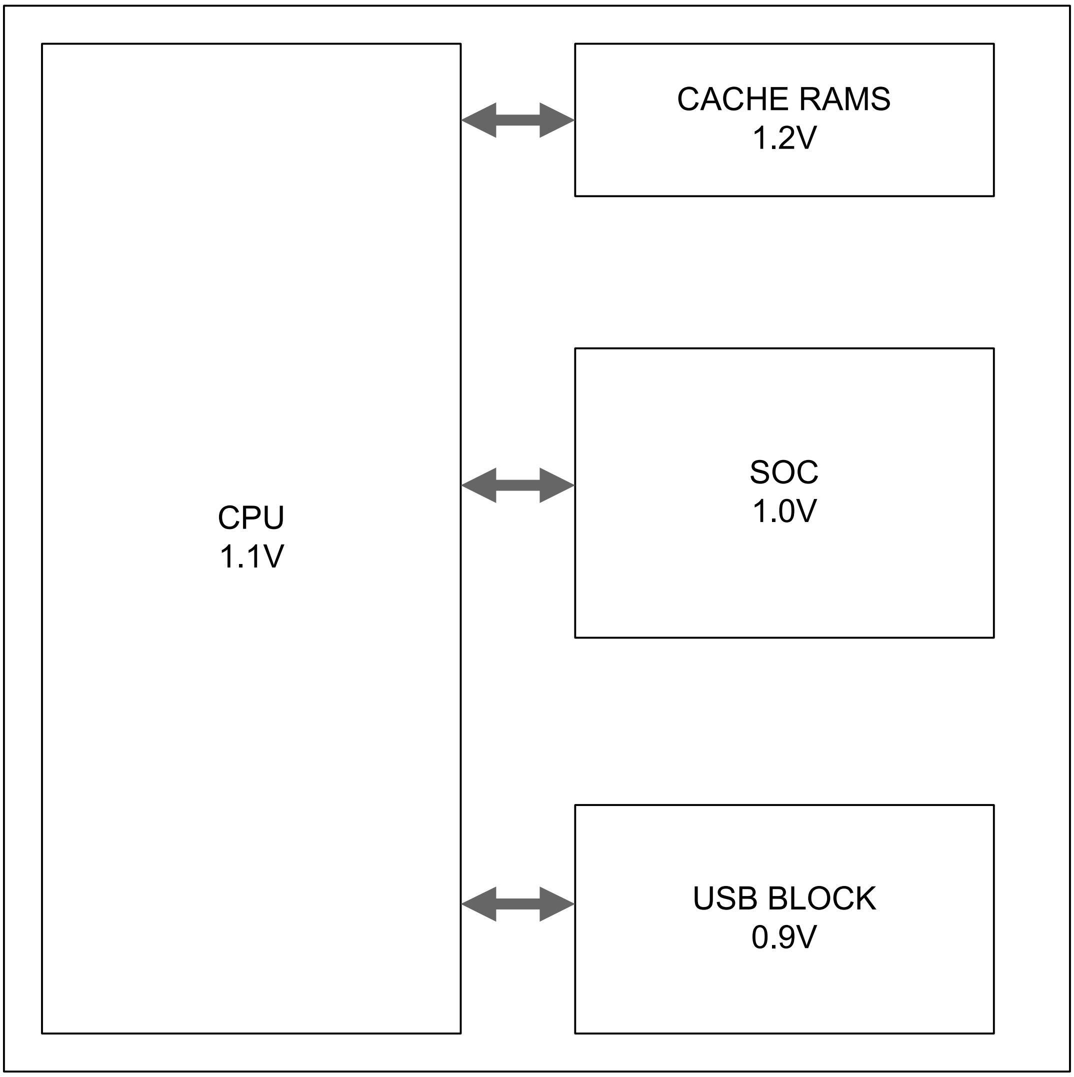


Figure 4.1: Multi-Voltage Architecture

- Multi-level Voltage Scaling (MVS): Extension of SVS with limited voltage levels for a block of system. Only a few fixed, discrete levels are supported for different operating modules.

- Dynamic Voltage and Frequency Scaling(DVFS): Extension of MVS with a higher number of voltage levels. The voltage levels are dynamically switched according to performance requirements

- Adaptive Voltage Scaling (AVS): Extension of DVFS where voltage is adjusted by a control loop

Dynamic Voltage and Frequency Scaling(DVFS), which is also known as Dynamic Voltage Scaling (DVS) [105] and Adaptive Voltage Scaling (AVS) are more advanced techniques since the voltage and/or frequency are/is dynamically adjusted. A DVFS enabled processor has the ability to dynamically vary the supply voltage and the operating frequency. This is done during the run-time of an application. The voltage frequency pair setting is adjusted according to the temporal performance requirement of the system. In contrast to traditional low power design techniques, the DVFS method does not sacrifice the throughput of the system. Previous low power design techniques such as CMOS scaling, generally sacrifice throughput. In CMOS scaling, longer battery life is achieved in portable systems such as PDA (personal digital assistance), mobile phones and laptops by reducing the supply voltage. However, in a standard PDA that is not using a DVS system, the throughput is significantly reduced due to lower power supply. This reduction is applied to all parts of the processor and the user can observe the impact when running applications that need higher throughputs such as video compression.

In contrast, DVFS exploits the fact that the clock frequency of a processor changes proportionally with the supply voltage, while the dynamic energy is proportional to the square of the processor's supply voltage. Running the processor at a slower speed means that the supply voltage can be lowered, yielding a quadratic reduction in the energy consumption at the expense of increased execution time.

DVFS techniques have been reported to reduce the system energy consumptions by up to 10 times [105]. This is done without sacrificing the desired throughput. It is made possible due to a time-varying computational load that is commonly found in most systems. An example of a microprocessor desired throughput in millions instructions per second (MIPS) as a function of time is shown in Figure 4.2.

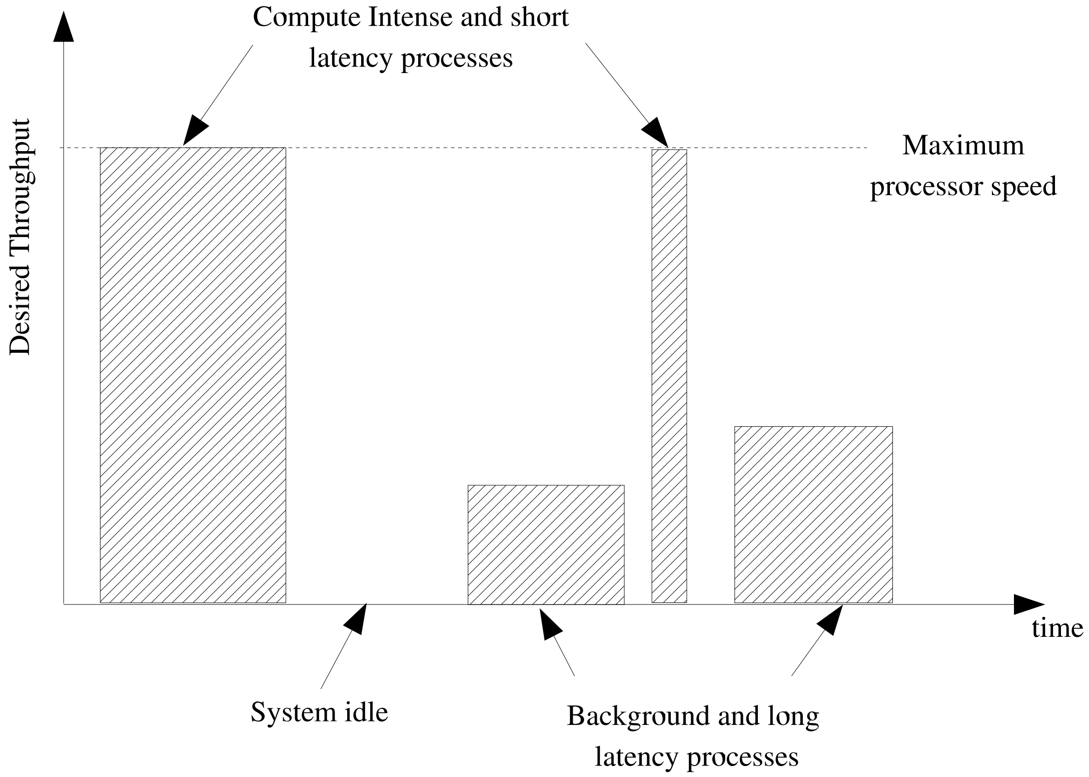


FIGURE 4.2: Processor usage model

Three categories of computational requirement are evident from the Figure: compute-intensive, low-speed and idle. Compute intensive process such as image compression, neural-network processing and complex mathematical calculations use the full throughput of the processor. These processes have short latency and have a strict timing requirement. On the other hand, low speed and long-latency tasks such as word processing and address book browsing do not need full throughput. Running these tasks faster than the required speed will not give any benefit. Therefore, the designer can adapt the processor voltage/frequency (V/F) pair setting according to the requirement and exploit the energy/speed trade off. As an efficient energy reduction technique, DVFS has been implemented in several contemporary embedded microprocessors such as Intel's XScale [5], Transmeta's Crusoe [6], and ARM's IEM [7] with different (V/F) pair settings.

4.2 Testing for Multi Voltage Design

With the growing interest and application of Multi Voltage Designs, there is a need to understand how defects behave in a multi voltage circuit. Even though testing itself has reached a mature stage, testing for multi voltage designs is a new research area. The main difference of multi voltage circuits in comparison to normal circuits is the operating condition. Multi voltage circuits are designed to operate at different voltages and sometimes at different Voltage/Frequency pair settings. To achieve the expected fault free operations, a naive approach is to test them at all these conditions. In theory, this might look straightforward and the only method that will give the highest test quality. However, in practical term, this will incur the highest test cost. Rajski [106] has addressed the need for high-quality low-cost test. Rajski outlined the requirement of an acceptable Design for Test (DFT) methodology. Among the required characteristics are that not only it should be able to handle any types of fault models but it must also be able to achieve acceptable test quality using low volume test data as well as short test time. Low volume of test data is important since it has direct impact on the required tester memory and cost of the tester. Minimising the time is important since test cost of a single device is determined by test time in seconds.

Rodriguez-Irago et al. [107] proposed a method to build characteristic histogram by using varying VDD as in multi voltage design. The characterisctic histogram shows the performance of the system when the supply voltage is varied. A delay model based on varying VDD for both gate delay and path delay were derived. By using the proposed model in [107], the minimum VDD level to ensure fault free operation can be computed. Rodriguez-Irago et al. [107] also claims that multi voltage test can uncover delay faults. This is achieved by looking at the difference between faulty and non-faulty histograms. The difference in histogram can further be used as a diagnosis tool. In their consequent work, Rodriguez-Irago et al. [108]

have addressed the similar issue. In addition to varying VDD as in multi voltage design, [108] have included varying temperature as well.

In general [107] and [108] have exploited the on-chip availability of multi VDD on Dynamic Voltage Scaling enabled chip. Both paper suggest using the difference in output signatures as a way to detect fault existence. BIST platform has been suggested for their approach. Even though they have shown the relation between varying voltage and delay fault, they have not indicate the relation between the voltage levels and how it impact different types of defects.

There are also extensive reporting on circuit testing at low voltages [40], [109], [110], [111], [112]. Most of these work was done with the aim of finding the highest fault coverage by reducing the supply voltage from the nominal supply voltage to a very low voltage. This method is known as Very Low Voltage (VLV) testing. Even though VLV testing does not address the testing issues for multiple voltage systems, the results are very relevant for our work.

Previous work shows the importance of testing at multi voltages. Experimental results in [113] show the effect of operating conditions and process variations on circuit delays. The variation of gate delay propagation with power supply for non-voltage-compensated circuits was demonstrated experimentally.

Another study, [114], reported that the electrical performance is affected by environmental and physical factors, of which the power supply is one of the most critical factors. Other factors include temperature and physical factors caused by processing and mask imperfection.

4.3 Very Low Voltage Testing

Hao and McCluskey [109] introduced the concept of very low voltage testing. They reported that there are certain types of defect which can be categorised as flaws and these defects will not be detected at the nominal test voltage. These undetected flaws will result in a weak Integrated Circuit (IC) which will not have failures in normal operating conditions but will suffer in degradation in performance and noise immunity.

These weak ICs can bring problems under two conditions. In the first condition, deterioration over time can cause a catastrophic problem. Among the flaws that can cause problems after a certain time are gate oxide shorts, missing material on metal wires and inadequate channel length. In the second condition, the chip might operate intermittently [109]. Resistive shorts and hot carrier effects flaws have been reported to cause intermittent problems. The problem can be observed when there is a change in operating environment or different operating conditions.

The general idea behind using low voltage testing is the voltage dependency of CMOS circuit operations. At high voltage, logic circuit switching is high and should be limited to avoid damage to the chip. On the other hand, a lower voltage will slow the switching speed of the circuit and below a certain value, the chip will not function at all. With the presence of flaws, the voltage dependencies are magnified and enable the flaws to be detected at lower voltage.

[109] explained the impact on voltage dependencies on resistive shorts. A short in a CMOS logic circuit will introduce a static current path between VDD and Ground. The resistance of the resistive short does not change much with the voltage and can be assumed as constant for simplification. On the other hand, as the power supply reduces, the gate-to-source voltage in the static current path also decreases and this makes the transistor more resistive. The resistance ratio between the

resistive short and the transistor will become smaller. This will eventually make the impact of the short more severe.

In theory, the power supply can be reduced to slightly higher than the threshold voltage for VLV testing. However, reducing to this theoretical limit will introduce new problems such a low noise margin and excessive circuit delays. McCluskey [110], [115], [116] have suggested that the best trade-off between fault coverage and supply voltage is to run the test at $2V_{th}$ to $2.5V_{th}$.

The most relevant publication for our study is another work by Chang and McCluskey [40] where the impact of VLV testing on delay faults has been presented. They have looked at non-operational delay faults, i.e. the circuit works without any degradation at designed speed and voltage but will expose the flaws when tested at a lower voltage. By looking at the voltage dependencies of the CMOS propagation delay, the usefulness of a lower voltage to detect delay flaws has been presented. It is observed that when the supply voltage is between 2 and $2.5V_{th}$, the maximum changing rate of the propagation delay is observed. This finding is in line with their previous findings on static faults. Detailed simulation results were given for three different faults: transmission gate open, threshold voltage shifts and diminished-drive gates.

Renovell et al. [117] have shown through simulations that by lowering the supply voltage, the range of critical resistance have been increased. These indicate increase in the range of resistance that can be detected. As a result the fault coverage increased by 40%. Their work was based on detecting logical error due to resistive bridging fault.

Yuyun et al. [86] addressed the advantage of VLV testing in relation to test vector analysis. In general, the maximum detectable resistance and fault coverage are dependent on the test vector. The random vector strategy which is normally used in industry has shown lower than normal fault coverage for certain types of bridging fault. However as the voltage is reduced to VLV level, the difference in terms of fault coverage between random test vectors and refined test vectors has reduced significantly.

Yan et al. [112] presented experimental results on testing circuits at low voltages. Their results taken from specially design test circuits confirmed that it is effective to correlate the observed delay fault with the change of supply voltages. Their aim is to use the result for diagnosis purpose. This is done by differentiating the resistive interconnect faults to other faults such as weak transistor defects and output capacitive faults.

4.3.1 Issue in VLV testing

VLV testing suffers from two main disadvantages. These are performance degradation and possible coverage loss.

Running the test at lower voltage will increase the test application time. The increase of test application time will eventually results in performance degradation [4]. As the voltage is reduced, the operating frequency decreases, and hence the number of test vectors that can be applied at a given time will reduce. On a scan chain based design for test (DFT), the scan speed is limited by three major factors: the tester capability, power during scan and the scan chain capability. With reduced supply voltage, the first two factors do not have any impacts. However, the scan chain capability will be reduced.

To illustrate the impact of reduced voltage on test application time, we have interpreted some published results for the Transmeta derived from the relation between supply voltage and operating frequency.

The operating frequency of the processor is given as in Equation 4.1

$$f = (L_d K_6)^{-1}((1 + K_1)V_{dd} + K_2 V_{bs} - V_{th1})^{\alpha} \tag{4.1}$$

L_d is the logic depth of the path, K_1,K_2 and K_6 are the constants for a given process technology and α is a measure of velocity saturation. V_{bs} is the bulk source voltage and V_{th1} is the threshold voltage. V_{bs} is set to zero since we are not considering the effect of body biasing in our analysis. The value of the threshold voltage is given as 0.359V. The normal processor operating voltage is between 1.2 to 1.6V.

Figure 4.3 shows the relation between the supply voltage and normalised frequency. At 1.2V, the operating frequency is around 70% of that at 1.6V. Tests at this voltage would result in the test application time increasing by 1.4 times compared to the time taken at the highest operating voltage.

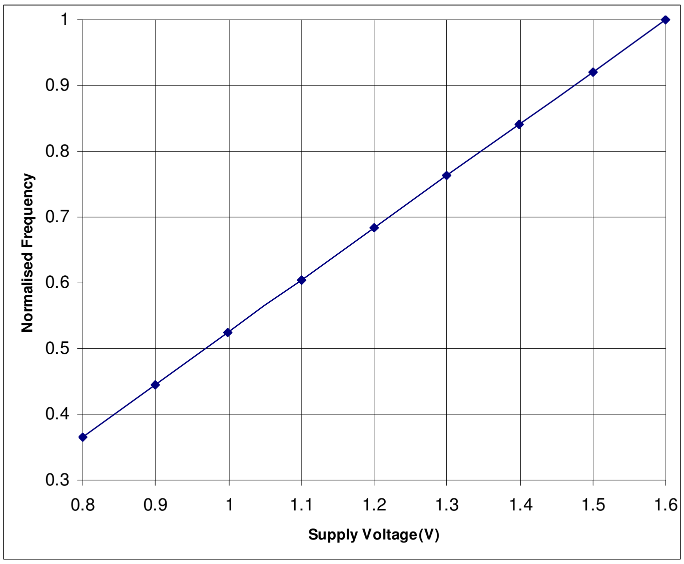


Figure 4.3: Supply Voltage versus frequency for Transmeta Crusoe 5600 Processor

In practice, the actual increase of test time is around 1 to 2% for every % drop of the voltage from nominal value [118]. Assuming a drop of 1% for every % drop of voltage, Figure 4.4 shows how the performance degradation can affect the quality of the test. If the test application time is limited to t_x, a test at nominal voltage at 1.6V can have 5 test vectors applied. As the testing voltage is reduced, the total number of test vectors will be reduced. For 0.8V supply voltage, only three test vectors can be applied in the give time, t_x. The analysis above will give the option for the user to choose between reducing the voltage or having higher test vectors applied at higher voltage.

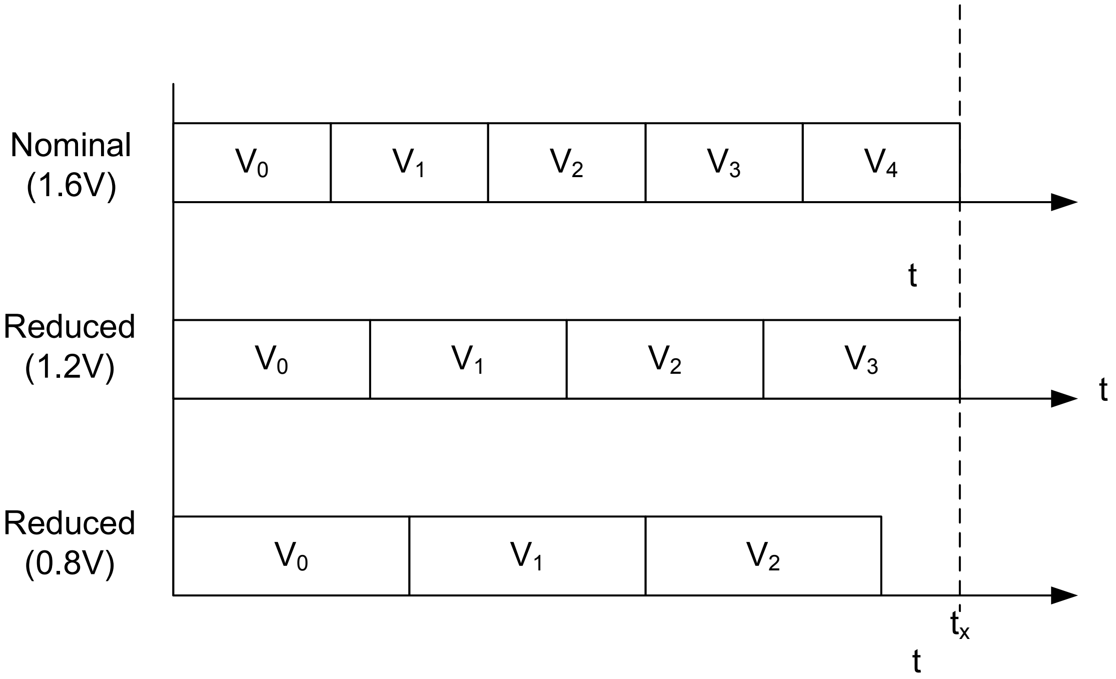


FIGURE 4.4: Supply Voltage versus frequency for Transmeta Crusoe 5600 Processor

It has also been reported in [2],[4] that VLV testing results in coverage loss. In other words, a particular range of resistive fault values is detectable at one voltage, but not at a lower voltage. The study in both [2] and [4] looked at resistive short faults that cause stuck faults. Their findings can be understood by referring to the circuit in Figure 4.5 and its corresponding R_{sh}-V diagram in Figure 4.6. These figures were taken from their publications.

In Figure 4.6, the nominal voltage response is shown in solid line V^{nom} and the corresponding threshold voltage line for two gates, C and D, are shown as Th_D^{nom} and Th_C^{nom}. The area detected at nominal voltage is the range of short resistance R_{sh} between R_C^{nom} and R_D^{nom}. When low voltage testing is applied, both the voltage

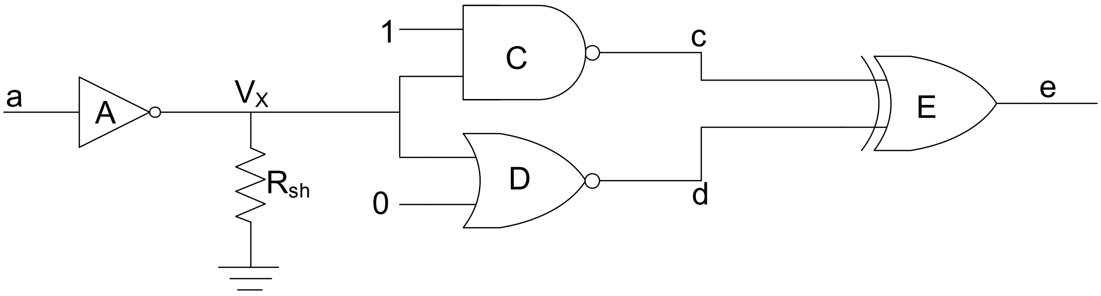


FIGURE 4.5: Example circuit for coverage loss study

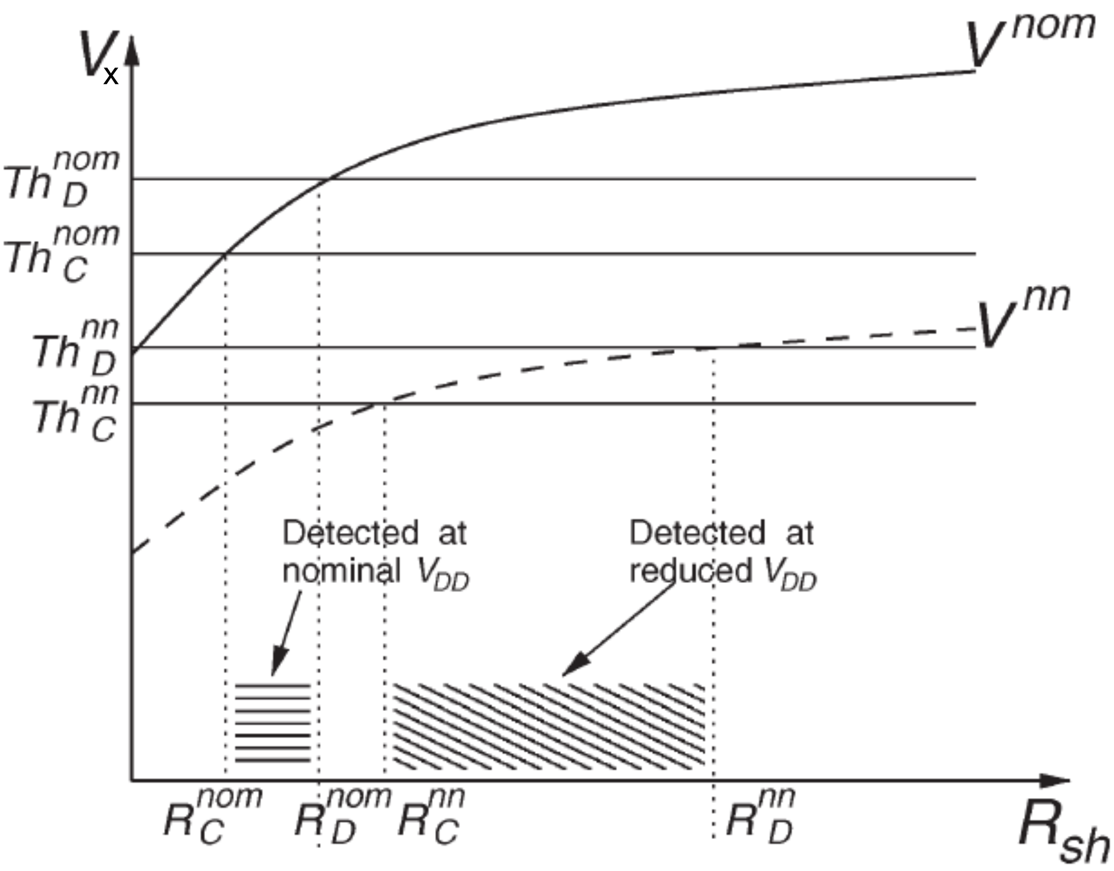


FIGURE 4.6: R_{sh}-V diagram showing the coverage loss [2]

characteristic and threshold shift down. This will result in a new critical resistance R_C^{nn} and R_D^{nn}. It is evident from Figure 4.6 that the new R_{sh} range between R_C^{nn} and R_D^{nn} is much larger than range covered by R_C^{nom} and R_D^{nom}. However, the new range obtained from low voltage testing did not cover the previously detected range under nominal voltage. Engelke et al. [2] have stated that the coverage loss at the lower supply voltage is due to propagation through an XOR gate as the reconvergency point. As a result, they have concluded that the behaviour is possible to happen for conventional transistors.

4.4 Level Shifter

One of the key circuit components in a multi voltage design is the logic-swing level shifter [119],[120], [121]. There is a need by industry to address the problem with regard to testing of level shifters in multi voltage designs [122].

In a multiple voltage system, more than one voltage domain are formed on a single Integrated Circuit(IC) or System on Chip (SoC). A voltage level shifter is required to interface between these different voltage domains. For a chip-level multi voltage system, level shifters are required between the core circuits and Input/Output circuits. In a block level multi voltage system, level shifters are required among the blocks in order to avoid crowbar currents at the received side.

Figure 4.7 shows a possible use of a level shifter in multiple voltage domains. The level shifter's function is to bring a signal from voltage domain x to domain y. It is also important to note that each block is powered by an external power supply. These power supplies are normally provided through bulk regulators.

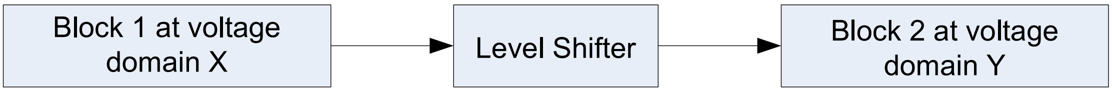


FIGURE 4.7: Level shifter in block level voltage domain

A Conventional Level Shifter (CVLS) as shown in Figure 4.8 requires two voltage supplies, the input domain supply VDDI and the output domain supply VDDO. The operation of the level shifter can be explained as below:

When the input signal in is at the VDDI value (inb is at GND value), MN1 turns ON (MN2 is off). This pulls the outb signal to GND. This transition of the outb signal turns on MP2 which pulls up the out signal to the VDDO value. When in is at GND (inb is at VDDI value), MN1 is off and MN2 is on, which turns on MP1. MP1 pulls up the outb to the VDDO value. Although there are no high leakage paths from VDDO to GND in this circuit, two supply voltages are required for the voltage level conversion. This can be a hard requirement to satisfy, especially if the VDDO and VDDI domains are separated by a large distance. The supply voltage wires typically need to be quite wide (especially if VDDO and VDDI are physically far apart), resulting in a large area penalty.

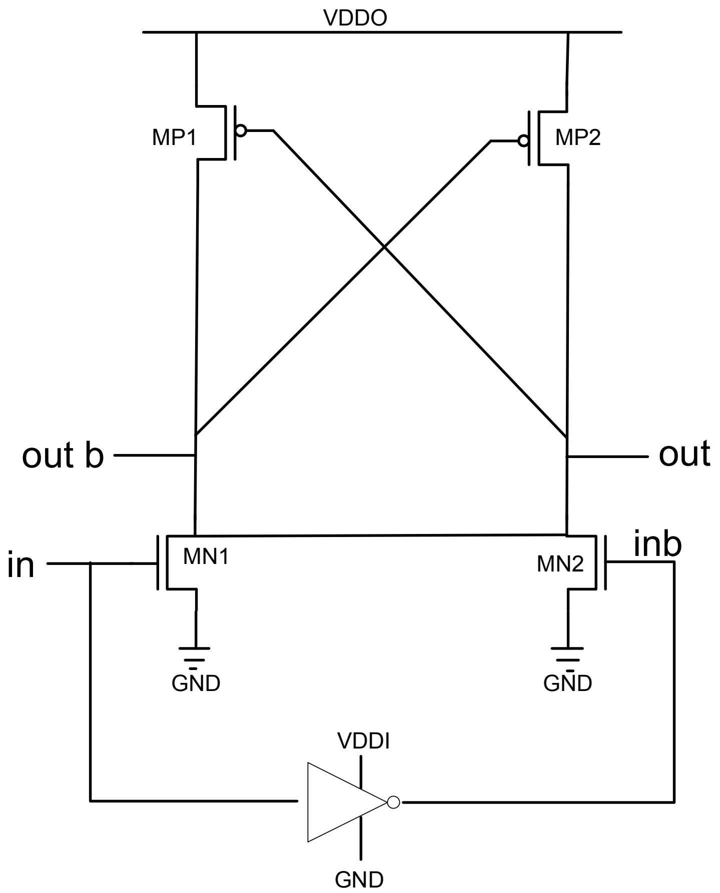


Figure 4.8: Conventional Level shifter

Due to the contention between pull-down transistors (MN1 and MN2) and pull-up transistors (MP1 and MP2), conventional level shifters have a large delay and high power consumption [119]. The problem of contention get severer when the low voltage VDDI changes. This is because it is difficult to get rid of the contention in both cases where VDDI is relatively low by proper sizing of transistors.

Canh [119] has proposed Contention Mitigated Level Shifters (CMLS) to handle the contention issues in conventional level shifters. Figure 4.9 shows the contention mitigated level shifter.

In the CMLS, the contention is reduced, since MN1 and MP3 (MN2 and MP4) comprise a quasi inverter. Therefore, the logical values of node A and node B are established faster than that of the conventional level shifter. Thus the delay of CMLS is less than that of conventional level shifters. The power consumption of the CMLS is reduced compared to conventional level shifters because the contention reduction also brings in the crowbar current reduction. Through simulations, the delay and power consumption of CMLS have reduced by 50% and 24% respectively. The increase in area overhead is only 4% [119] .

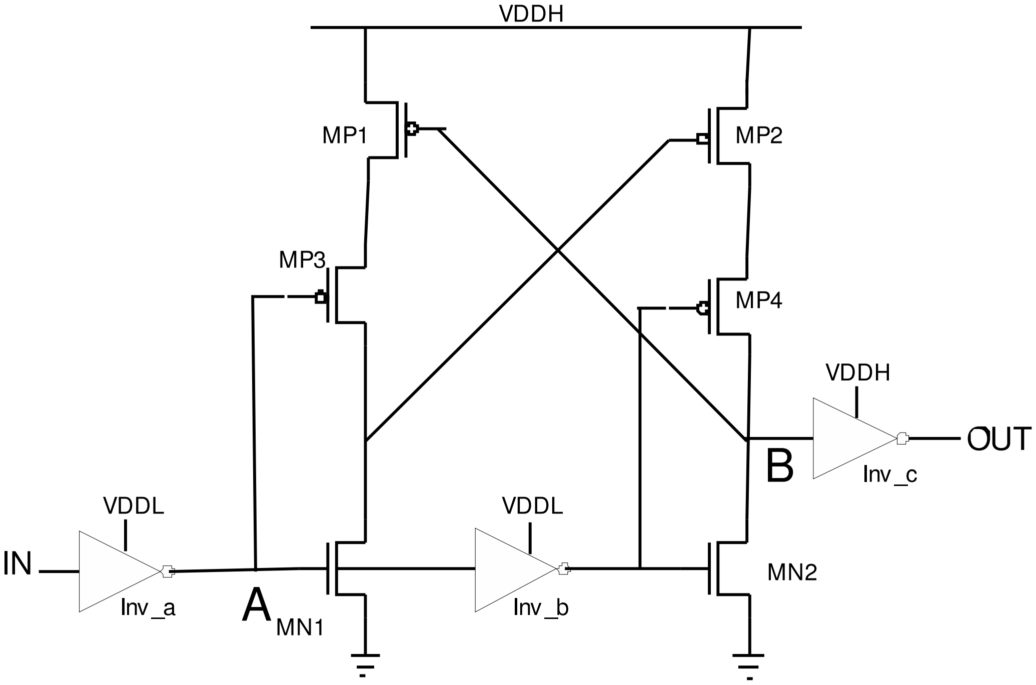


Figure 4.9: Contention mitigated level shifter

The second type of level shifter proposed by [119] is Bypassing Enabled Level Shifters (BELS). BELS has a bypass function which is active when both input and outputs are at the lower voltage domain (VDDL). By using the bypass function, the logic value is established factor at node B and the contention at node B will be reduced. Even though the BELS have delay and power reductions of 65% and 50% respectively, it has penalty in term of area overhead increase by 60%.

Khan [123] and Rajesh [121] have proposed single supply level shifters for multi-voltage systems. The proposed design is claimed to shift any low voltage to a higher voltage with reduced leakage current and is also capable of shifting at a high frequency. The proposed single supply voltage have strict transistor sizing. As a result, even minor error from design specification might cause the level shifter to malfunction.

In general, level shifters are used to shift from low to high translation. For a high to low translation, normally, two inverters in series are used to shift down the voltage. There will be a very small buffer delay and their impact on timing is insignificant. In practice, the level shifters are better located at the destination domain, i.e. at the lower domain for high to low transactions and in the higher domain for low to high domain [124].

4.4.1 Logic Interpretation Voltages in Multi Voltage Systems

The objective of this section is to demonstrate why a defect in a level shifter can cause performance degradation as well as functional failure. Multi voltage systems normally operate in a wide range of operating voltages. For an example, the minimum and maximum voltages for the Intel PXA270 are 0.8075V and 1.705V[125] respectively. This implies that a circuit must be capable of handling both extremes of voltages without any loss in performance. The input threshold voltage for a given circuit varies with the supply voltage. As we change the supply voltage the minimum required voltage to switch from 'High' to 'Low' or vice versa will change. This is also called the logic interpretation voltage [16] or switching threshold.

To illustrate this, we simulated an inverter in 0.12μm ST Technology. The input and supply voltages were varied to find the exact point where the output changes from 'ON' to 'OFF' and vice versa. The input voltage ranges used in these simulations are the voltage ranges of the PXA270 processor. The logic interpretation voltages for different supply voltages (VDD) exhibit an interesting result. There are clear crossovers between 'ON' and 'OFF' for different values of VDD. Figure 4.10 shows the logic interpretation voltages for different supply voltages. From the Figure, 0.45V can be an 'ON' voltage for VDD of 0.8V but an 'OFF' voltage for a VDD of 1.0V. Similarly at the upper end, 0.64V is an 'ON' voltage for VDD of 1.6V and an 'OFF' voltage for a VDD of 1.2V. These simulation results give some indications of the possible effects of a defective level shifter. It implies that a defect on the level shifter can cause a weak 'HIGH' to be misinterpreted as a 'LOW' signal.

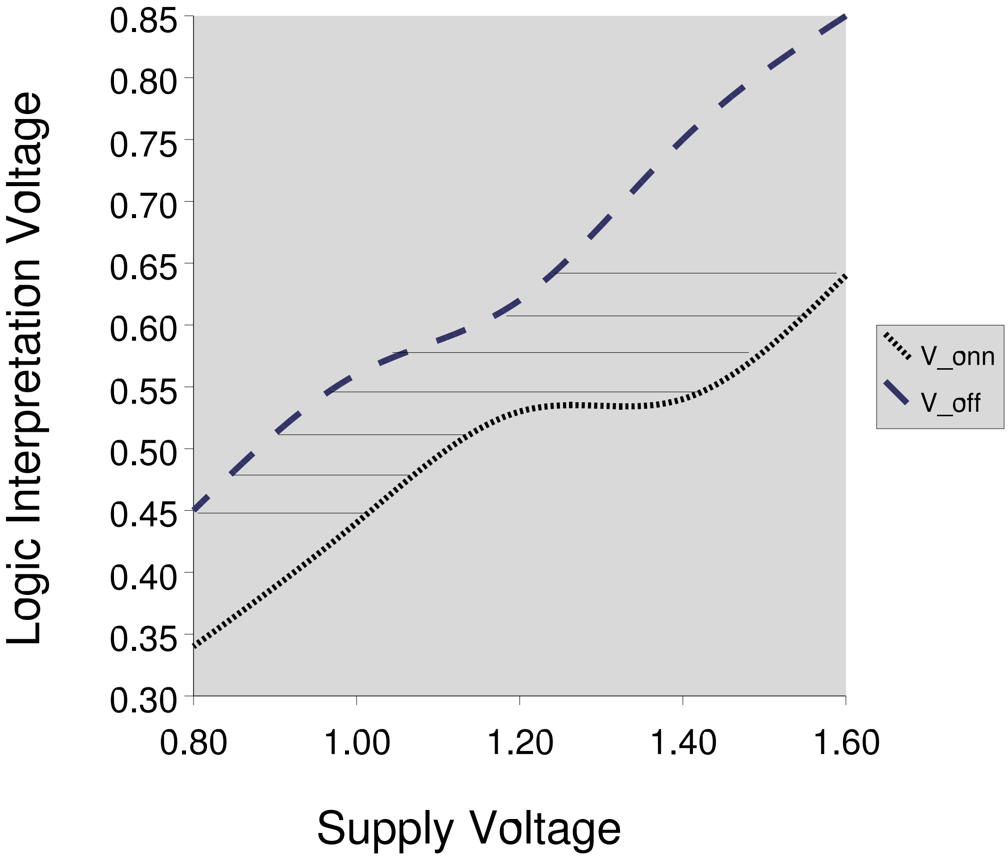


Figure 4.10: Logic interpretation voltages for different power supply

4.5 Summary

This chapter looked at issue in direct relations to testing for multi voltage design. The evolution of multi voltage design from the general concept of low power design was presented. Next, the requirement for acceptable testing methods for multi voltage design was discussed. Very low voltage testing has its strong correlation to testing for multi voltage design. Thus, details of VLV and its issues were discussed. Finally, different types of level shifters and impact of logic interpretation voltage in multi voltage design were shown.

Chapter 5

Delay fault testing For Multi Voltage Designs

5.1 Introduction

During normal operation, a Multi Voltage Design system can run at several different Voltage and/or Frequency settings. It is therefore necessary to ensure that the system will function correctly at each possible Voltage/Frequency setting. Research on very low voltage (VLV) testing [4], [110], [109] has shown that while some faults cannot be observed at the nominal power supply voltage, they become apparent at different operating conditions, such as lower supply voltage. This suggests that traditional test methodologies, assuming a fixed/nominal power supply voltage and clock frequency, may not guarantee an acceptable fault coverage for Multi Voltage Designs.

The aim of this work is to recommend optimal supply voltage setting for delay fault testing in Multi Voltage Design. It attempts to answer this question: Can we test all the defects at minimum set of voltages and achieve maximum fault coverage? Two different defects, resistive opens and bridging faults were used in this case study. Delay characteristics are studied in relation to varying supply voltage settings.

At first, we have looked at conditions that can impact the test results in Section 5.2. Next, general fault simulation setup was given in Section 5.3. This was then followed up by specific settings and analysis for resistive open in Section 5.4. Section 5.5 discusses simulation setup and results for bridging fault. The last chapter summarises the chapter by giving overall observation for work done within this chapter.

5.2 On Test Condition for Optimal Testing

In general there are three main factors that will impact the quality of testing [126], [110], [117], [127] and [31]. Test quality is measured by taking the probability of all detectable faults detected by the fault model. These are the speed of the test, the temperature and the supply voltage. There are also other factors such as process variations [111], power supply noise [128], [129] which can impact the test quality. However we have focused our attention to the conditions that the test engineers can have their influence on. The process variations and power supply noise are external factors that cannot be directly controlled by the test engineers.

In the next sections, we analyse the importance of each factor on test quality. For the first factor, we discuss how the speed of the circuit under test impacts the test quality. The impact of the temperature and the significance in achieving better fault coverage is discussed next. Finally, we discuss the dependency of supply voltage on CMOS propagation delay.

5.2.1 Impact of the test speed

An understanding of the delay-voltage relationship of CMOS devices is required to determine the test speed. The relationship between the test speed and the supply voltage depends on how the delay of the critical path of a circuit changes with the supply voltage. Horowitz et al [130] showed that the delay-voltage relationship of a CMOS gate is predictable. Various sizes of circuits were used to show that CMOS circuits with the same process technology have similar speed-voltage scaling ratios. The delay-voltage scaling ratio is the ratio between the propagation delay at any voltage and that of nominal voltage. The maximum deviation is within 15%. However in [113], Wagner and McCluskey showed that unlike the gate delay, the interconnect delay of an integrated circuit is independent of supply voltage. Due to the difference between the delay-voltage scaling ratio of the CMOS gate delay and the interconnection delays, the critical paths may be different at different supply voltages. This condition exists when there are other paths whose delays are shorter but similar to those of the critical paths at normal operating voltages.

Chang and McCluskey [110] suggest two different methods to determine the test speed. The first method is by using a constant scaling factor. This uses the pre-characterised delay-voltage scaling ratio of a basic CMOS gate. The test speed determined from this method guarantees that the test will not fail good circuits. This is true since the delay-voltage scaling ratio of the interconnection delay is replaced by that of a CMOS gate. Since the interconnection delay is scaled too, the test speed will be slower than any clock rates for possible new critical paths.

However, since the interconnects and CMOS gates have different delay-voltage scaling ratios, the determined test speed may not be the optimum test speed. Even though the method will not fail any good circuits, the test may miss some of the defects that cause timing failures. In the second method, proposed by [110], the circuit needs to be analysed to improve the flaw coverage of timing failures.

New critical paths as well as the test clock frequency at low voltage were found for the circuit under test. [110] has also given the equations to be used to find the critical path delays at different voltages.

The delay-voltage relationship is more critical in a static test since the impact of the delay cannot be observed while running the test. In a dynamic test, the relationship needs to be understood to determine the upper limit of the test speed. The lower limit of the test speed is determined by the size of delay that needs to be captured.

Generally, speed-binning is used as a way to detect the circuit that fails the required speed [131]. In the case of speed-binning not being used, the circuit is normally tested to work at the required speed with the worst process and working conditions. However, the objective of delay fault test (which is to detect faults that cause the circuit to malfunction) will not be achieved. This is due to the fact that even though the defect causes extra delay, the delay is not large enough to cause the circuit to fail at the tested speed.

The argument that the circuit will still work even with the existence of defects has to be ruled out for the reason of reliability [109]. These defective chips are vulnerable to changes in the operating environment and a different operating condition may increase the effect of the flaw. This can cause chip malfunction. In addition, the defect might cause malfunctions when the defect site is crossed by another path. Adaptive delay-fault testing has been recommended as a way to handle these conditions [132]. Adaptive delay-fault testing claims to detect small delay faults. The method is based on grouping conventional delay-fault patterns into sets of almost equal-length paths. Since the path length distribution has been narrowed, the probability of small delay faults undetected due to masking by longer paths will be reduced. The limitation of this method is that the paths have to be hazard free.

Recent experimental work on speed of test [133] has shown that the quality of the applied test sets have more impact on the fault coverage than speed of the test. In their work they have also shown that when Very Low Voltage testing is applied at characterised speed, the fault coverage is almost 100%.

Simulation, in this chapter, conducted at 250Mhz. 200MHz to 250Mhz, is the common speed used for stuck-at-fault test in an ATE [118]. By using the speed of ATE, we can ensure that we are only measuring delay that is causing dynamic faults. Once the value of the fault exceeds the 250MHz limit, we can assume that the fault will be detected by means of static-fault-tests.

The only difference that the speed of the test will make for our testing methodology is the value of the actual delay. However, the absolute value of the delay is immaterial as we are only concerned with the probability of the defect being detected. This is achieved by studying the delay ratio which is the ratio between the delay caused by the defect and delay from the fault free circuit.

5.2.2 Impact of temperature on test quality

The impact of temperature on defect coverage has been studied in [73], [3], [134] and [2]. Cold temperatures have been reported to improve the detectability for certain types of defects such as opens since the defect free silicon is faster by 0.1-0.2%/K while the defective paths generally become slower . The increase of delay in the defective paths is due to the increase of resistance from the opens. [73] studied a path selection technique for path delay fault test generation that takes into account possible variations in operating conditions such as temperature and voltage. Analysis in [73] suggested that using all corner cases will improve the detectability.

The effectiveness of low-temperature testing has been demonstrated for three real defect classes at Intel in [134]. All the three defects which were found through Failure Analysis(FA) required testing at low temperatures and showed a significant difference in results when tested over the temperature range. The impact of low voltage testing, low temperature testing and the combination of them were presented in [3] and [2]. The fault coverage improvement of low temperature testing is limited to hard defects only. Engelke et. al [2] has also reported that in comparing the performance of the combined low voltage and low voltage testing and low voltage alone, the relatively high cost of temperature control does not appear to be justified for detecting the hard defects.

As varying temperature has limited advantage, as well as, not very cost effective , room temperature has been assumed for fault simulations throughout our studies. Another reason for assuming a single temperature is that the increase of temperature will only indirectly affect the delay detected through the increased resistance value. Therefore, temperature impact is considered as just a modification of the resistance value.

5.2.3 Voltage dependance of CMOS Propagation Delay

Due to nonlinear behaviour during a transition, it is difficult to find a closed-form analytical solution for the propagation delay of a CMOS gate. [135], [40], [136] and [137] provide equations to estimate the propagation delay of a CMOS gate. Anantha et al [137] show a first order estimation of the relationship between the propagation delay T_d and the supply voltage of an inverter. We have used this equation, which is shown as Equation 5.1 since the relationship can be observed more directly.

$$T_d = \frac{C_L Vdd}{\mu Cox(W/L)(Vdd - Vt)^2} \tag{5.1}$$

where T_d is the propagation delay, C_L is the loading capacitance, μ is the mobility of electrons, Cox is the oxide capacitance, W and L are the width and length of the transistor respectively, Vdd is the supply voltage and Vt is the threshold voltage. We have used high-speed transistors which have threshold voltage of 380mV and 390mV for NMOS and PMOS respectively.

Equation 5.1 can further be reduced to the Equation 5.2.

$$T_d = K\frac{Vdd}{(Vdd - Vt)^2} \tag{5.2}$$

Recent work by Bota et al [111] has used similar equation as in Equation 5.2 by replacing the square value as α with alpha ranging from 1 to 2. This further enhances the argument of using the equation.

The equivalent delay equation with constant K shows that the delays are mainly dependent on the value of supply voltage, Vdd. In order to verify the accuracy of the equation, the simulated results were compared with the results obtained through calculations. Simulations were conducted on a buffered inverter, as shown in Figure 5.1. The inverter under test is G2. G1 and G2 are input and output buffers respectively. Falling delays were measured between the input and output of inverter G2. Input and output buffers are required in order to achieve more realistic measurements.

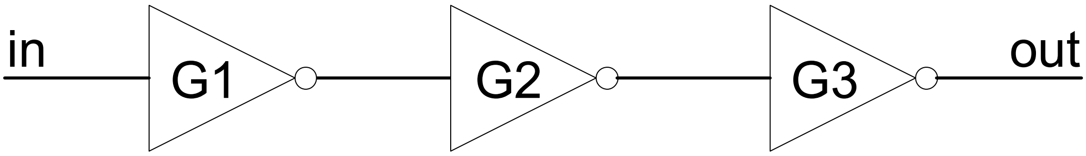


FIGURE 5.1: Simulation setup for propagation delay measurement

The simulation results were compared to the results obtained from Equation 5.2.

Figure 5.2 shows the delay response from simulation as well as response calculated by using Equation 5.2. It is important to point out that the region of our interest

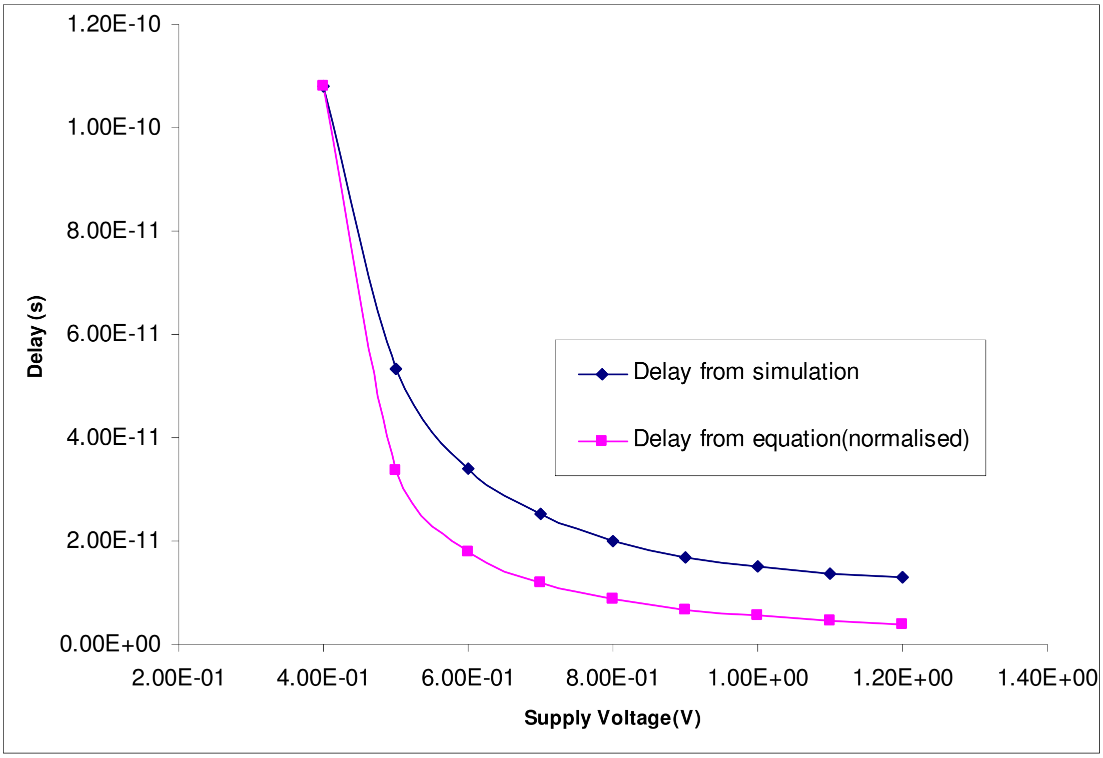


FIGURE 5.2: Gate delay comparison between simulation and Equation 5.2

is between the supply voltage of 0.80V to 1.20V. Even though the actual values between measured and simulated have large difference, the ratio is consistent. The consistent ratio proof that Equation 5.2is good approximation to the actual simulation results.

It also shows that the propagation delay increases significantly at lower voltages than the increment at the nominal voltage.Another observation is that the propagation delay of a CMOS circuit increases monotonically as the supply voltage decreases from normal operating voltage to a value closer to the threshold voltage.

In the next section, the delay of a CMOS circuit when the input signal is degraded is analysed.

5.2.3.1 CMOS Propagation Delay for Degraded Signal

A degraded signal is a signal with less than the full strength of the actual signal. The input signal to a gate will be decreased to a degraded value of the supply voltage when a fault exists in the circuit [40], [138]. The input signal will be degraded from Vdd to Vdd/a as shown in Figure 5.3. It is important to observe that the supply voltage for the gate is at nominal voltage of Vdd. Only the input signals were degraded to Vdd/a. We labelled the gate with the degraded signal as DG. The solid lines represent the input and output signals of the DG gate and the dashed line represents the same information for the fault free circuit.

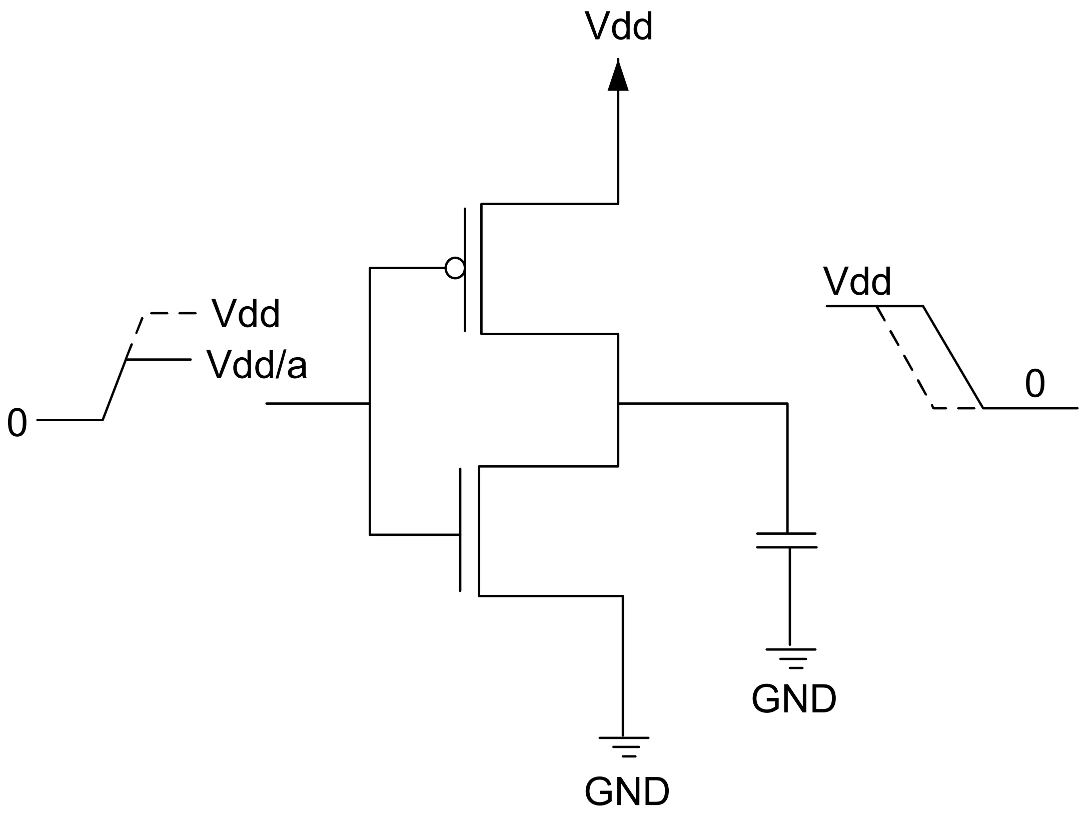


FIGURE 5.3: Effect of degraded signal

From Equation 5.2, the propagation delay for the DG inverter with nominal supply voltage Vdd can be approximated as in Equation 5.3, where $\boldsymbol{a}$ is a number greater than one.

$$T_{DG} = K\frac{Vdd}{(Vdd/a - Vt)^2} = aK\frac{Vdd/a}{(Vdd/a - Vt)^2} \tag{5.3}$$

Equation 5.3 shows that the propagation delay for a gate with degraded signal, DG, at a nominal supply Vdd can be approximated by the propagation delay of the fault free inverter operating with a supply voltage of Vdd/a, (as in Equation 5.2), multiplied by ***a*** . Simulations were conducted to verify Equation 5.3 and the results shown in Table 5.1. In Table 5.1, the nominal supply voltage is 1.2V. The actual delays from simulation obtained with different supply voltages are shown in the forth column. These are fault free delays. The last two columns show the delays with degraded input signals and supply voltage of 1.2V. As we can observe, the calculated delay obtained using Equation 5.3 gives a good approximation to the actual simulation results for ***a*** less than 2.0. However, as the value of ***a*** increases and the degraded signal is closer to the threshold voltage, the Equation does not hold. When the degraded signal is less than 500mV, the results show a stuck-at-1 behaviour.

Table 5.1: Comparison of Simulation Result and Equation 5.3

Vdd (V)	**Vdd/a (V)**	**a**	**Delay at Vdd/a as supply voltage (s)**	**Delay with degraded signal**	
				Calculation	**Simulation**
1.20	1.20	1.00	1.28E-11	1.28E-11	1.28E-11
1.20	1.10	1.09	1.39E-11	1.52E-11	1.49E-11
1.20	1.00	1.20	1.52E-11	1.82E-11	1.78E-11
1.20	0.90	1.33	1.70E-11	2.26E-11	2.22E-11
1.20	0.80	1.50	2.01E-11	3.02E-11	3.00E-11
1.20	0.70	1.71	2.52E-11	4.32E-11	4.65E-11
1.20	0.60	2.00	3.42E-11	6.84E-11	1.09E-10
1.20	0.50	2.40	5.34E-11	1.28E-10	S-A-1
1.20	0.40	3.00	1.08E-10	3.24E-10	S-A-1

Even though Equation 5.3 does not hold when the degraded signal is less than half of the supply voltage, it still can be used to demonstrate the general behaviour of the gate with a degraded signal. In general, it shows that the delay of a degraded signal will be more than delay of a gate with a supply voltage equivalent to degraded signal. The simulation results also show that as the degraded signal goes below a certain value, the circuit will be malfunctioning.

5.3 Simulation Setup

The impact of specific defect locations is investigated by the use of SPICE simulations of ST 0.12μm technology using Cadence Virtuoso Spectre Circuit Simulator. SPICE (Simulation Program with Integrated Circuit Emphasis) is a general purpose analog electronic circuit simulator. It is a powerful transistor level simulator that is commonly used to check the integrity of circuit designs and to predict circuit behaviour. The spice netlist generated for the multiplier circuit is shown in Appendix B.

Defects were injected at different locations to observe the fault impact. The values of resistance were varied from the nominal to total open for resistive open circuits. For resistive shorts, the resistances were varied from short circuit conditions to fault free conditions. The clock period of the signal was set to 4 ns which gave the frequency of operations at 250MHz. Transition fault model was used throughout the simulations.

We have used a 4 x 4 unsigned multiplier for our simulations [139], as shown in Figure 5.4. We have selected this circuit since it gives us many different options for a path to be sensitised for a single error. The multiplier is built using 12 full adder cells. The full adder circuit is shown in Figure 5.5. Figure 5.6 describes how the multiplication is done using the circuits shown in Figure 5.4 and Figure 5.5. The summands $P_k = Y_i X_j$ are generated in parallel with AND gates and then added in arrays of 1-bit full-adders. As far as our simulations are concerned, the primary input in the multiplier circuits are the 16 combinations of Y and X. The primary outputs are eight bit values noted as P_0 to P_7. We have used the Single Fault Assumption where there will be only a single fault at one point of time.

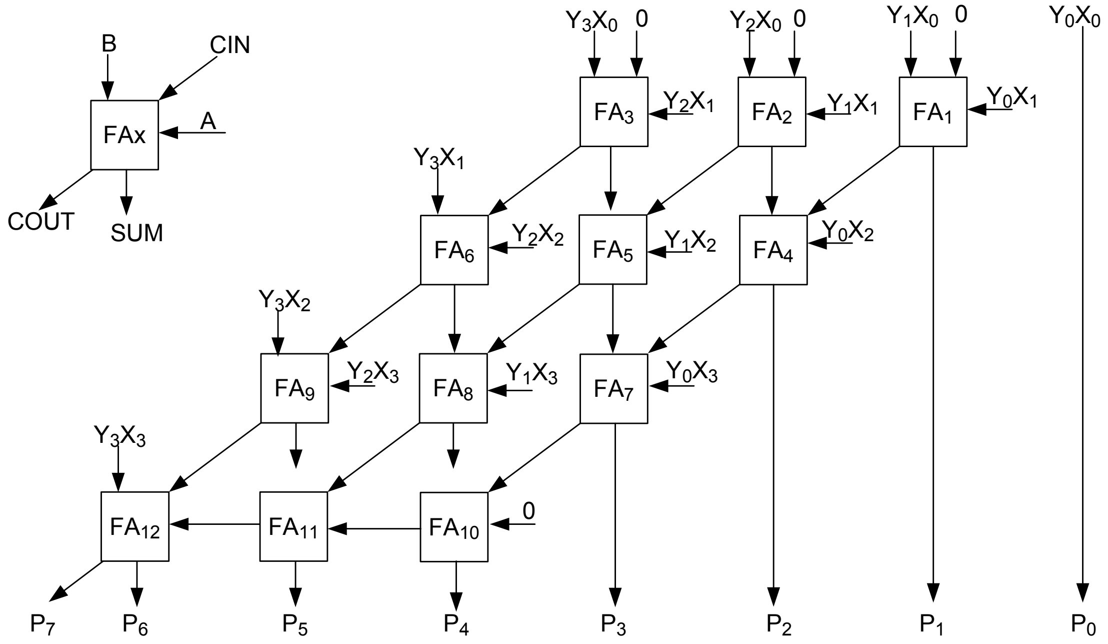


FIGURE 5.4: Multiplier Array

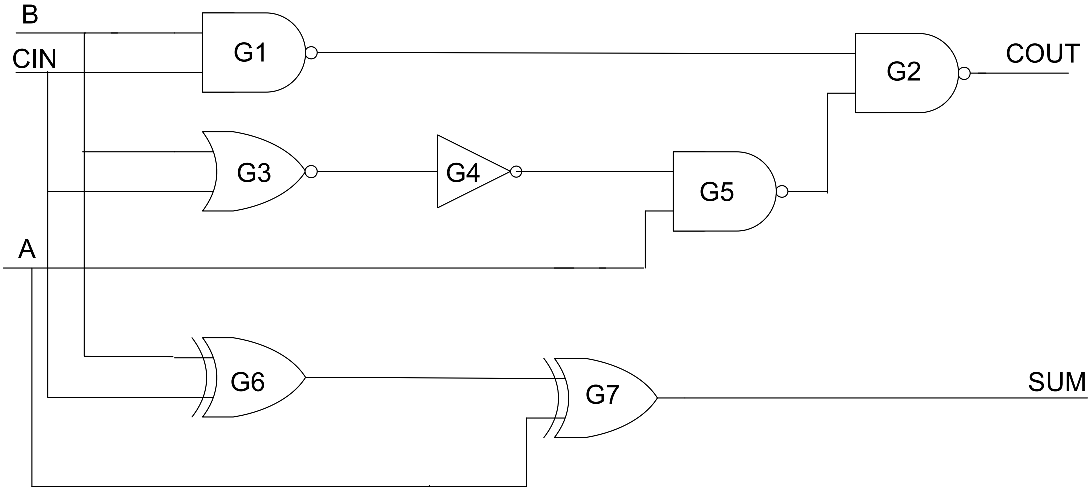


FIGURE 5.5: Full Adder Circuit

5.4 Simulation for Resistive Open

An open defect, which traditionally means an unconnected node, can cause a logical error. This is also known as a hard open. Another class of open defects is

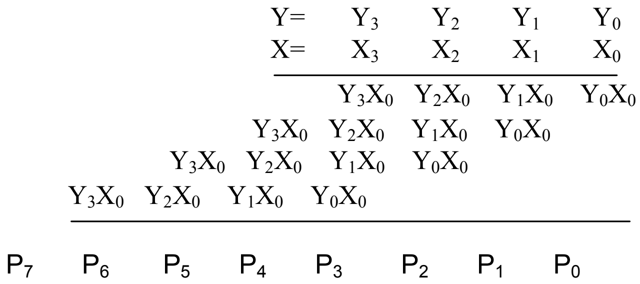


Figure 5.6: Partial Products of a 4 x 4 unsigned integer multiplication

resistive opens. These can cause an additional delay in a transition as well as a logical error. The behaviour of the circuit will depend mainly on the value of the resistance as well as the supply voltage [65].

The faults were injected in the full adder circuit and the fault effects were propagated through the circuit until they reached the primary output. Since the transition fault model is used, both rising and falling time were observed. At first, the delay with resistance values of 0 Ω was observed. This is the fault free value and will be used as a reference for comparison with the faulty conditions. The value of the resistance is then increased until the circuit shows stuck-at-fault behaviour.

Open defects were injected at a number of places of the multiplier circuit. We had run extensive simulation at a large number of places in the circuit. At initial stages the layout was extracted using Cadence Virtuoso tool. Only possible open places were selected. This is important in order to rule-out the locations which are physically unlikely to occur in the silicon. After observing the results from a larger set of defect locations, the defect locations were grouped into four areas. The grouping of the defects location are done by observing the fault behaviour. Faults having similar responses were grouped into same class of defect location. Fault A represents defect at interconnect. The main cause of this class of open is not completely filled via [126]. Fault B represent defects at the primary input of the subcircuit. Fault C and D represent defects at critical path of the circuit.

Fault C and D have the same defect locations, however, the path sensitised are different.

The locations of the faults are shown in Figure 5.7. The faults location which are labelled with resistor R_a, R_b represent Fault A, Fault B. R_c, d represent both Fault C and Fault D.

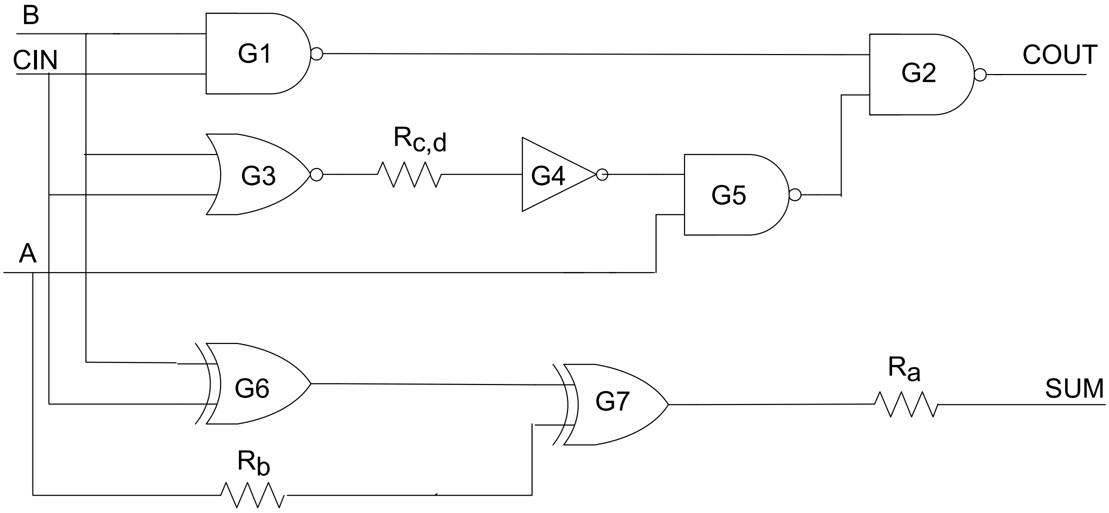


FIGURE 5.7: Full Adder Circuit

5.4.1 Fault A

Fault A is a resistive open fault located after the XOR gate G7 in Figure 5.7. The location is marked as R_a in Figure 5.7. The defect was injected in the full adder circuit FA_2. This fault represents an open at the interconnect between FA_2 and FA_4. The path that has been sensitised is from the primary input Y_1X_1 to primary output P_2. To enable this path, the input at Y_1X_1 has to be toggled between HIGH and LOW input signals and rest of the inputs have to be set at non controlling values, see section 2.6.3.3 in Chapter 2 for detail on non-controlling value (NVC). The simulations were repeated for three values of supply voltage : 0.8V, 1.0V and 1.2V.

Table 5.2: Path Delay and Path Delay Ratio for Fault A

RES **(Ω)**	**Voltage** **(V)**	**Path Delay:** **Rise (s)**	**Ratio** **Rise**	**Path Delay:** **Fall (s)**	**Ratio** **Fall**
0.00	1.20	1.74E-10	1.00	1.44E-10	1.00
1.00k	1.20	1.86E-10	1.07	1.57E-10	1.09
10.00k	1.20	3.06E-10	1.76	2.95E-10	2.05
50.00k	1.20	8.05E-10	4.64	8.11E-10	5.62
75.00k	1.20	1.09E-09	6.28	1.11E-09	7.68
100.00k	1.20	1.33E-09	7.65	1.36E-09	9.46
200.00k	1.20	1.91E-09	10.98	2.05E-09	14.20
300.00k	1.20	2.24E-09	12.88	2.45E-09	16.99
400.00k	1.20	SF	-	SF	-
0.00	1.00	2.10E-10	1.00	1.74E-10	1.00
1.00k	1.00	2.21E-10	1.06	1.86E-10	1.07
10.00k	1.00	3.43E-10	1.64	3.23E-10	1.86
50.00k	1.00	8.49E-10	4.05	8.44E-10	4.85
75.00k	1.00	1.14E-09	5.43	1.14E-09	6.54
100.00k	1.00	1.38E-09	6.60	1.39E-09	8.02
200.00k	1.00	1.99E-09	9.48	2.09E-09	12.00
300.00k	1.00	2.34E-09	11.17	2.50E-09	14.38
400.00k	1.00	SF	-	SF	-
0.00	0.80	2.84E-10	1.00	2.35E-10	1.00
1.00k	0.80	2.95E-10	1.04	2.47E-10	1.05
10.00k	0.80	4.16E-10	1.47	3.80E-10	1.62
50.00k	0.80	9.39E-10	3.31	9.32E-10	3.96
75.00k	0.80	1.23E-09	4.34	1.22E-09	5.20
100.00k	0.80	1.48E-09	5.23	1.47E-09	6.24
200.00k	0.80	2.13E-09	7.51	2.17E-09	9.21
300.00k	0.80	2.51E-09	8.87	2.60E-09	11.04
400.00k	0.80	SF	-	SF	-

Table 5.2 shows the detailed results for fault A. The value of the resistive open is shown as RES. The rising path delay which is measured from the primary input Y_1X_1 to the primary output P_2 is labelled as Path Delay:Rise and the values are given in seconds. The path delay ratio for rising transitions is shown as Ratio:Rise. Similarly, the next two columns show the details for falling transition delays and its ratios respectively. Delay ratios are the delay of the circuit compared to the delay for the fault free case i.e. when the resistance value is equal to 0.

A number of key observations can be derived from Table 5.2.

- **The delay of the fault free circuit increases with the reduction of supply voltage**
- **The delay of the faulty circuit increases with the increase of resistance**
- **The delay ratio of the faulty circuit reduces with the reduction of supply voltage**
- **Increasing the resistance above a certain value will result in stuck-at-fault behaviour**

The absolute delay of the circuit increases with the resistance for both rise and fall cases. The delay observed for the fault free case has the nominal delay due to propagation and transition delay. As the supply voltage is reduced, the absolute delay increases. For an example, at 1.2V the value of fault free delay is 1.74E-10s and 1.44E-10s for the rising and falling delay respectively. As the supply voltage is reduced to 0.8V, the fault free values for the rising and falling delays are 2.84E-10s and 2.35E-10s . For these fault free cases, the absolute increment in the delay for the rising case is 63.22% and for the falling case is 63.19%.

As we move to faulty conditions, where the resistance value is no longer zero, the delay increases for all three values of supply voltage. The increased delay is due to the defects injected. However the increment of the delay for all three cases are not same. For readability purposes, results for fault free and one value of resistance, 1.04E+04Ω, for Fault A is tabulated in Table 5.3. Due to similarity of results, only falling delay cases is shown in the table.

From Table 5.3, the falling path delay ratio for 1.2V is 2.05. In other words, the delay has increased by 105% from the fault free case. For the same fault, the

Table 5.3: Result for Fault A with resistive open of 1.04E+04 Ω

RES (Ω)	**Voltage (V)**	**Path Delay: Fall (s)**	**Ratio Fall**
0.00	1.20	1.44E-10	1.00
10.00k	1.20	2.95E-10	2.05
0.00	1.00	1.74E-10	1.00
10.00k	1.00	3.23E-10	1.86
0.00	0.80	2.35E-10	1.00
10.00k	0.80	3.80E-10	1.62

ratios for 1.0V and 0.8V are 1.86% and 1.62% respectively. This means, the delay has increased by 86% for 1.0V and by 62% for 0.8V. This pattern of results with reducing ratio were observed for the rising cases as well.

If the test was conducted in these three voltages, the fault would most likely be detected at higher voltage. The statement is made from the fact that the highest voltage gave the largest delay ratio.

To emphasise the importance of the likelihood of detection with regard to fault coverage, we have rerun the simulations for Fault A for resistance value from 0 Ω to 10000 Ω with 1000 Ω steps. Results for resistance value from 0 Ω to 5000 Ω are shown in Table 5.4.

An Automated Test Equipment (ATE) normally has a margin of 15 to 25% due to non ideal power supply of the ATE [118]. The margin is required to compensate the voltage loss at the high inductance power cable from the tester to the Circuit Under Test (CUT). This means an extra 15 to 25% of delay margin is added on top of the actual designed clock speed [118]. In practice, a delay ratio of 15 to 25% will not be detected as it is considered as an extra delay margin.

If an assumption is made that the tester can detect a minimum 18% difference to fault free case, some remark can be made from Table 5.4. In this case, a resistive open with 2000 Ω will be detected at 1.2V since it has an increased delay of 19%.

However the delay fault will not be detected at 1.0V and 0.8V. This is because the 2000 Ω resistance has only delay increment of 15% and 10% at 1.0V and 0.8V respectively. The value of resistance that will be undetected by this assumption is grayed in Table 5.4.

The result from Table 5.2 and Table 5.4 together with our simple assumption shows the importance of using appropriate voltage settings for detection of a defect. An undetected fault will result in lower fault coverage.

TABLE 5.4: Path Delay and Path Delay Ratio for Fault A with lower resolution

RES Ω	**Voltage** V	**Path Delay: Rise (s)**	**Ratio Rise**	**Path Delay: Fall (s)**	**Ratio Fall**
0.00	1.20	1.74E-10	1.00	1.44E-10	1.00
1.00k	1.20	1.86E-10	1.07	1.57E-10	1.09
2.00k	1.20	1.98E-10	1.14	1.71E-10	1.19
3.00k	1.20	2.12E-10	1.22	1.87E-10	1.29
4.00k	1.20	2.25E-10	1.30	2.02E-10	1.40
5.00k	1.20	2.39E-10	1.38	2.18E-10	1.51
0.00	1.00	2.10E-10	1.00	1.74E-10	1.00
1.00k	1.00	2.21E-10	1.06	1.86E-10	1.07
2.00k	1.00	2.34E-10	1.11	1.99E-10	1.15
3.00k	1.00	2.47E-10	1.18	2.14E-10	1.23
4.00k	1.00	2.61E-10	1.24	2.30E-10	1.32
5.00k	1.00	2.74E-10	1.31	2.45E-10	1.41
0.00	0.80	2.84E-10	1.00	2.35E-10	1.00
1.00k	0.80	2.95E-10	1.04	2.47E-10	1.05
2.00k	0.80	3.07E-10	1.08	2.60E-10	1.10
3.00k	0.80	3.19E-10	1.13	2.74E-10	1.16
4.00k	0.80	3.33E-10	1.17	2.88E-10	1.23
5.00k	0.80	3.46E-10	1.22	3.03E-10	1.29

Finally, it can be observed from Table 5.2 that increasing the resistance above a certain value i.e the critical resistance will result in a logical error. For all cases of supply voltages, as the resistance is increased to 400K Ω, the defect manifests itself as a stuck-at-fault. This is noted as 'SF' in Table 5.2. We have rerun the simulation to find the critical values that cause stuck-at-fault behaviour.

From the simulation results, we found that for a supply voltage of 1.2V, the critical resistance is between 340K and 350K Ω and for 0.8V supply voltage, the value is between 330K and 340K Ω. This is in line with our general findings that the absolute impact is higher at lower voltage. The general relationship between critical resistance and supply voltage for resistive open is shown in Figure 5.8. As the supply voltage increases, the value of the critical resistance also increases. Figure 5.8 is not drawn to scale and was simply used to show the relation between the critical resistance and supply voltage.

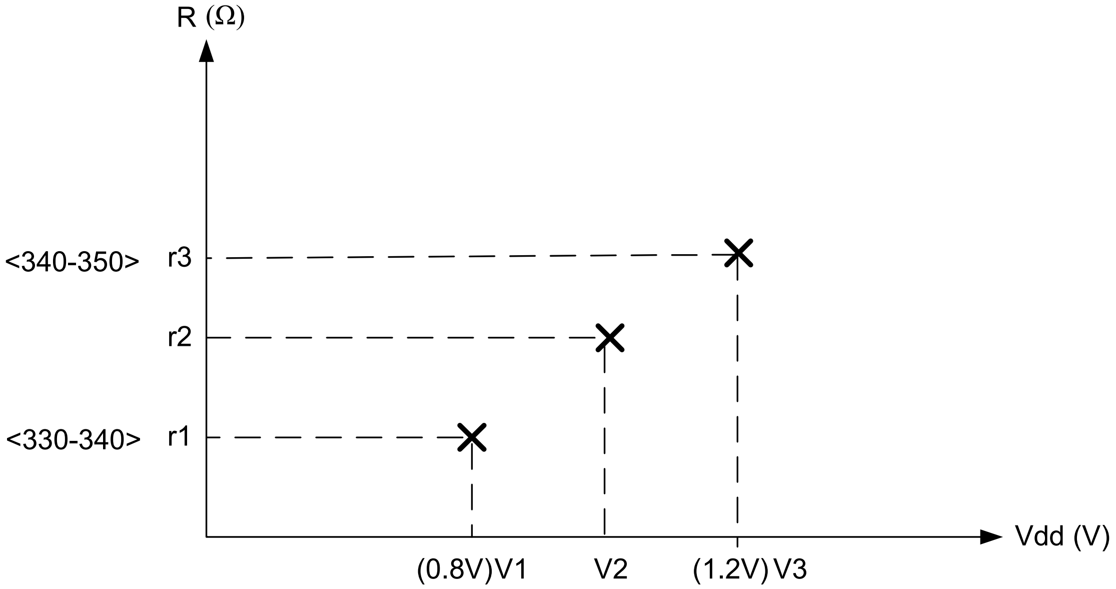


FIGURE 5.8: Voltage dependency of critical resistance for resistive open

5.4.2 Fault B,C and D

Three more resistive open faults cases were simulated and their responses were observed. They are labelled as Faults B, C and D. Fault B is selected such that the sensitised path is similar to that of Fault A. However, the resistive open defect was injected at location R_b in Figure 5.7. Faults C and D were selected such that the fault locations are the same but the path sensitised to observe the fault effects are different. Table 5.5 shows the detail of the faults and signal propagation path for Faults A, B, C and D.

Detailed results of the fault simulation of B, C and D are shown in Table 5.6,, 5.7 and 5.8 respectively.

Results from Fault B show a similar pattern as for Fault A. Furthermore, it can be observed that the actual delay values are different for both cases even though the fault were injected on the same adder cell. In addition, the critical resistance for Fault B is a value between 400K Ω and 500K Ω. This is different to Fault A where the critical resistance was between 300K Ω and 400K Ω. In short, the actual location of the defect will have different results even though the paths sensitised are the same.

In contrast to Fault B, Faults C and D have different path sensitised for the same defect as shown in Table 5.5. Fault C has a higher fault ratio than fault D even though the absolute delays are higher in fault D. This can be explained since fault D has a longer sensitised path than fault C. However, the extra delays caused by the defect are same. For example, consider the falling delay for the case of open resistance of 75k Ω at 1.2V supply voltage. From Table 5.7, for fault C, the extra delay due to the defect is 7.45E-10 - 3.80E-10 = 3.65E-10. Similarly, from Table 5.8, for fault D, the extra delay due the defect is 8.71E-10 - 5.06E-10= 3.65E-10. Even though the defect causes the same extra delay, the delay ratios for both cases are 1.96 and 1.72. This signifies that the fault C will be most likely detected since the delay ratio is higher.

TABLE 5.5: Signal propagating path

Fault	Fault Site	Signal Processing Path
A	Open between FA_2 and FA_4	$Y_1X_1 \rightarrow FA_2 \rightarrow FA_5 \rightarrow FA_7 \rightarrow P_3$
B	Open between FA_2 and FA_4	$Y_1X_1 \rightarrow FA_2 \rightarrow FA_5 \rightarrow FA_7 \rightarrow P_3$
C	Open between G3 and G4 in FA_9	$Y_2X_1 \rightarrow FA_1 \rightarrow FA_5 \rightarrow FA_9 \rightarrow FA_{12} \rightarrow P_4$
D	Open between G3 and G4 in FA_9	$Y_2X_1 \rightarrow FA_3 \rightarrow FA_5 \rightarrow FA_7 \rightarrow FA_{10} \rightarrow FA_{11} \rightarrow FA_{12} \rightarrow P_7$

Table 5.6: Path Delay and Path Delay Ratio for Fault B

RES Ω	**Voltage** **V**	**Path Delay:** **Rise (s)**	**Ratio** **Rise**	**Path Delay:** **Fall (s)**	**Ratio** **Fall**
0.00	1.20	1.74E-10	1.00	1.44E-10	1.00
10.00k	1.20	2.63E-10	1.51	2.38E-10	1.65
75.00k	1.20	8.01E-10	4.61	8.08E-10	5.60
400.00k	1.20	1.81E-09	10.42	2.21E-09	15.30
500.00k	1.20	SF	-	SF	-
0.00	1.00	2.10E-10	1.00	1.74E-10	1.00
10.00k	1.00	3.01E-10	1.44	2.67E-10	1.54
75.00k	1.00	8.44E-10	4.03	8.29E-10	4.77
400.00k	1.00	1.91E-09	9.09	2.22E-09	12.75
500.00k	1.00	SF	-	SF	-
0.00	0.80	2.84E-10	1.00	2.35E-10	1.00
10.00k	0.80	3.82E-10	1.35	3.34E-10	1.42
75.00k	0.80	9.38E-10	3.31	8.92E-10	3.79
400.00k	0.80	2.08E-09	7.32	2.28E-09	9.71
500.00k	0.80	SF	-	SF	-

Table 5.7: Path Delay and Path Delay Ratio for Fault C

RES Ω	**Voltage** **V**	**Path Delay:** **Rise (s)**	**Ratio** **Rise**	**Path Delay:** **Fall (s)**	**Ratio** **Fall**
0.00	1.20	4.47E-10	1.00	3.80E-10	1.00
10.00k	1.20	4.91E-10	1.10	4.30E-10	1.13
75.00k	1.20	7.95E-10	1.78	7.45E-10	1.96
400.00k	1.20	1.61E-09	3.60	1.17E-09	3.07
750.00k	1.20	1.40E-08	31.37	1.29E-08	34.03
0.00	1.00	5.41E-10	1.00	4.62E-10	1.00
10.00k	1.00	5.84E-10	1.08	5.11E-10	1.11
75.00k	1.00	8.86E-10	1.64	8.30E-10	1.80
400.00k	1.00	1.70E-09	3.15	1.28E-09	2.77
750.00k	1.00	1.41E-08	26.08	1.31E-08	28.25
0.00	0.85	7.35E-10	1.00	6.33E-10	1.00
10.00k	0.85	7.78E-10	1.06	6.82E-10	1.08
75.00k	0.85	1.08E-09	1.47	1.01E-09	1.59
400.00k	0.85	1.91E-09	2.59	1.49E-09	2.36
750.00k	0.85	1.43E-08	19.50	1.33E-08	20.92

TABLE 5.8: Path Delay and Path Delay Ratio for Fault D

RES Ω	**Voltage** **V**	**Path Delay:** **Rise (s)**	**Ratio** **Rise**	**Path Delay:** **Fall (s)**	**Ratio** **Fall**
0.00	1.20	5.45E-10	1.00	5.06E-10	1.00
10k	1.20	5.89E-10	1.08	5.56E-10	1.10
75k	1.20	8.93E-10	1.64	8.71E-10	1.72
400k	1.20	1.71E-09	3.13	1.29E-09	2.56
750k	1.20	1.41E-08	25.89	1.31E-08	25.79
0.00	1.00	6.64E-10	1.00	6.16E-10	1.00
10k	1.00	7.07E-10	1.06	6.65E-10	1.08
75k	1.00	1.01E-09	1.52	9.84E-10	1.60
400k	1.00	1.83E-09	2.75	1.43E-09	2.33
750k	1.00	1.42E-08	21.45	1.32E-08	21.42
0.00	0.85	9.08E-10	1.00	8.46E-10	1.00
10k	0.85	9.51E-10	1.05	8.95E-10	1.06
75k	0.85	1.25E-09	1.38	1.22E-09	1.44
400k	0.85	2.08E-09	2.29	1.71E-09	2.02
750k	0.85	1.45E-08	15.99	1.35E-08	15.91

5.5 Simulation of Resistive Bridging Fault

Shorts between circuit nodes are the predominant type of manufacturing defect [1],[4],[77]. There are two types of shorts: intra-gate shorts between nodes within a logic gate and inter-gate or external shorts between the outputs of different logic gates. Inter-gate shorts or bridging fault or bridging faults account for up to 90% of all shorts. As discussed in Chapter 3, the probability of having zero-ohm bridges is far less than probability of having resistive bridges. A resistive bridging fault is an accurate representation of a resistive short defect.

In our simulations we have looked at three different cases of bridging fault: a bridge between a primary input and gate output, bridge between two gate outputs and bridge between outputs of two gates. The difference between these gates are explained in section 3.1.2 in chapter 3.

In all our simulations, resistive short defects were injected at three different places to observe the behaviour of the circuit. The locations of the defects are shown in

Figure 5.9. For example, the injected fault ,R_f, where the resistive short happens between the output of gate G6 and input A. We can have two faulty conditions from this settings. The first condition is a short to ground when input A is LOW and a short to VDD when the input A is set to HIGH. The condition of a resistive bridging fault will happen when two nodes of the resistors have different driving values i.e (LOW, HIGH) or (HIGH, LOW). Otherwise if both nodes have the same driving value, the injected resistance will not have any impact. Therefore, the correct selection of test vectors is crucial such that the path sensitised can trigger the defect and thus enable the defect to be detected.

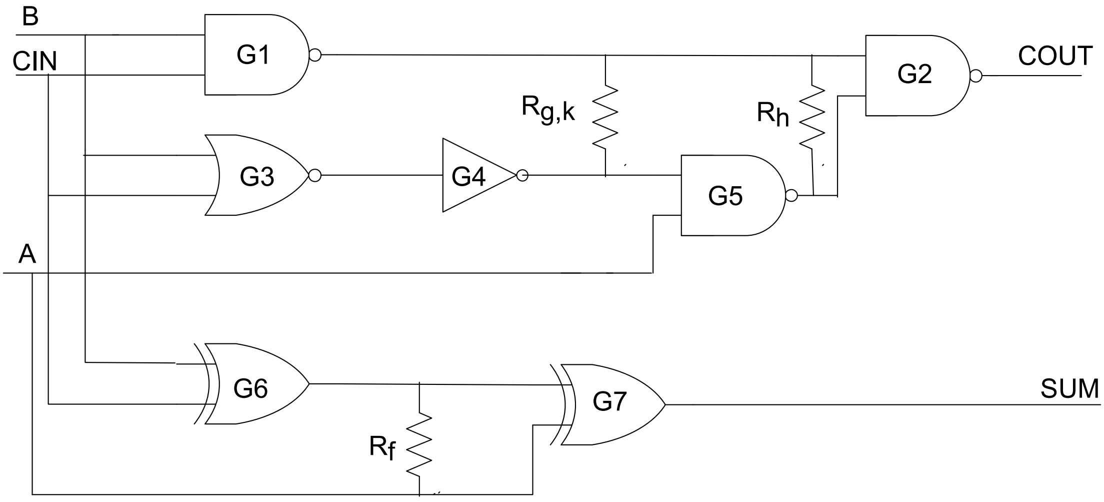


Figure 5.9: Fault locations for resistive bridging fault

5.5.1 Fault F

The location of fault F is shown in Figure 5.9 as R_f. The defective full adder cell was placed at FA_2. This is an example of a bridge between a primary input and a gate output. In this case, the fault will propagate through G7 then through the output SUM before being fed to the next full adder cell FA_4. The path that has been sensitised is from Y_2 X_0 to P_2. By toggling the input signal at primary input Y_2 X_0, the fault was propagated to the primary output P_2. All other inputs were set to non-controlling values to prevent other parts of the circuit from having any impact on the propagated fault effect. The detailed simulation results are shown in Table 5.9. The fault free case for the resistive bridging fault is when the value of the resistance is large enough that it does not have any impact on the circuit. Ideally it will be similar to a circuit in which the resistance does not exist.

TABLE 5.9: Path Delay and Path Delay Ratio for Fault F

RES **Ω**	**Voltage** **V**	**Path Delay:** **Rise (s)**	**Ratio** **Rise**	**Path Delay:** **Fall (s)**	**Ratio** **Fall**
2.00k	1.20	4.39E-10	2.04	1.34E-10	0.75
2.50k	1.20	2.84E-10	1.32	1.55E-10	0.87
3.00k	1.20	2.60E-10	1.21	1.59E-10	0.89
3.50k	1.20	2.49E-10	1.15	1.61E-10	0.90
4.50k	1.20	2.38E-10	1.11	1.64E-10	0.92
6.00k	1.20	2.30E-10	1.07	1.68E-10	0.93
8.00k	1.20	2.26E-10	1.05	1.70E-10	0.95
10.00k	1.20	2.24E-10	1.04	1.71E-10	0.95
20.00k	1.20	2.19E-10	1.02	1.74E-10	0.97
2.00M	1.20	2.15E-10	1.00	1.79E-10	1.00
2.00k	1.00	SF	-	SF	-
2.50k	1.00	4.49E-10	1.73	1.77E-10	0.82
3.00k	1.00	3.46E-10	1.33	1.88E-10	0.87
3.50k	1.00	3.18E-10	1.22	1.92E-10	0.89
4.50k	1.00	2.96E-10	1.14	1.96E-10	0.91
6.00k	1.00	2.84E-10	1.09	2.00E-10	0.93
8.00k	1.00	2.76E-10	1.06	2.03E-10	0.94
10.00k	1.00	2.72E-10	1.05	2.04E-10	0.95
20.00k	1.00	2.66E-10	1.02	2.09E-10	0.97
2.00M	1.00	2.60E-10	1.00	2.16E-10	1.00
2.00k	0.85	SF	-	SF	-
2.50k	0.85	SF	-	SF	-
3.50k	0.85	5.62E-10	1.59	2.50E-10	0.86
4.50k	0.85	4.44E-10	1.26	2.61E-10	0.89
6.00k	0.85	4.04E-10	1.15	2.67E-10	0.91
8.00k	0.85	3.86E-10	1.09	2.71E-10	0.93
10.00k	0.85	3.77E-10	1.07	2.73E-10	0.93
20.00k	0.85	3.63E-10	1.03	2.80E-10	0.96
2.00M	0.85	3.53E-10	1.00	2.92E-10	1.00

Simulation results show that the result for R = 2M Ω is similar to the case with no resistance. Thus, we have taken the R = 2M Ω as the fault free case and used it as a reference to obtain the delay ratio. Delay ratios for resistive bridging faults are the ratio between the delay with the defect and delay without the defect, i.e a large resistance.

From Table 5.9, it can be observed that for rising transition path delay, the delay decreases as the value of resistance is increased. This is true for all three cases of supply voltage. When the value of the resistance reaches a point, which we note as the critical resistance, the circuit exhibits stuck-at-fault behaviour. It can also be observed that the values of critical resistance are dependent on the supply voltage. It is higher for a lower supply voltage. In this case, for the supply voltage of 1.2V the stuck-at-fault behaviour happens at R = 1900 Ω and as the voltage is reduced, the stuck-at-fault behaviour can be observed at higher value of resistance: R = 2000 Ω for V=1.0V and R = 2500 Ω for V=0.8V.

Also, we can observe that the rising path delay ratio is higher for lower supply voltages. This is in contrast with results for resistive opens. For example, the rising path delay ratio for R = 3500 Ω for 1.2V is 1.15.The ratio is 1.22 and 1.59 for 1.0V and 0.8V respectively. This indicates that the probability of a similar defect being detected at a lower voltage is much higher than if the same defect were tested at a higher voltage. For the same reason explained in section 5.4.1, if it is assumed that the ATE can detect a delay difference of more than 25% of the nominal value, the resistive short defect of 3500 Ω can only be detected at 1.2V. This is due to the fact that the delay ratio has only increased at 15% and 22% for 1.2V and 1.0V respectively. Figure 5.10 exhibits the general behaviour of voltage dependency of critical resistance in resistive short and bridging circuits. r1, r2 and r3 are the values of critical resistance that can be detected at different values of resistance. As the supply voltage Vdd increases, the range of the resistance that can be detected as a logical fault will decrease.

For the case of falling transition delay, the circuit shows an increasing speed up with the reduction of the resistance value. As the resistance is tied to the ground,

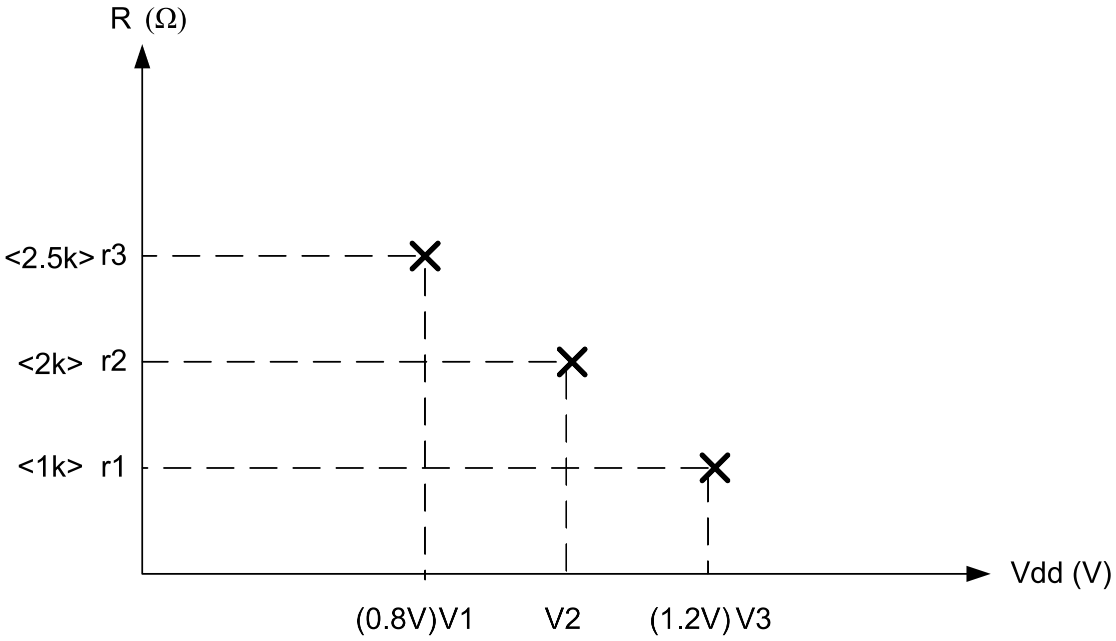


Figure 5.10: Voltage dependency of critical resistance for resistive bridging fault

the signal falls much faster than it would without the fault. Gate G7 will have a stronger driving strength with the reduction of the resistance as the LOW signal will be connected directly to upper input of G7. Even though the speed-up will not cause any malfunction or performance degradation it is very important to detect the speed-up for the reason of reliability.

5.5.2 Fault G,H and K

Three more cases of resistive bridging fault were injected. The first of them, fault G was injected at the location indicated by R_{gk} in Figure 5.9. This is an example of bridge between two gate outputs whereby the bridged nodes feed into different gates. The two gate outputs, i.e from G1 and G4 were bridged and fed to G5 and G2. The faults were injected at full adder cell FA_1. The results show a similar pattern to Fault F. The critical resistances for fault G are 1000 Ω for 1.2V, 1300 Ω for 1.0V and 1700 Ω for 0.8V. These values are slightly lower than the value of the critical resistance for fault F.

The next fault, Fault H is an example of a bridge between the outputs of two gates where the bridged nodes feed into the same gate. The location is indicated by R_h in Figure 5.9.The two gate outputs i.e from G1 and G5 were bridged and fed to the same gate G2. Even though the results show a similar pattern to Faults F and G, there are other interesting observations from the results of Fault H. Contrary to previous results, none of the resistance values at any supply voltage resulted in stuck-at-fault behaviour. Even with zero resistance, the output from bridged gate G2 only has delay values. The waveform of the two inputs to gate G2 and the response at the output of gate G2 is shown in Figure 5.11. The driving strengths of the NAND gates G1 and G5 do not diminish even with zero resistance. The LOW value fall to 0.4V to enable the pull-up transistors in the NAND gate to turn ON which resulted in the correct value at the output of gate G2.

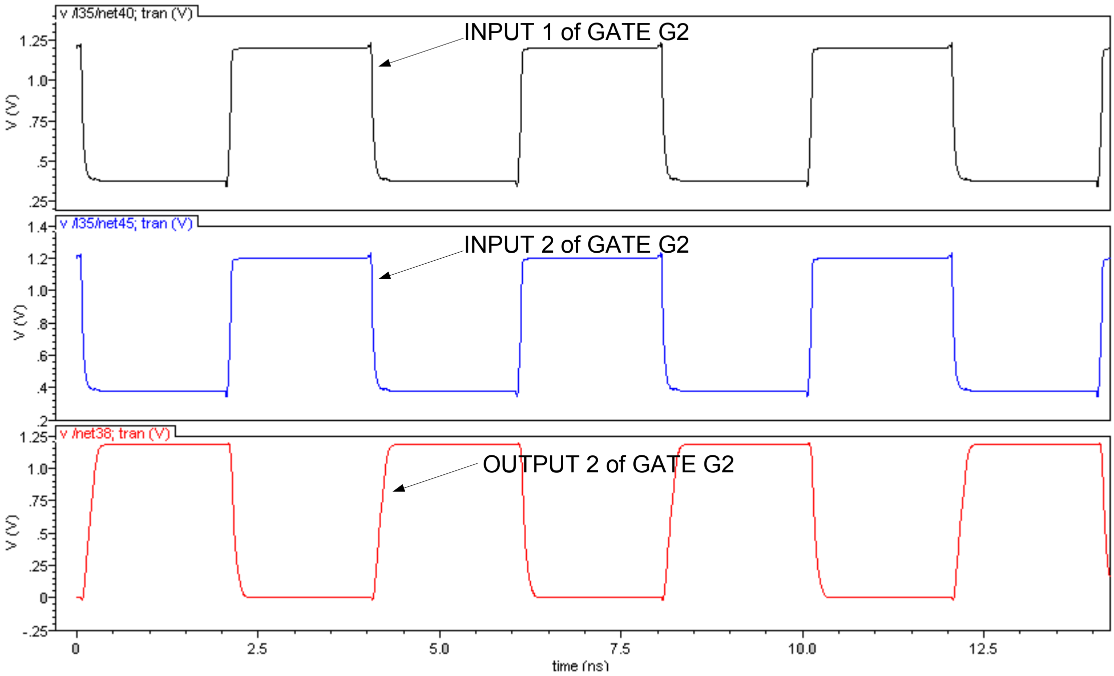


FIGURE 5.11: Response of bridging fault of 0 Ω for fault H

The last fault case simulated is Fault K. In this case, the objective is to study the impact when we have a longer path sensitised. Faults were injected at location $R_{g,k}$ indicated in Figure 5.9. This is similar to Fault G. However, the faults were injected at full adder cell FA_2 and a longer path sensitised so that the output was observed at primary output P4. Table 5.10 shows the details of the signal propagation path for each of the faults.

Fault	Fault Site	Signal Processing Path
F	R_a at FA_2	$Y_2X_0 \rightarrow FA_2 \rightarrow P_2$
G	R_b at FA_1	$Y_1X_0 \rightarrow FA_1 \rightarrow FA_4 \rightarrow P_2$
H	R_c at FA_1	$Y_1X_0 \rightarrow FA_1 \rightarrow FA_4 \rightarrow P_2$
K	R_b at FA_2	$Y_2X_0 \rightarrow FA_2 \rightarrow FA_5 \rightarrow FA_8 \rightarrow FA_10 \rightarrow P_4$

Table 5.10: Signal propagation path

Table 5.11: Path Delay and Path Delay Ratio for Fault G

RES Ω	**Voltage** V	**Path Delay: Rise (s)**	**Ratio Rise**	**Path Delay: Fall (s)**	**Ratio Fall**
2.00k	1.20	2.24E-10	0.95	3.11E-10	1.46
3.50k	1.20	2.29E-10	0.97	2.51E-10	1.18
2.00k	1.00	2.68E-10	0.94	4.39E-10	1.69
3.50k	1.00	2.76E-10	0.97	3.21E-10	1.24
2.00k	0.85	3.41E-10	0.89	1.03E-09	2.91
3.50k	0.85	3.69E-10	0.97	5.03E-10	1.42

Table 5.12: Path Delay and Path Delay Ratio for Fault K

RES Ω	**Voltage** V	**Path Delay: Rise (s)**	**Ratio Rise**	**Path Delay: Fall (s)**	**Ratio Fall**
2.00k	1.20	4.40E-10	0.97	5.67E-10	1.25
3.50k	1.20	4.46E-10	0.98	4.95E-10	1.09
2.00k	1.00	5.30E-10	0.97	7.54E-10	1.37
3.50k	1.00	5.39E-10	0.98	6.17E-10	1.12
2.00k	0.85	7.00E-10	0.94	1.53E-09	2.03
3.50k	0.85	7.28E-10	0.98	9.10E-10	1.21

Table 5.12 shows selected results from Fault K simulations. The results were directly compared to results from fault G which are shown in Table 5.11. By comparing these two tables, some conclusion can be drawn. Faults propagating in the longer path cause the absolute delay to be longer than the same fault propagating through a shorter path. However, the delay ratio for a shorter path is higher than the delay ratio of a longer path. This is true for all cases of supply

voltage. For example, a resistive bridging fault of 2000 Ω with a supply voltage of 1.2V has a falling path delay of 3.11E-10 s for Fault G and 5.67E-10 s for Fault K. There is an increased delay of 2.56E-10 s. This delay can be attributed to the extra transition and switching delay along the path. The delay ratios are 1.46 and 1.25 for Fault G and Fault K respectively.

5.6 Summary

This chapter presents an approach to a testing strategy for delay faults in Multi Voltage Design. Through extensive simulations, we have analysed defect behaviours of two main classes of fault: resistive open and resistive bridging faults. In both classes of defects, the absolute delay values increased when the supply voltage was reduced.

From resistive open fault simulations, results show that a higher supply voltage gives a better delay ratio. This directly shows that higher fault coverage could be achieved by testing for the fault at a higher voltage. Even though the lower supply voltage increases the nominal delay, it rarely increases faster than the transistor delay.

For the case of resistive bridging faults, the lower the power supply voltage, the higher the fault coverage will be. This is due to the fact that a lower power supply voltage will cause a larger fraction of resistive shorts to malfunction. For both classes of faults, studies show that faults propagated through longer paths have lower delay ratios which further emphasise the importance of using voltage specific delay fault tests.

We have observed similar observations for circuit simulation using ST 0.35μm technology. Transmission gate open and resistive short defects were used in the case study. The results and discussions can be found in Appendix A.

The overall conclusion is that in order to guarantee the test quality of Multi Voltage Designs it will be necessary to select a number of voltage-specific delay fault tests, in addition to voltage-independent stuck-fault tests. Initial testing can be done at the highest operating voltage and this will reduce the time and cost of the test. The escaped defects can be detected at lower mid-range voltages without the need to go to the lowest voltages.

Chapter 6

Testing of Level Shifter

6.1 Introduction

In the previous chapter, we have made suggestions for voltage settings to run delay fault testing for multi voltage design(MVD) circuits. The assumption made is that all parts of the circuit are digital. However, there are part of the MVD circuits which are not purely digital. An example of such a component is a level shifter. A methodology for testing these components is critical in order to achieve the required total fault coverage.

An interesting fact is that the inclusion of level shifters in multi voltage design is no longer optional. The total number of level shifters in a typical SoC chip can run to approximately 4700 of which 2500 are Up-Shifters [140]. Practically, that is a huge number and a proper testing method needs to be extensively studied.

A level shifter provides a clean interface between two voltage domains. This will reduce timing closure problems and excessive crowbar switching currents. The two problems, if not properly addressed, will jeopardise the overall power minimisation goal.

This work attempts to answer two key questions. Can we test a level shifter as a digital circuit using conventional Design-For-Test(DFT) techniques? Can we test the level shifters using one voltage or a set of voltages (at one time)?

In section 6.2 , it is shown that insensitive input vectors such as (1,1) for a bridging fault become sensitive with the inclusion of level shifters in the circuit. Section 6.3 studies the ability of the level shifters to propagate the fault effect. Section 6.4 investigates the feasibility of using a single supply voltage to detect faults in level shifters. Although, by definition, level shifters are designed to operate between two voltage domains, it would be easier if tests could be performed using a single supply voltage. This would reduce the complexity of the Design-For-Test(DFT) architecture.

6.2 Level Shifters in the Presence of Resistive bridging faults

In previous works [83],[86], [57],[1] it has been noted that bridging defects are dormant as long as both driving signals are the same. In these works, all driving signals have the same voltage level and the driven gates have the same threshold voltages.

However, in a multi voltage chip, the driving signals can originate from different cores that are operating at different supply voltages. The receiving end of the circuit might have a different threshold. A level shifter will have the role of interface between these two domains. If the level shifter is defective, there is the possibility that the output signal from the level shifter is misinterpreted by the receiving end. For an example, the driving signal (1,1) can be interpreted as (0,1) or (1,0).

To demonstrate this, we have conducted simulations in which two defects were injected: a malfunctioning level shifter and a bridging fault. In these simulations we have overruled the classic Single Fault Assumption (SFA). Hiroshi [141] has reported that multiple faults are not avoidable in a combinational circuit. Hiroshi has shown that tests generated under the single fault assumption maybe invalid for combinational circuits with multiple faults. Moreover, since dealing with multiple faults is necessary in fault diagnosis [142], we have considered two faults at the same time. The input voltage ranges used in these simulations are the voltage ranges of the Intel PXA270 processor which operates between 0.8075V and 1.705V[125]

Figure 6.1 shows the circuit model used in our study. The level shifters' function is to bring up the voltage signal from VDDL to VDDH. VDDL and VVDH are set to 0.85V and 1.7V respectively for these simulations. R_bridge is the bridging fault resistance between nodes Vn0 and Vn1. The driving strength from inverters connected to the resistive bridge (Inv_a and Inv_d) together with the value of the resistance will determine the intermediate voltage values at nodes Vn0 and Vn1. These intermediate voltages will then decide the output at Inv_b and Inv_e. As we have seen from the previous chapter, the injected resistive bridge can cause timing failure as well as logical error.

Two scenarios which represent two different cases were simulated. In the first case, Level Shifter A and Level Shifter B are functional thus the outputs of the level shifters are VDDH (1.7V). We ran simulations using different values of bridging fault resistance, from 100Ω to 5MΩ. Significant results are presented in Table 6.1 as the Single Fault Case. The 'Actual Path Delay' is measured from the time when the input signals reach 50% of VDDL to the time the output signals reach 50% of VDDH. The Path Delay ratio is calculated by taking the ratio between the delay without the bridging fault to the delay when the R_bridge is present.

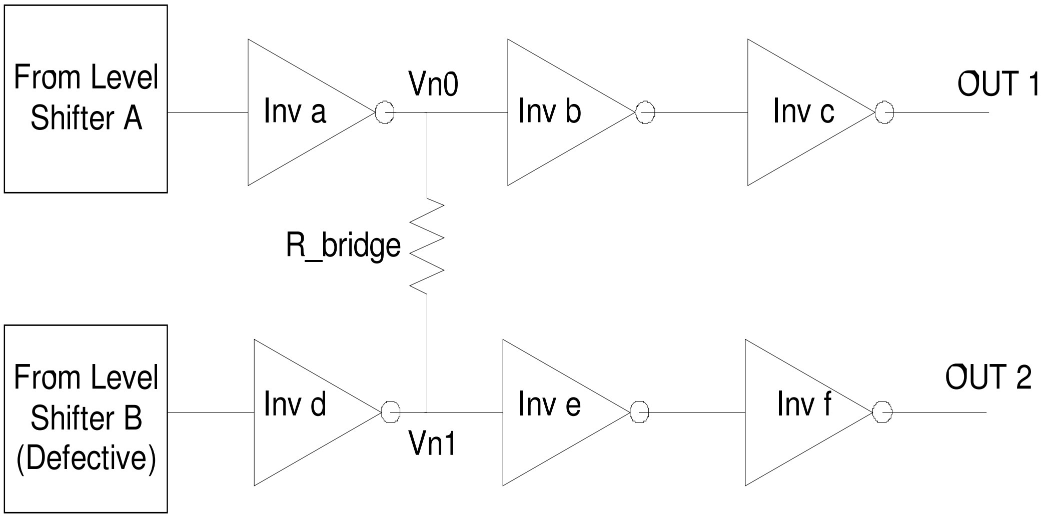


Figure 6.1: Circuit model for defective level shifter in presence of bridging fault

In the second scenario which is noted as the Double Fault Case, Level Shifter B is defective. The defective level shifter cannot bring the signal up to VDDH and will give a degraded output (VDDL). The delay ratio in this case is the ratio between these two delays i.e. the 'Actual Path Delay' and delay measured for the defective level shifters but without the bridging fault.The results are shown as the 'Double Fault Case' in Table 6.1.

Table 6.1: Path delay and path delay ratio for single and double faults

R_bridge	Single Fault Case		Double Fault Case	
	Actual Path Delay (s)	Path Delay Ratio	Actual Path Delay (s)	Path Delay Ratio
1M	1.88E-11	1.00	3.37E-11	1.00
125k	1.88E-11	1.00	3.40E-11	1.00
25k	1.89E-11	1.01	3.54E-11	1.05
2.5k	2.09E-11	1.11	9.39E-11	2.79
1.5k	2.26E-11	1.20	51.80E-11	15.41
250	3.88E-11	2.07	52.00E-11	15.46
100	4.59E-11	2.44	52.10E-11	15.50

From Table 6.1 we observe that for the single fault case, the circuit delay increases with the reduction of resistance. The increase in delay is more significant with smaller values of fault resistance. Similar observations apply to the path delay ratio.

In the Double Fault Case, the actual delay increases when the defective input from the level shifter is used as an input. The value of the delay as well as the delay ratio is much higher than for the single fault case. The extra delay is due to the lower driving strength from the defective level shifter.

From the results, we can infer that a defective level shifter will exaggerate the impact of a bridging fault when both faults are present at the same time. These results are useful for two reasons: pattern generation and fault diagnosis. Even though inputs of (1,0) and (0,1) will have higher fault coverage [83], it is important to note that the defective (1,1) input is important for diagnosis reasons. We have demonstrated that this input may detect some of the bridging faults with different voltage domains.

6.3 Propagation of fault effect in level shifters

In order to include the level shifters in the path of a digital scan chain, it is important to know if the level shifters can propagate the fault effects. To demonstrate this, two types of faults, resistive opens and bridging faults, were used in simulations. The Contention Mitigated Level Shifters (CMLS) [119], Figure 6.2, has been used in all our fault simulations.

6.3.1 Simulation Setup for Resistive Open Defects

Two blocks of 8 bit ripple carry adders linked together were used in the simulations. Resistive open defects were injected at carry in (CIN) and the longest possible path in the circuit was sensitized.

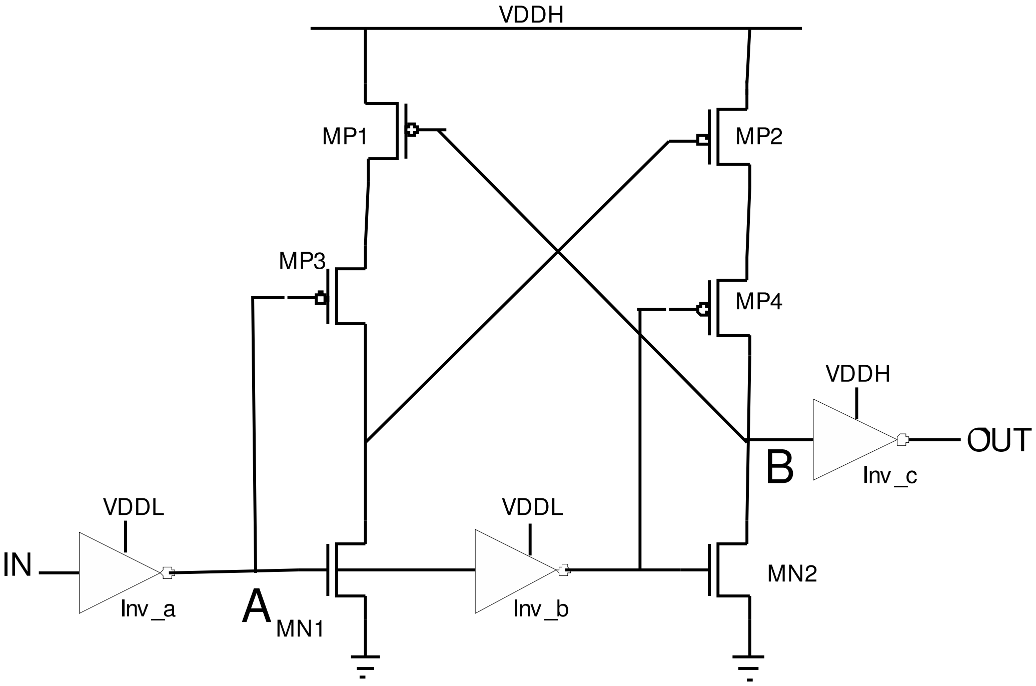


Figure 6.2: Contention mitigated level shifter

Figure 6.3(a) and (b) show the simulation setup for resistive open defects. In Figure 6.3(a), V1 and V2 are set to the same level. We have simulated the circuit over a large range of resistances and 5 sets of voltage settings. Due to the similarity of the results, only selective results are shown in this thesis. Table 6.2 shows the results of simulations for three different values of supply voltage: 0.85, 1.2 and 1.7V.

The actual path delay was measured between the carry in (CIN_0) for ADDER block 1 and carry out (COUT_1) of ADDER block 2. It was observed that as the value of the resistor increases, the path delay and path delay ratio increased for all values of voltage. It can also be observed that the delay ratio for higher values of supply voltage is higher. This is an expected pattern of result in line with our findings from Chapter 2.

The simulation was repeated by adding a level shifter between the two adders. The simulation setup is shown in Figure 6.3(b). We have set the level shifter to shift a low voltage of 0.85V to 1.7V. Table 6.3 shows the simulation results.

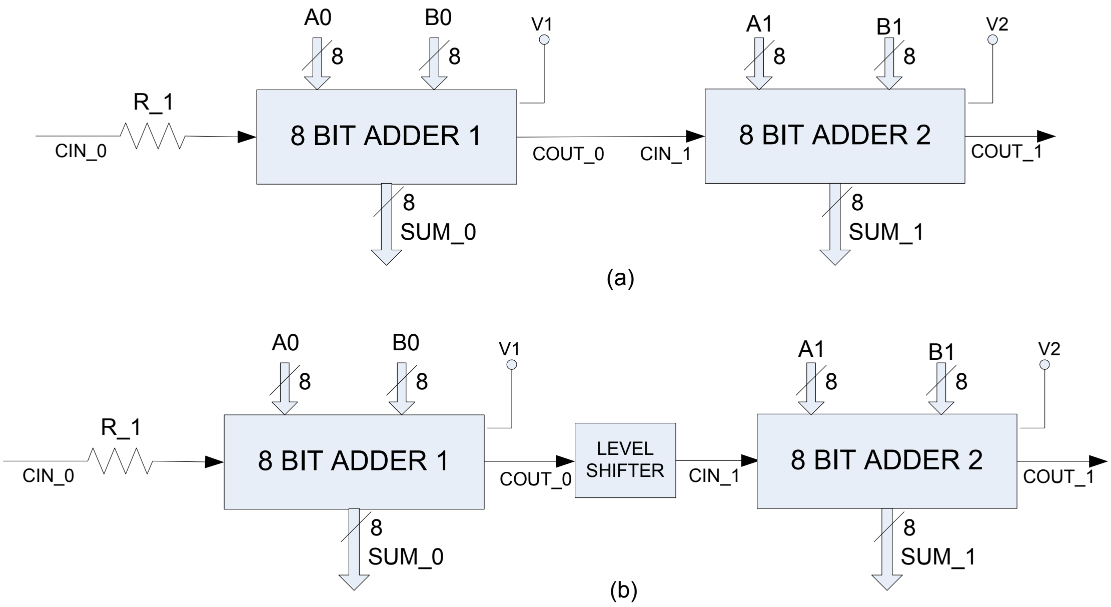


Figure 6.3: Simulation setup for resistive open fault

Table 6.2: Delay and Delay Ratio for circuit without level shifters

Resistor Value Ω	Supply Voltage (V)	Actual Path Delay(s)	Path Delay ratio
0	0.85	3.04E-09	1.00
1k	0.85	3.06E-09	1.01
10k	0.85	3.20E-09	1.06
50k	0.85	3.77e-09	1.24
250k	0.85	1.26E-08	4.14
0	1.20	2.00E-09	1.00
1k	1.20	2.01E-09	1.01
10k	1.20	2.14E-09	1.07
50k	1.20	2.66e-09	1.33
250k	1.20	7.92E-09	3.97
0	1.70	1.52E-09	1.00
1k	1.70	1.54E-09	1.01
10k	1.70	1.66E-09	1.09
50k	1.70	2.14e-09	1.40
250k	1.70	7.19E-09	4.72

It is evident from Table 6.2 and Table 6.3 that the level shifter has propagated the fault effect. The path delay ratio for a circuit with a level shifter resembles the ratio for the circuit without the level shifter.

Table 6.3: Delay and Delay Ratio for circuit with level shifters

Resistor Value Ω	Supply Voltage, V1(V)	Supply Voltage, V2(V)	Actual Delay(s)	Path Delay ratio
0	0.85	1.70	2.36E-09	1.00
1k	0.85	1.70	2.38E-09	1.01
10k	0.85	1.70	2.53E-09	1.07
50k	0.85	1.70	3.10E-09	1.31
250k	0.85	1.70	1.19E-08	5.04

6.3.2 Simulation Setup for Bridging Fault Defects

First, simulations were conducted using the setup without the level shifters as in Figure 6.4 (a) and later with the level shifters as in Figure 6.4 (b) to observe if the level shifters can propagate fault effects. The bridging faults were injected at the second bit of ADDER1. Four fault locations were chosen from all possible fault locations. These fault locations are as shown in Figure 6.5.

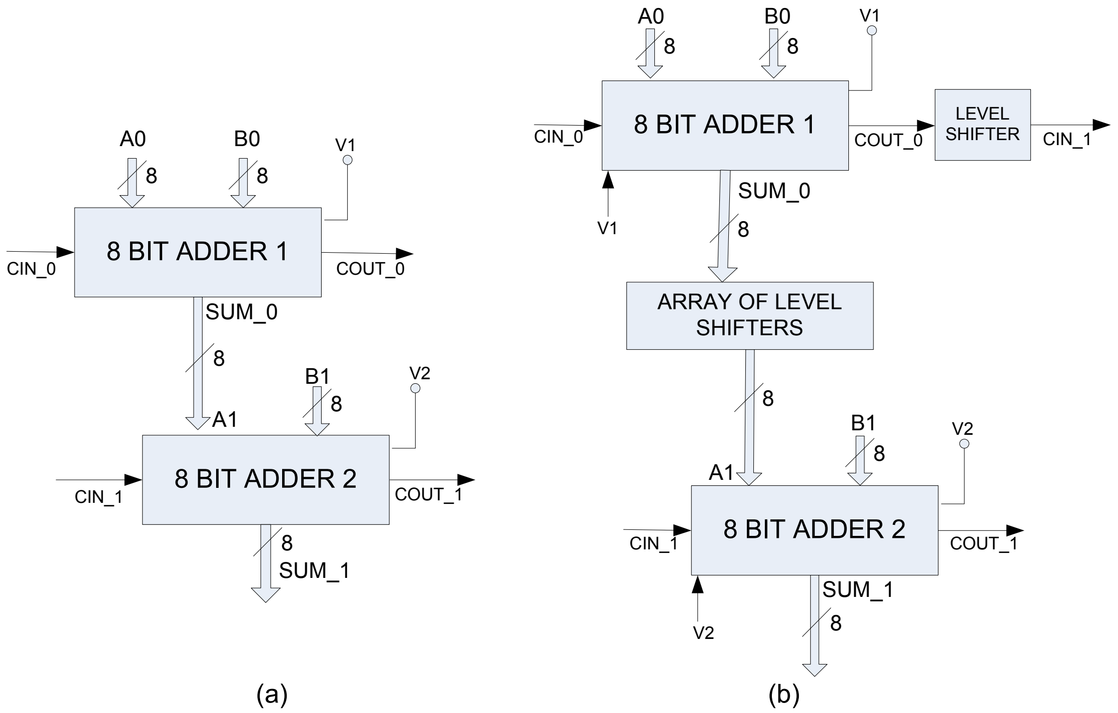


Figure 6.4: Simulation setup for bridging fault

The outputs are observed at two locations, both at ADDER2. These locations are: SUM of the second bit and SUM of the third bit. Due to the similarity of

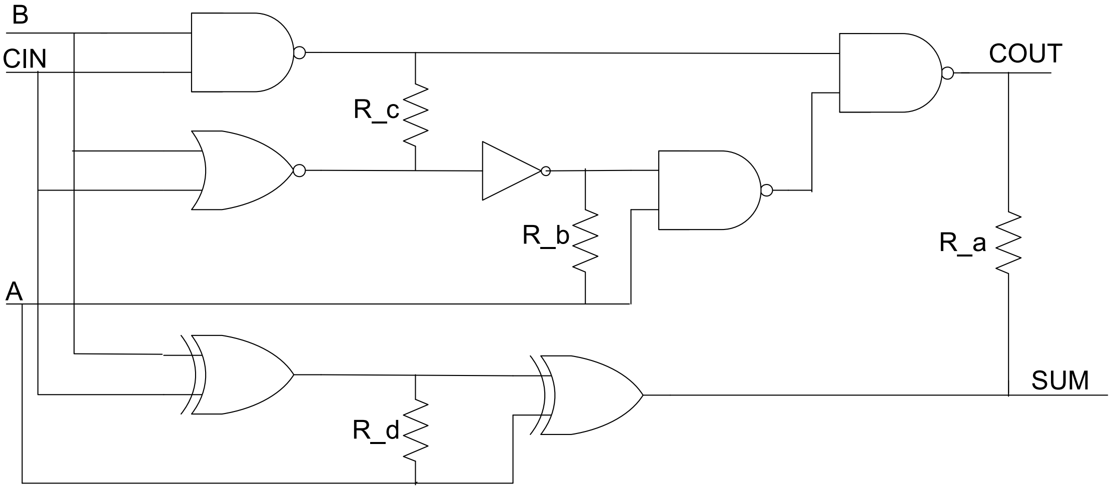


FIGURE 6.5: Location of Bridging Fault in Adder Slice Circuit

the results, only one of the results is presented and discussed in this thesis. Table 6.4 and Table 6.5 shows the path delay and path delay ratio for both cases: with and without level shifters. Similar observations were made for three other fault locations.

TABLE 6.4: Path Delay and Path Delay Ratio for circuit without level shifters

Resistor Value Ω	Supply Voltage (V)	Actual Delay(s)	Path Delay ratio
500k	0.85	2.32E-10	1.00
5k	0.85	2.46E-10	1.06
500	0.85	4.33E-10	1.87
500k	1.70	1.18E-10	1.00
5k	1.70	1.18E-10	1.00
500	1.70	4.33E-10	3.67

TABLE 6.5: Delay and Delay Ratio for circuit with level shifters

Resistor Value Ω	Supply Voltage, V1(V)	Supply Voltage, V2(V)	Actual Delay(s)	Path Delay ratio
500k	0.85	1.70	3.20E-10	1.00
5k	0.85	1.70	3.26E-10	1.05
500	0.85	1.70	5.16E-10	1.66

These results show that the level shifter can propagate the fault effect. From these observations both from resistive open and resistive short defects, we can conclude that the level shifters can be included in the digital DfT architecture. This conclusion is drawn since digital defects can be propagated without any loss of information through a level shifter.

6.4 Defects in Level Shifters

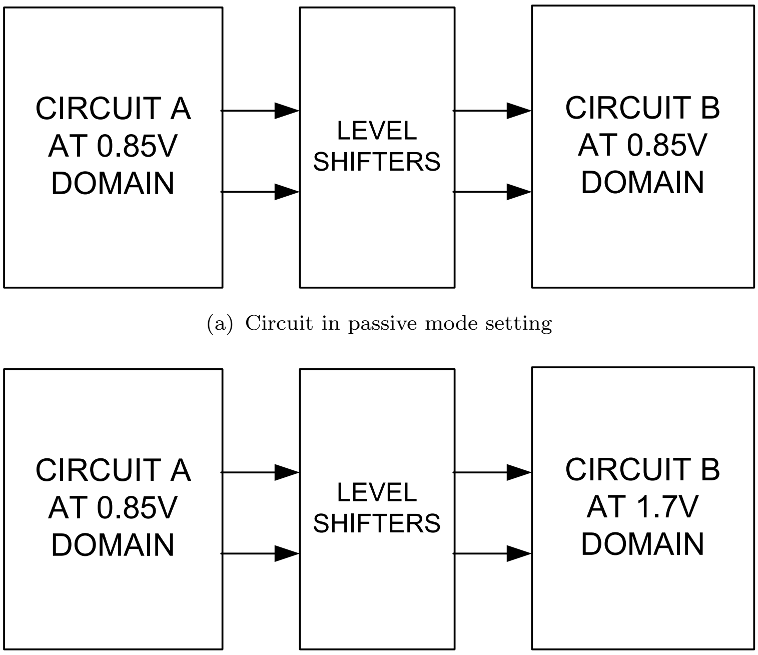
From the previous section, it is evident that a level shifter can propagate a digital fault effect. This fault originates from another part of a circuit and the level shifters are responsible to shift the voltage level. In this section, faults within a level shifter and their impact on the overall circuit are studied.

We have simulated a number of possible defects in the level shifter circuit and examined their responses at different supply voltages. The fault simulations are conducted under two conditions. In the first condition, two voltage domains are used.

We introduce the term ACTIVE mode to mean that the level shifter is actively shifting from a low voltage domain to a high voltage domain. In the second condition, which we name the PASSIVE mode (no active shifting), all parts of the circuit are operating with the same supply voltage. Figure 6.6 shows an example of both ACTIVE and PASSIVE mode setting.

In Figure 6.6(b), the level shifters are responsible for shifting the input voltage of 0.85V from circuit A to 1.7V which is the operational voltage of circuit B. In Figure 6.6(a) both circuit A and circuit B are set to operate 0.85V. In this case, the level shifter will be passive.

In our simulations, circuit A is represented by ADDER1 and circuit B is represented by ADDER2. The two main classes of defects studied in this thesis are resistive shorts and resistive opens.



(a) Circuit in passive mode setting

(b) Circuit in active mode setting

Figure 6.6: Example of PASSIVE and ACTIVE mode setting for circuits with integrated level shifters

We have used 5 steps of input voltage: 0.85V, 1.00V, 1.20V, 1.40V and 1.70V. For each of fault, there is a total of ten combinations of voltages at circuit A and circuit B. It is important to note that we are only considering cases of up-shift i.e shifting from low voltage to higher voltage.

6.4.1 Resistive Open

Four locations were selected and the resistive open faults were injected in the Contention Mitigated Level Shifters (CMLS). These four locations were selected by considering the probability of the fault occurrence as well as the impact factor of the faults on the circuit. If the possible fault location has a higher probability of fault occurrence due to the layout structure and at the same time has a higher impact on the circuit, it will be selected. The locations of the faults are shown in Figure 6.7. Each fault is injected singly. The setup in Figure 6.4 (b) was used for all the simulations for resistive opens in both ACTIVE and PASSIVE modes. The faulty level shifter was placed at the first bit of the array of level shifters.

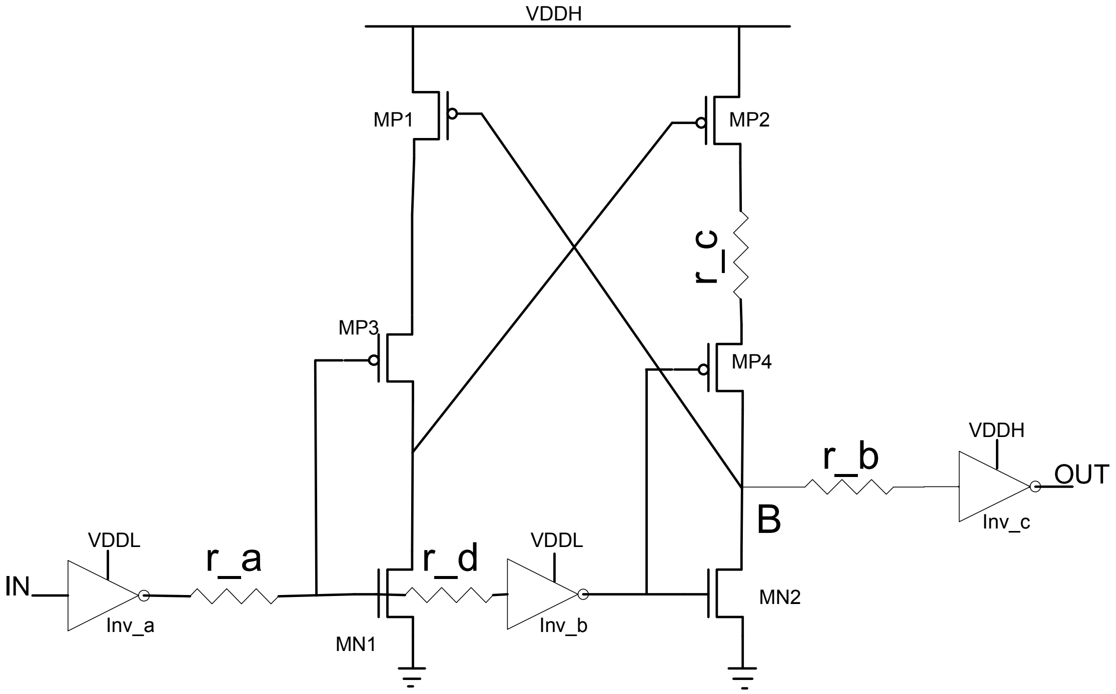


Figure 6.7: Defect location for resistive opens

Each defect was simulated at a total of five PASSIVE mode settings and ten ACTIVE mode settings. We have looked in detail at results from fault location r_a which is labeled as Fault A.

6.4.1.1 Simulation Results

Table 6.6 shows the result of fault simulations in PASSIVE mode settings for Fault A. The resistance values were increased from the non-faulty value to the almost open value. For readability purposes, we are only showing results from four different values of resistor. From Table 6.6 the absolute delay which is labelled as Path Delay Rise and Path Delay Fall, decreases with the increase of the supply voltage. However, on the contrary, the delay ratio labelled as Ratio Rise and Ratio Fall increases at higher voltages. The highest voltage setting at 1.7V has almost twice the value in delay ratio compared with the lowest voltage setting of 0.85V. This is in line with our findings from chapter 5.

Table 6.6: Path Delay and Path Delay Ratio for Fault A in PASSIVE Mode

RES	V1 (V)	V2 (V)	Path Delay Rise (s)	Ratio Rise	Path Delay Fall (s)	Ratio Fall
0.00	0.85	0.85	2.69E-10	1.00	2.81E-10	1.00
25k	0.85	0.85	4.00E-10	1.49	4.07E-10	1.45
250k	0.85	0.85	1.42E-09	5.29	1.40E-09	4.96
1M	0.85	0.85	1.04E-08	38.72	1.03E-08	36.68
0.00	1.00	1.00	2.17E-10	1.00	2.26E-10	1.00
25k	1.00	1.00	3.45E-10	1.59	3.49E-10	1.55
250k	1.00	1.00	1.35E-09	6.23	1.34E-09	5.95
1M	1.00	1.00	1.02E-08	47.02	1.03E-08	45.83
0.00	1.20	1.20	1.78E-10	1.00	1.85E-10	1.00
25k	1.20	1.20	3.05E-10	1.71	3.09E-10	1.67
250k	1.20	1.20	1.29E-09	7.27	1.31E-09	7.10
1M	1.20	1.20	1.00E-08	56.14	1.04E-08	56.33
0.00	1.40	1.40	1.56E-10	1.00	1.62E-10	1.00
25k	1.40	1.40	2.81E-10	1.81	2.88E-10	1.78
250k	1.40	1.40	1.26E-09	8.09	1.30E-09	8.03
1M	1.40	1.40	9.87E-09	63.37	1.05E-08	64.75
0.00	1.70	1.70	1.36E-10	1.00	1.42E-10	1.00
25k	1.70	1.70	2.62E-10	1.92	2.70E-10	1.90
250k	1.70	1.70	1.23E-09	9.00	1.29E-09	9.06
1M	1.70	1.70	9.75E-09	71.52	1.05E-08	73.98

Table 6.7 shows the results of simulation for ACTIVE mode setting for resistive open r_a. There are a number of observation from this table.

For a given V1, the delay ratio increases as V2 increases. For example, for V1 of 0.85V, the rise delay ratios for resistance value of 2.50E+05 Ω are 5.45 for V2 of 1.00V. When V2 is increased to 1.70V, the rise delay ratio increases to 6.43.

The next observation is that the delay ratio increases as the V1 is increased to higher values. For example, the delay ratio for V1=0.85, V2=1.70 and R=2.5E+05 Ω is 6.43. For the similar setting with V1=1.40V, V2=1.70 and R=2.5E+05 Ω the delay ratio is 8.36.

The two observations above suggest that running the test at a higher voltage will result in higher fault coverage.

Direct comparison between the PASSIVE mode and ACTIVE mode were conducted to find the more preferable method to test the level shifters. It is observed that the PASSIVE mode setting with highest voltage setting at 1.70V has the highest delay ratio. For the lower voltage of 1.40V, the comparison is made between V1=1.20V and V2=1.40V in ACTIVE mode and V1=V2=1.40V in PASSIVE mode. Clearly, the PASSIVE mode settings has a better delay fault ratio.

It is an added advantage to observe that the PASSIVE mode testing has a better delay ratio. This is due to the fact that setting up the test circuit in ACTIVE mode requires a more complicated setup. On the other hand, a circuit in default always has one level of voltage. As such, the PASSIVE mode will not cost any extra effort in term of time or resources.

Resistive open was injected singly at other fault locations r_b, r_c and r_d which are labelled as Fault B, Fault C and Fault D respectively. As explained in section 6.4, for each fault location, five PASSIVE mode settings and ten ACTIVE mode settings were conducted and results were obtained. Due to similarity of results, selected results are presented here.

Table 6.8 shows results from Fault B, C and D for ACTIVE mode settings. Rather than presenting the full comprehensive results, we have chosen two values of resistance for each fault case. The open resistances of 25k Ω and 500K Ω are sufficient to demonstrate the results pattern. Also, we have chosen four different active voltage setting combinations for illustration. Very similar patterns as for Fault A were observed in all fault cases.

Table 6.9 shows results from PASSIVE mode settings. Results for two values of resistance are presented. As for the case of Fault A, it is evident that the PASSIVE mode setting at the highest voltage results in a better delay fault ratio. This

Table 6.7: Path Delay and Path Delay Ratio for Fault A in ACTIVE Mode

RES	**V1 (V)**	**V2 (V)**	**Path Delay Rise (s)**	**Ratio Rise**	**Path Delay Fall (s)**	**Ratio Fall**
0.00	0.85	1.00	2.53E-10	1.00	2.54E-10	1.00
25k	0.85	1.00	3.85E-10	1.52	3.82E-10	1.50
250k	0.85	1.00	1.41E-09	5.59	1.39E-09	5.45
1M	0.85	1.00	1.04E-08	41.25	1.04E-08	40.73
0.00	0.85	1.20	2.44E-10	1.00	2.38E-10	1.00
25k	0.85	1.20	3.77E-10	1.54	3.71E-10	1.56
250k	0.85	1.20	1.42E-09	5.79	1.43E-09	6.00
1M	0.85	1.20	1.05E-08	42.88	1.05E-08	44.28
0.00	0.85	1.40	2.46E-10	1.00	2.33E-10	1.00
25k	0.85	1.40	3.80E-10	1.54	3.72E-10	1.60
250k	0.85	1.40	1.43E-09	5.82	1.47E-09	6.29
1M	0.85	1.40	1.05E-08	42.79	1.07E-08	45.84
0.00	0.85	1.70	2.63E-10	1.00	2.36E-10	1.00
25k	0.85	1.70	3.97E-10	1.51	3.81E-10	1.61
250k	0.85	1.70	1.47E-09	5.59	1.52E-09	6.43
1M	0.85	1.70	1.06E-08	40.35	1.08E-08	45.74
0.00	1.00	1.20	2.06E-10	1.00	2.07E-10	1.00
25k	1.00	1.20	3.35E-10	1.63	3.32E-10	1.61
250k	1.00	1.20	1.35E-09	6.56	1.34E-09	6.47
1M	1.00	1.20	1.02E-08	49.73	1.04E-08	50.17
0.00	1.00	1.40	2.03E-10	1.00	1.98E-10	1.00
25k	1.00	1.40	3.32E-10	1.64	3.28E-10	1.65
250k	1.00	1.40	1.36E-09	6.71	1.37E-09	6.91
1M	1.00	1.40	1.03E-08	50.77	1.05E-08	52.85
0.00	1.00	1.70	2.07E-10	1.00	1.95E-10	1.00
25k	1.00	1.70	3.37E-10	1.63	3.31E-10	1.70
250k	1.00	1.70	1.38E-09	6.69	1.41E-09	7.23
1M	1.00	1.70	1.03E-08	50.00	1.06E-08	54.28
0.00	1.20	1.40	1.72E-10	1.00	1.74E-10	1.00
25k	1.20	1.40	3.00E-10	1.74	3.00E-10	1.72
250k	1.20	1.40	1.30E-09	7.54	1.31E-09	7.50
1M	1.20	1.40	1.00E-08	58.21	1.04E-08	59.79
0.00	1.20	1.70	1.71E-10	1.00	1.68E-10	1.00
25k	1.20	1.70	2.99E-10	1.75	2.97E-10	1.77
250k	1.20	1.70	1.32E-09	7.69	1.33E-09	7.94
1M	1.20	1.70	1.01E-08	59.04	1.05E-08	62.53
0.00	1.40	1.70	1.52E-10	1.00	1.53E-10	1.00
25k	1.40	1.70	2.79E-10	1.83	2.80E-10	1.83
250k	1.40	1.70	1.27E-09	8.36	1.29E-09	8.45
1M	1.40	1.70	9.91E-09	65.25	1.05E-08	68.39

implies that the impact of the defect becomes more visible when the level shifter is in PASSIVE mode. In addition, as we have observed in the previous chapter, the highest operating voltage is preferred to achieve better fault coverage.

Table 6.8: Falling Path Delay Ratio for Fault B, Fault C and Fault D in ACTIVE mode

		Fault B		Fault C		Fault D	
V1	V2	25k	500K	25k	500K	25k	500K
0.85	1.70	1.52	7.26	2.14	SF	1.61	9.70
1.20	1.40	1.73	9.82	2.70	SF	1.71	11.28
1.20	1.70	1.75	9.57	2.68	SF	1.77	11.97
1.40	1.70	1.82	10.73	2.85	SF	1.83	12.82

Table 6.9: Falling Path Delay Ratio for Fault B, Fault C and Fault D in PASSIVE mode

	Fault B		Fault C		Fault D	
V_{dd}	25k	500K	25k	500K	25k	500K
0.85	1.47	7.11	2.29	SF	1.45	7.23
1.00	1.57	8.33	2.48	SF	1.55	8.81
1.20	1.69	9.58	2.69	SF	1.67	10.65
1.40	1.78	10.51	2.85	SF	1.78	12.15
1.70	1.89	11.48	3.02	SF	1.90	13.79

6.4.2 Resistive Short

Resistive short faults were injected at different points in the level shifters. Five different locations of the defects are shown as R_e, R_f, R_g, R_h and R_j in Figure 6.8. Rather than reporting the full detailed results of the simulation, we have selected two faults and presented their results. These are at locations R_e and R_h.

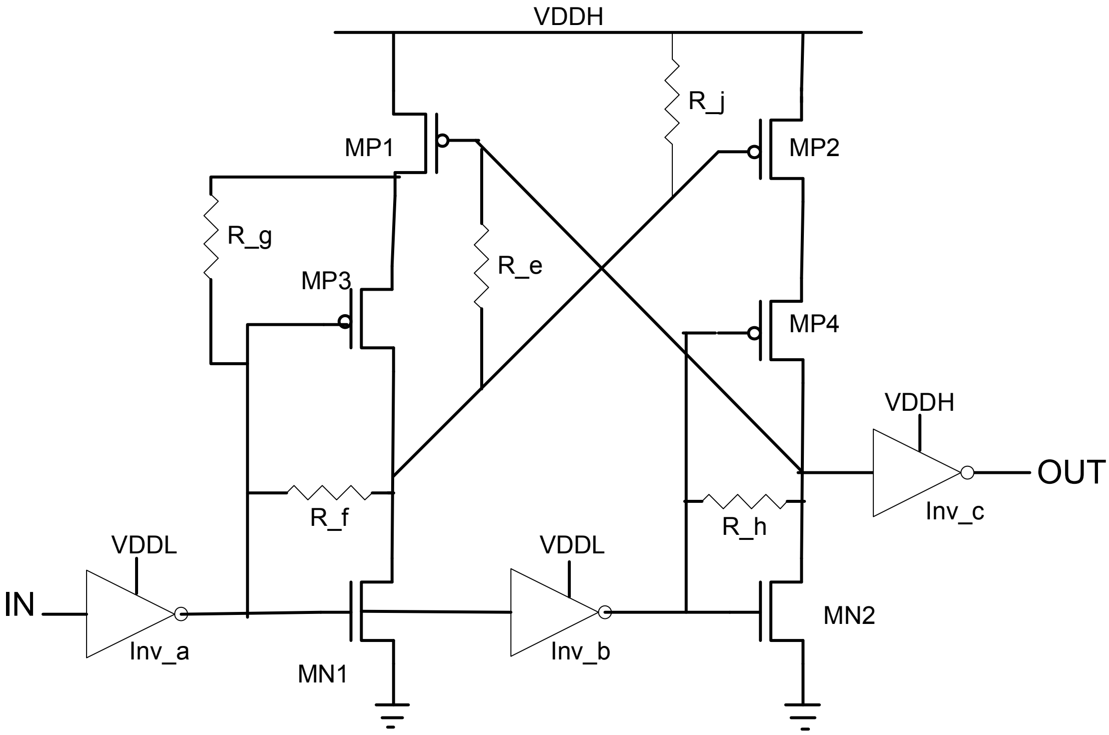


Figure 6.8: Defect location for resistive shorts

6.4.2.1 Simulation Results

The simulation results for resistive shorts for both the ACTIVE and PASSIVE modes are tabulated in Table 6.10 and Table 6.11, respectively. It can be noted that some of the faults cause speed up. In some cases the speed up is up to 26% as in the case of R= 1.5kΩ, fault R_d, ACTIVE mode. Here, the resistive short causes imbalance in the symmetry of the level shifter circuit and causes the speed up. Even though the speed up will not cause any performance degradation or logical error, it is important to detect these faults. An undetected fault might cause other performance degradation in a different circuit environment. SF in the table means the result is a logical error and is categorized as a Stuck at Fault. These faults can be detected both by means of delay fault testing as well as stuck-at-fault testing.

These results suggest that a PASSIVE mode test will result in better fault detection. It is also preferable to conduct the test at the lowest possible voltage in order to have the highest fault coverage for the case of resistive short faults. As

Table 6.10: Path Delay and Path Delay Ratio for Fault R_e and Fault R_h in ACTIVE mode, V1=0.85V and V2=1.7V

Resistor Value Ω	Fault R_e Actual Path Delay (s)	 Path Delay Ratio	Fault R_h Actual Path Delay (s)	 Path Delay Ratio
10M	2.63E-10	1.00	2.63E-10	1.00
3k	2.56E-10	0.97	2.10E-10	0.80
1.5k	2.90E-10	1.11	1.94E-10	0.74
1k	3.94E-10	1.50	-	SF
750	-	SF	-	SF

Table 6.11: Path Delay Ratio for Fault R_e and Fault R_h in PASSIVE mode

	Fault R_e			Fault R_h		
V_{dd}	3K	2k	1500	3K	2k	1500
1.70	0.97	0.96	0.94	0.97	0.96	0.94
1.40	0.97	0.96	0.84	0.97	0.96	0.84
1.20	0.97	0.95	SF	0.97	0.95	SF
1.00	0.97	SF	SF	0.97	SF	SF
0.85	0.97	SF	SF	0.97	SF	SF

can be observed, the lowest supply voltage i.e 0.85V in PASSIVE mode will detect resistive shorts up to 2.5kΩ. On the other hand, the highest voltage, 1.7V, in ACTIVE mode can only detect resistive shorts smaller than 1kΩ for Fault H.

6.5 Analysis

From the simulation results for the two different classes of faults, it can be observed that a fault in the level shifters can be detected using a digital fault model, either the stuck-at or delay fault model. Another important observation is that the fault can be detected at a single supply voltage i.e in the PASSIVE mode. The question then arises as to which voltage gives better fault coverage. Results from resistive open faults suggest that a higher voltage gives a better path delay ratio for resistive open defects. This signifies that there will be a larger variation in the path delay between the fault-free and faulty circuits at higher voltages. For the case of resistive shorts, results show that the path delay ratio is more pronounced at a lower voltage. However, testing at a lower voltage will increase the test application time [4], [2].

6.6 Summary

Testing of level shifters in a multi-voltage design has been studied. Experiments show how a defect in a level shifter can cause performance degradation in terms of timing as well as functional failure. A bridging fault effect will be amplified in the presence of a defective level shifter.

The key findings of this work which relate to the two initial questions are:

- The level shifter can be tested as a digital circuit using conventional Design-For-Test(DFT). This is justified since the defects in level shifters cause digital effects. In addition, level shifters can also propagate digital fault effects such as resistive opens and shorts.
- The level shifters can be tested using a single supply voltage, in the PASSIVE mode.
- To achieve maximum fault detection, the level shifters have to be tested in the PASSIVE mode at two voltage settings: the lowest and highest voltages. Resistive open faults have a better fault ratio at the highest voltage but resistive shorts have a better fault ratio at the lower voltage.

Chapter 7

Delay Fault Modelling/Simulation using VHDL-AMS in Multi-Vdd Systems

As design complexity increases, the time spent on each stage of design and test will also increase significantly. Testing phase will also be affected by these increases which will eventually increase the cost per item. In the previous two chapters, we have studied and recommended suitable supply voltage settings to achieve desirable test quality for different classes of defects.

In this chapter, we explore the idea of using mixed-mode languages such as VHDL-AMS for fault simulations. Rather than running the whole simulation at circuit level, part of circuits were modelled behaviorally. This model was then incorporated into mixed-mode simulations. The final aim is to reduce the total time involved which eventually reduces the cost per item.

The saving in processing time is presented in Section 7.1. The concept of circuit-level fault modelling and simulation is explained in Section 7.2. Model derivation for two classes of defects is given in Sections 7.3 and 7.4. Section 7.5 gives the simulation results used to verify the derived model. Analysis of the results is given in Section 7.6 and finally Section 7.7 summarises the work.

Different levels of fault simulation are presented and advantages are discussed. Section 7.2 is the main part of this chapter where the method of deriving the delay models are presented.

7.1 Quantifying the processing time

It is obvious that a design at SPICE level will take a longer time to be simulated in comparison with a design simulated at VHDL level. However VHDL level simulation will not give detailed delay simulations when faults are injected. Ideally, we need a combination of these levels to achieve time efficiency as well as accuracy in results.

For the fault simulation purpose there can be two options. In the first option, the general circuit description is given in VHDL. However, the faulty part is represented at the SPICE level. By doing this, the simulation time taken will be much less than having the whole design simulated in SPICE. Let us call the time taken to simulate the design in combination of VHDL and SPICE $\delta 1$. In the second option, the general description of the circuit is given in VHDL as in option 1. However, the faulty parts are written in VHDL-AMS in its behavioural level. Let us say the time taken to simulate the design in combination of VHDL and VHDL-AMS is $\delta 2$.

It is expected that $\delta 1$ will be higher than $\delta 2$. We have run repeated simulations on circuit designs in both combinations. We have observed that the VHDL and SPICE combination takes around 13% more time than combinations of VHDL and VHDL-AMS These simulations were conducted with the Single Fault Assumption made in all cases.

It is important to point out that the run-time measurement was not taken directly from the actual simulation tools used in this work. This is due to the fact that different computing platforms were used for both simulations. Mentor Graphics's System Vision VHDL-AMS were used as VHDL-AMS simulator and it can only be operated on Windows Platforms [143]. On the other hand, SPICE level simulation was conducted using Cadence Virtuoso Spectre Circuit Simulator which runs on Unix systems. In order to have direct comparison, we have run the SPICE simulation on Windows platform using HSPICE [144]. Even though the measurement was not directly obtained, the measurement from the HSPICE can be directly used for comparison purposes as our aim is to demonstrate the difference between computation time of two different simulation environment.

Even though the 13% might look as a small improvement, it is important to note that each fault simulation of the VHDL and SPICE combination will have a 13% time increase. For complex circuits, with a large number of fault simulations, the total saving of 13% will be very significant.

In the next section, the modelling and simulation of defects has been studied in detail. Firstly, a defect is simulated at SPICE level. This model is then implemented at the VHDL-AMS level. Finally, the model is verified at the VHDL and VHDL-AMS combination level.

7.2 Circuit Level Fault Modelling and Simulation

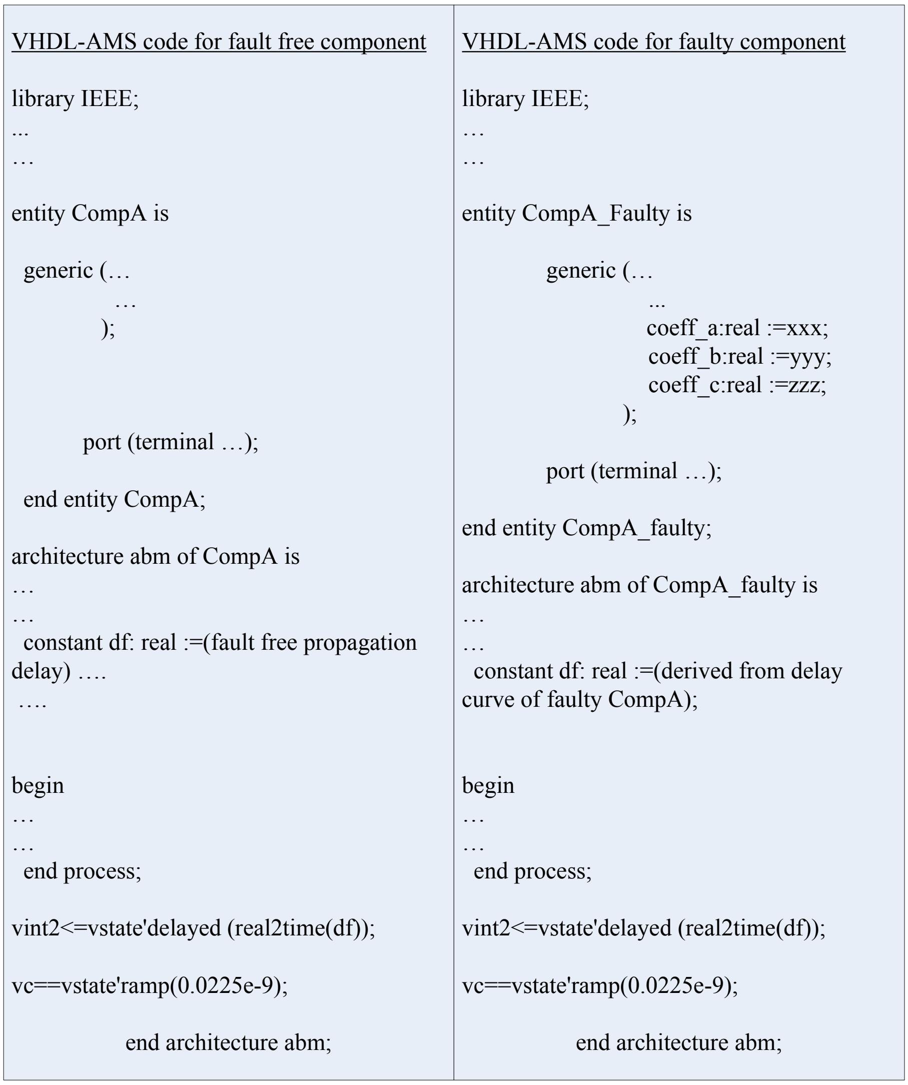
When a fault exists on a non feedback circuit, it will have an impact from the fault origin until the end of the circuit path. It can also have an impact on more than one path. However, there will not be any impact on any location before the fault location [8]. If the gate with the fault that will have the initial impact can be modelled in VHDL-AMS, the rest of the circuit elements can be modelled in simple VHDL(or VHDL-AMS). In Figure 7.2(a), the resistive open fault which is labelled as R will have an impact on gates G4 and G5 with gate G4 as the gate with initial impact. On the other hand, gates G1, G2 and G3 will not have any impact from the resistive fault. This principle is true for any non-feedback circuits.

The fault and the gate with initial impact can be modelled as shown in Figure 7.2(b). The shaded gate G4' represents the dotted area in Figure 7.2. The dotted area is simulated externally at the SPICE level to analyse its behaviour. Once the behaviour is quantified, it is then modelled at the VHDL-AMS level. An example of VHDL-AMS code for fault free component and faulty component is shown in Figure 7.1. The original code for the fault free component, *CompA*, which is shown on the left column has nominal propagation delay. When a defect is introduced the propagation delay is replaced by the delay model derived from the SPICE level measurements as depicted in the right column.

We have analysed two main classes of fault studied in detail in previous chapters. These are resistive open and resistive short defects. As we have observed in earlier chapters, these defects can cause performance degradation as well as logical error.

For resistive open defects, we have looked at 2-input-NAND and 2-input-NOR gates. These gates are the most commonly used gates [126]. In terms of transistor level circuits, a NAND gate has stacking pull-up architecture and a NOR gate has stacking pull-down architecture. It is expected that by deriving the model for these two generic gates, it will help to derive models for other gates using similar methods. For these two gates, we have derived models for one, two and four fan-outs. In total, there will be 6 cases. Model for each of the cases was derived and compared with the actual delay values.

For resistive shorts, we have used simplified models of resistive shorts. Inverter



VHDL-AMS code for fault free component

```
library IEEE;
...
…

entity CompA is

  generic (…
           …
         );



      port (terminal …);

  end entity CompA;

architecture abm of CompA is
…
…
  constant df: real :=(fault free propagation
delay) ….
 ….


begin
…
…
  end process;

vint2<=vstate'delayed (real2time(df));

vc==vstate'ramp(0.0225e-9);

          end architecture abm;
```

VHDL-AMS code for faulty component

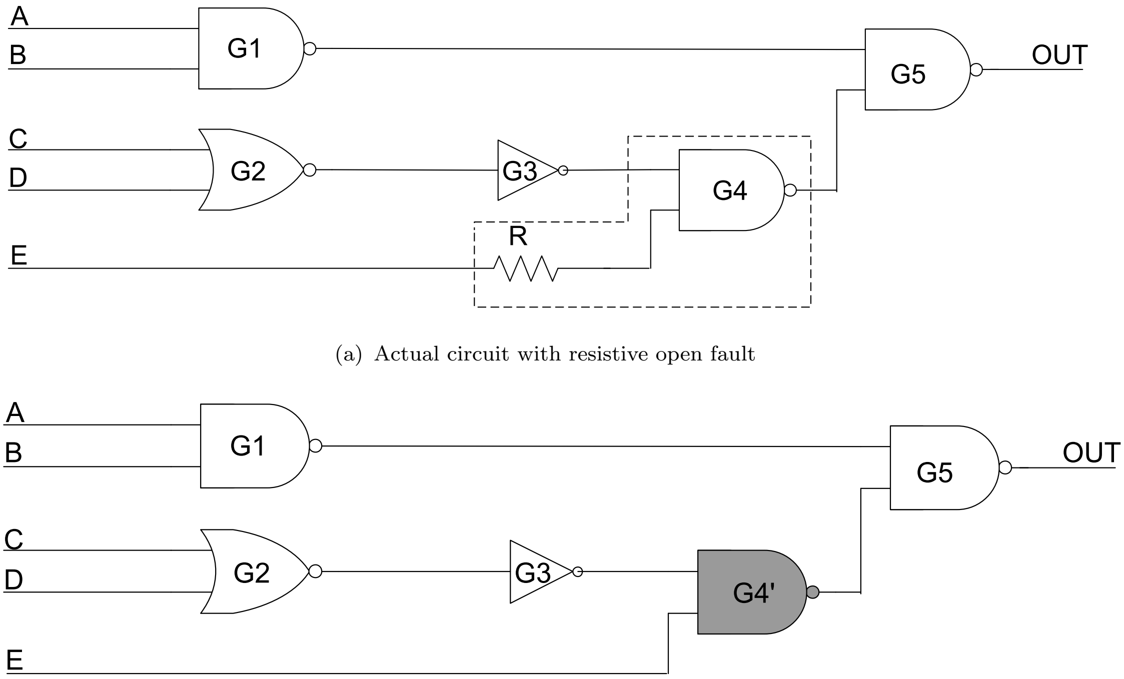
```
library IEEE;
…
…

entity CompA_Faulty is

       generic (…
                ...
                coeff_a:real :=xxx;
                coeff_b:real :=yyy;
                coeff_c:real :=zzz;
              );

       port (terminal …);

end entity CompA_faulty;

architecture abm of CompA_faulty is
…
…
  constant df: real :=(derived from delay
curve of faulty CompA);


begin
…
…
  end process;

vint2<=vstate'delayed (real2time(df));

vc==vstate'ramp(0.0225e-9);

          end architecture abm;
```

FIGURE 7.1: general VHDL-AMS

gates were used to derive the model. The delay model was then used in VHDL-AMS and the results were later compared with transistor level simulations.

7.3 Resistive Open Defects

As has been observed in previous chapters, an open defect can cause an additional delay in a transitions as well as a logical error. It has been shown that the value of resistive open as well as the supply voltage will have its impact of the circuit behaviour.



(a) Actual circuit with resistive open fault

(b) Circuit with fault model incorporated into gate G4

FIGURE 7.2: Faulty circuit representation in VHDL-AMS

In addition to the previous observations, we have also studied the impact of fan-out on the defective gate. The number of fan-outs of a CMOS gate will impact on the gate delay of the circuit. For the case of fan-in, the underlying assumption is that we are using standard cells thus the impact of fan-in will be taken care of by modelling the delay individually for different types of gates. However, the impact of fan-out has to be taken into consideration since a different fan-out means different loading capacitance. As the capacitive loading changes the gate delay of CMOS gate will also change [135]. Delay models for 1, 2 and 4 fan-out have been derived to model different loading capacitance.

7.3.1 Deriving A Delay Model of NAND gate with one fan-out

A 2-input NAND gate was used to derive the delay model. The NAND gate was designed following the ST 0.12μm technology. The clock frequency was set at

55Mhz for these simulations. Inverters were used as input buffers as well as load for the circuit. This is depicted in Figure 7.3. Gate G1 and G2 are the input buffers and G4 is the output buffer . Gate G4 also act as load for the circuit.

The path from input 'b' to the NAND GATE was sensitised. The input 'a' is always set at 'LOW'. By toggling the input 'b', the output of the NAND gate will be toggling between 'HIGH' and 'LOW' thus creating rise and fall conditions. The delay was measured between the output of inverter G2 and output of NAND gate, G3.

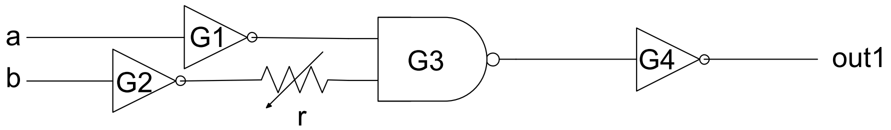
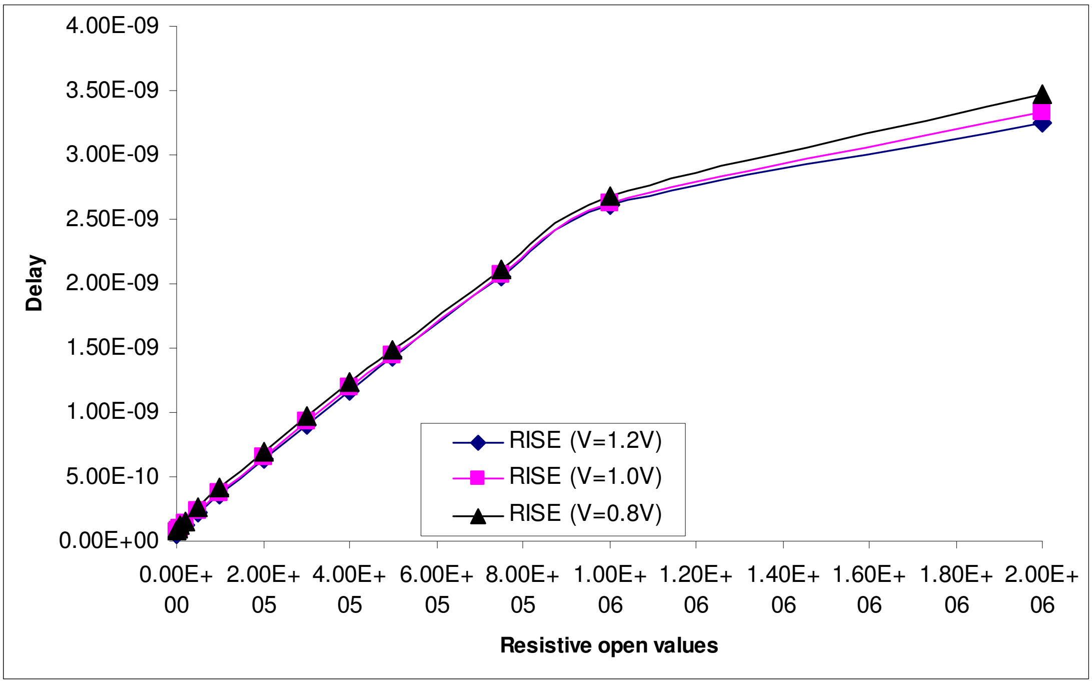


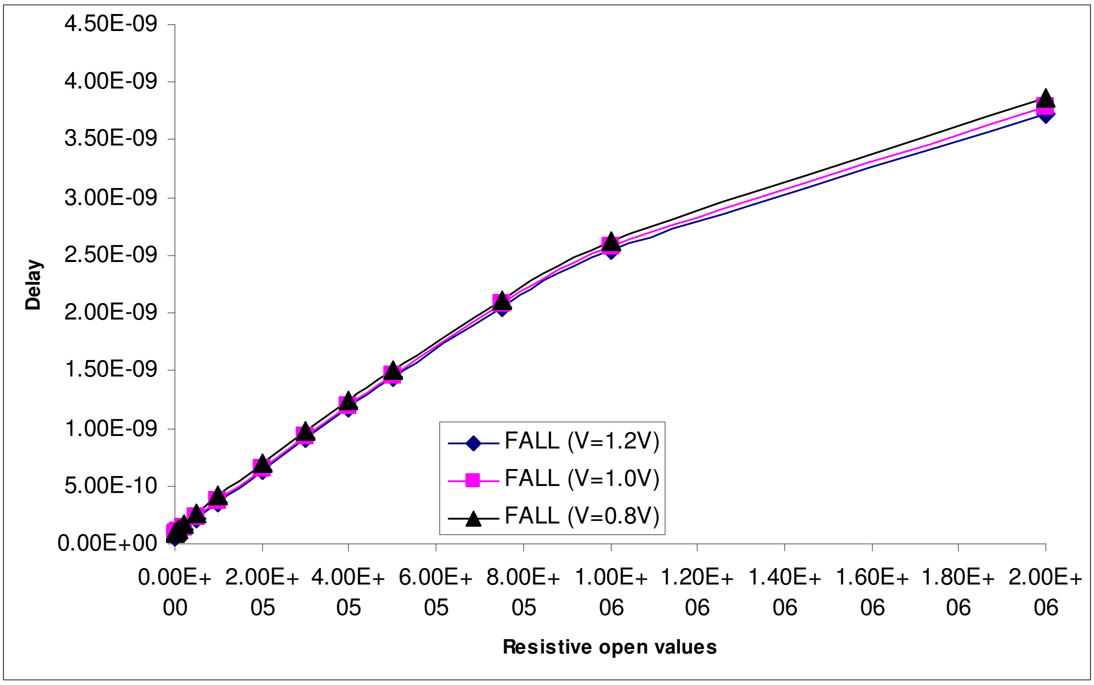
FIGURE 7.3: Transistor level fault simulation setup for a basic cell

The resistor value is varied between zero and several Megohms. The zero value represents non-faulty behaviour and is used as a benchmark to compare with other faulty conditions. The larger values of resistance will model the effect closer to an open circuit. The simulations were conducted at three different voltage levels: 0.8V, 1.0V and 1.2V.

Figure 7.4 shows the relationship between the delay at three different voltage settings at a specific resistance value. The rise and fall delays are measured between the output of gate G2('input') and output of gate G3('output'). RISE is used for the time for the signal to rise from 50% of LOW input to 50% of HIGH at the output of gate G3. Similarly, FALL is used to measure the time for the signal to drop from HIGH to LOW. It is evident from Figure 7.4 that the delay increases as the value of the resistance increases. As we further increase the value of resistance above 2 M Ω, the fault causes a logical error for all cases and we categorise them as stuck-at-fault(SF). As a result, we have limited the value of resistive fault up to 2 M Ω in our studies. As expected, the delay at lower voltages is always higher than the delay at higher voltages. This is true for both rise and fall cases.



(a) Rise Delay



(b) Fall Delay

FIGURE 7.4: Relation between delay and resistive open values for 3 different supply voltages

There are six curves in Figure 7.4 which represent 3 cases of rising delay and 3 cases of falling delay. It would be repetitive to obtain the delay model for all the six cases. If we can choose the best case that will give the highest fault coverage, it will suffice to model the fault for that case. Table 7.1 shows the delay ratio for each case. Ratios were measured by comparing the absolute delay values of the faulty conditions to the non-faulty cases where the resistance values are zero. From this table, we observe that the fall and rise cases for the highest voltage gave the largest ratio. For this reason we have used the fall case at 1.2V for delay modelling.

Table 7.1: Delay Ratio for rising delay of NAND gate for different VDD

	Vdd=1.2V		**Vdd=1.0V**		**Vdd=0.8V**	
RES (Ω)	**RISE RATIO**	**FALL RATIO**	**RISE RATIO**	**FALL RATIO**	**RISE RATIO**	**FALL RATIO**
0.00	1.00	1.00	1.00	1.00	1.00	1.00
100.00	1.01	1.01	1.01	1.01	1.01	1.01
1.00k	1.15	1.13	1.14	1.10	1.12	1.08
5.00k	1.72	1.58	1.66	1.50	1.57	1.40
10.00k	2.38	2.12	2.25	1.95	2.09	1.78
20.00k	3.64	3.18	3.35	2.84	3.04	2.49
50.00k	7.22	6.27	6.42	5.42	5.56	4.47
100.00k	12.99	11.27	11.35	9.58	9.50	7.63
200.00k	24.28	21.10	20.97	17.71	17.15	13.75
300.00k	35.42	30.83	30.50	25.76	24.68	19.79
400.00k	46.51	40.51	39.92	33.75	32.19	25.75
500.00k	57.51	50.07	49.36	41.61	39.64	31.60
750.00k	84.24	72.26	72.15	59.98	57.73	45.33
1.00M	107.86	90.43	92.47	75.11	73.98	56.78
2.00M	135.28	133.33	118.03	111.45	96.98	84.60

Using the fall case for 1.2V, best fitting regression curve was searched. In this exercise, the objective was to achieve the lowest sum of square relative error.

It is found that the best fitting regression curve is NIST MGH09 With Offset [145]. The general curve equation is given in equation 7.1 where y is the value of delay and x is the value of resistance.

$$y = \frac{a(x^2 + bx)}{(x^2 + cx + d)} + e \tag{7.1}$$

The coefficients for equation 7.1 are:

a = 1.3442817920651515E-09

b = 1.1339297457734438E+07

c = 1.7259136506074388E+05

d = 5.6456895070085156E+12

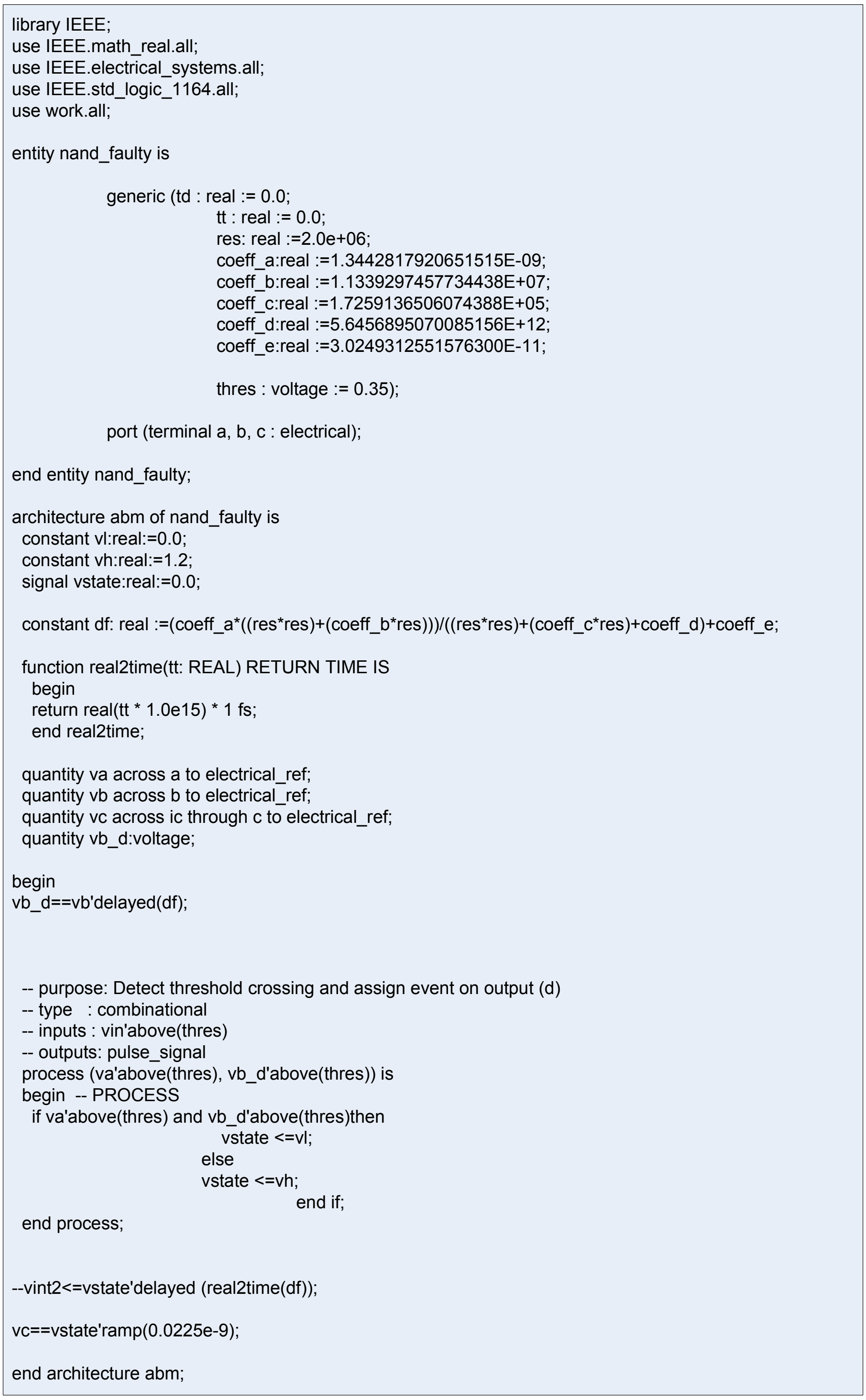
e = 3.0249312551576300E-11

Equation 7.1 with the coefficients was used to calculate the predicted value of the delay. The actual value and the predicted value were further validated by looking at the absolute error and error percentage. This is shown in Table 7.2. The table shows that the maximum error percentage is 11.21% and the minimum error percentage is around 0%. The average error percentage for the 14 sets (without the fault free case) of data is around 0.90%.

Figure 7.5 shows the VHDL-AMS code for the faulty NAND gate. Equation 7.1 with the coefficients obtained are used to model the delay for the defective gate.

One of the well accepted methods to evaluate the accuracy of the results is by observing the value of the coefficient of determination , R^2 [146]. The coefficient of determination ,R^2, explains how much of the variability in the y's can be explained by the fact that they are related to x. In other words, how close the points are to the actual line. The ideal case will have R^2 of 1.0 . The equation to calculate R^2 is shown in Equation 7.2.

$$R^2 = 1 - \frac{SSRes}{SSTotal} \tag{7.2}$$



```
library IEEE;
use IEEE.math_real.all;
use IEEE.electrical_systems.all;
use IEEE.std_logic_1164.all;
use work.all;

entity nand_faulty is

                  generic (td : real := 0.0;
                                      tt : real := 0.0;
                                      res: real :=2.0e+06;
                                      coeff_a:real :=1.3442817920651515E-09;
                                      coeff_b:real :=1.1339297457734438E+07;
                                      coeff_c:real :=1.7259136506074388E+05;
                                      coeff_d:real :=5.6456895070085156E+12;
                                      coeff_e:real :=3.0249312551576300E-11;

                                      thres : voltage := 0.35);

                  port (terminal a, b, c : electrical);

end entity nand_faulty;

architecture abm of nand_faulty is
  constant vl:real:=0.0;
  constant vh:real:=1.2;
  signal vstate:real:=0.0;

  constant df: real :=(coeff_a*((res*res)+(coeff_b*res)))/((res*res)+(coeff_c*res)+coeff_d)+coeff_e;

  function real2time(tt: REAL) RETURN TIME IS
    begin
    return real(tt * 1.0e15) * 1 fs;
    end real2time;

  quantity va across a to electrical_ref;
  quantity vb across b to electrical_ref;
  quantity vc across ic through c to electrical_ref;
  quantity vb_d:voltage;

begin
vb_d==vb'delayed(df);



  -- purpose: Detect threshold crossing and assign event on output (d)
  -- type   : combinational
  -- inputs : vin'above(thres)
  -- outputs: pulse_signal
  process (va'above(thres), vb_d'above(thres)) is
  begin  -- PROCESS
    if va'above(thres) and vb_d'above(thres)then
                                     vstate <=vl;
                                    else
                                    vstate <=vh;
                                                     end if;
  end process;


--vint2<=vstate'delayed (real2time(df));

vc==vstate'ramp(0.0225e-9);

end architecture abm;
```

Figure 7.5: VHDL-AMS code for NAND gate with resistive open

SSRes is defined as residual sum of squares which is the sum of squared absolute error in our case. SSTotal is defined as total sum of squares of the actual values. The sum of squared absolute error was 1.68e-22 and total sum of squares was 2.72e-17. This will give the coefficient of determination ,R^2, of 0.99999382. Since the ideal case will have R^2 of 1.0, this shows the regression curve is a very close approximation to the actual value.

Table 7.2: Error ratio for Regression Equation for Fall at VDD=1.2V for a 2-input NAND gate with one fan-out

RES (Ω)	Actual Value	Predicted Value	Absolute Error	Error Percentage
0.00	2.72E-11	3.02E-11	3.05E-12	11.21
100.00	2.75E-11	3.05E-11	3.02E-12	10.98
1.00k	3.07E-11	3.29E-11	2.25E-12	7.33
5.00k	4.30E-11	4.38E-11	7.53E-13	1.75
10.00k	5.76E-11	5.73E-11	-3.36E-13	0.58
20.00k	8.65E-11	8.43E-11	-2.19E-12	2.54
50.00k	1.71E-10	1.66E-10	-5.42E-12	3.17
100.00k	3.06E-10	3.01E-10	-4.68E-12	1.53
200.00k	5.74E-10	5.73E-10	-1.39E-12	0.24
300.00k	8.38E-10	8.41E-10	3.30E-12	0.39
400.00k	1.10E-09	1.10E-09	4.75E-12	0.43
500.00k	1.36E-09	1.36E-09	5.23E-13	0.04
750.00k	1.96E-09	1.95E-09	-6.55E-12	0.33
1.00M	2.46E-09	2.46E-09	3.05E-12	0.12
2.00M	3.62E-09	3.62E-09	-1.19E-13	0.00

7.3.2 Deriving Delay Model of NAND gate with two and four fan-outs

Using a modified setup as in Figure 7.3, delay models for a NAND gate with two and four fan-outs were derived.

The number of output inverters were increased to two and four to simulate two fan-out cases and four fan-out cases respectively. For each case, simulations were conducted at three different supply voltages. Gate delays for both rising and falling cases were observed. In order to find the best case that will give best fault coverage, delay ratios were calculated.

Simulation results show that the case for fall at 1.2V gives the best ratio for both two and four fan-outs. This is similar to the case in Section 7.3.1. It is also found that the best fitting regression curve is NIST MGH09 With Offset as in equation 7.1.

For the case of two fan-outs the coefficient values for the equations are:

a = 1.4568075423528656E-08

b = 9.6821377575852648E+02

c = 4.9456206276761824E+06

d = 1.4347573664032402E+09

e = 3.7999893944894651E-11

Table 7.3 shows the details with regard to the predicted value using the NIST MGH09 With Offset equation for the two fanout cases. The sum of squares absolute error was 1.37e-21 and total sum of squares was 1.46e-17. This will give the coefficient of determination ,R^2, a very close value to 1 (0.99990616).

For the case of four fan-outs, the coefficient values for equation 7.1 are:

a = 1.3442817920651515E-09

b = 1.1339297457734438E+07

c = 1.7259136506074388E+05

d = 5.6456895070085156E+12

e = 3.0249312551576300E-11

Using equation 7.1, and the coefficient, the value of sum of squared absolute error was 3.42e-22 and total sum of squares was 2.94e-17.This will give the coefficient of determination ,R^2, a very close value to 1.0 (0.99998836).

Table 7.3: Error ratio for Regression Equation for Fall at VDD=1.2V for a 2-input NAND gate with 2 fan-out

RES (Ω)	**Actual Value**	**Predicted Value**	**Absolute Error**	**Error Percentage**
0.00	3.80E-11	3.80E-11	-1.06E-16	0.00
100.00	3.83E-11	3.88E-11	5.06E-13	1.32
1.00k	4.18E-11	4.25E-11	6.93E-13	1.66
5.00k	5.46E-11	5.46E-11	2.17E-16	0.00
10.00k	6.96E-11	6.93E-11	-2.64E-13	0.38
20.00k	9.97E-11	9.86E-11	-1.06E-12	1.06
50.00k	1.86E-10	1.86E-10	-2.17E-13	0.12
100.00k	3.25E-10	3.29E-10	3.70E-12	1.14
200.00k	5.97E-10	6.06E-10	9.18E-12	1.54
300.00k	8.63E-10	8.73E-10	1.01E-11	1.17
400.00k	1.13E-09	1.13E-09	7.09E-20	0.00
500.00k	1.39E-09	1.38E-09	-1.25E-11	0.90
750.00k	1.99E-09	1.96E-09	-3.18E-11	1.60
1.00M	2.49E-09	2.49E-09	0.00	0.00

In both two fan-out and four fan-out cases, the derived regression curves are very close approximations to the actual value. In the next section, the model for 2 input NOR gate will be derived.

7.3.3 Deriving Delay Model of NOR gate with one fan-out

As for the NAND gate in Section 7.3.1, a 2 input NOR gate was used to derive the delay model. The simulation setup is shown in Figure 7.6. The path from input 'b' to the NOR GATE was sensitised. The input 'a' is always set at 'HIGH'. By toggling the input 'b', output of the NOR gate will be toggling between 'LOW' and 'HIGH' thus creating rise and fall conditions. The delays were measured between the output of inverter G2 and output of NOR gate, G3.

The relation between delays at three different voltage settings is shown in Figure 7.7. Figure 7.7(a) shows the rise delay for the NOR gate and Figure 7.7(b) shows the fall delay. As expected, the delay value increases with the resistance. It has

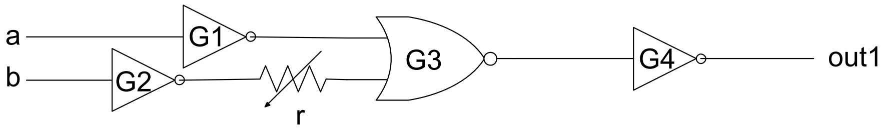


Figure 7.6: Transistor level fault simulation setup for a basic cell

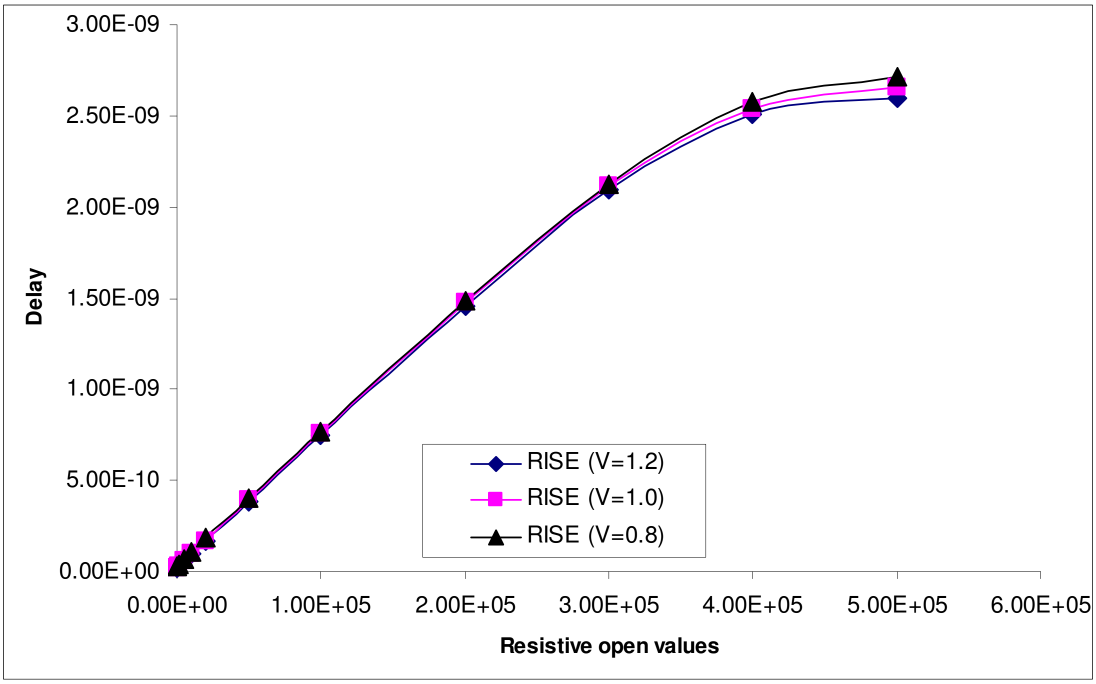
also been observed that the delay causes stuck-at-fault (SF) behaviour at smaller values of resistance than for the NAND gate. Since resistance values above 500K Ω cause stuck-at-fault behaviour, we have limited our studies for the NOR gate up to 500K Ω only.

In order to find the best case to model the delay, the gate delay ratios were calculated. Table 7.4 shows the delay ratio for all six cases. It can be observed that the fall case for the highest voltage has the highest delay ratio. For this reason we have used the fall case at 1.2V for delay modelling.

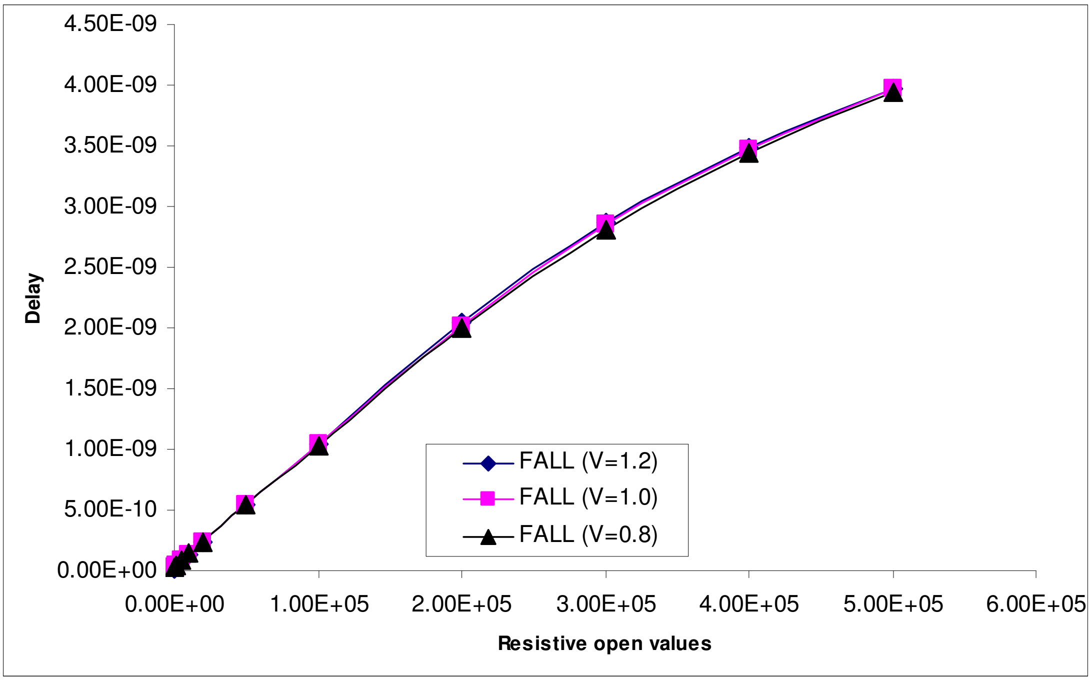
Table 7.4: Delay Ratio for rising delay of NOR gate for different VDD

	Vdd=1.2V		**Vdd=1.0V**		**Vdd=0.8V**	
RES Ω	**RISE RATIO**	**FALL RATIO**	**RISE RATIO**	**FALL RATIO**	**RISE RATIO**	**FALL RATIO**
0.00	1.00	1.00	1.00	1.00	1.00	1.00
100.00	1.06	1.07	1.05	1.06	1.04	1.04
1.00k	1.52	1.64	1.46	1.53	1.36	1.40
5.00k	3.41	3.82	3.06	3.36	2.61	2.78
10.00k	5.68	6.39	4.97	5.49	4.04	4.36
20.00k	10.15	11.45	8.72	9.69	6.82	7.44
50.00k	23.41	26.54	19.82	22.18	15.01	16.56
100.00k	45.34	51.62	38.21	42.89	28.56	31.71
200.00k	88.59	100.39	74.52	83.23	55.39	61.15
300.00k	127.02	141.16	106.94	117.39	79.51	86.32
400.00k	152.16	171.32	128.67	143.02	96.22	105.58
500.00k	157.62	195.04	134.30	163.33	101.53	121.01

We found that the best fitting regression curve is NIST MGH09 with an offset which is similar to the model used for the NAND gate delay. The coefficient values are:



(a) Rise Delay



(b) Fall Delay

Figure 7.7: Relationship between delay and resistive open values for 3 different supply voltages

a = 5.0689954752511920E-09

b = 5.4734337456823164E+05

c = 2.9313666929908784E+05

d = 2.7250910292711859E+11

e = 3.8309501836100739E-11

Using the coefficient values and equation 7.1, the predicted values are calculated. These values and corresponding errors details are shown in Table 7.5. From the table, the maximum error percentage is 4.55% The average error percentage for all cases not including the fault free case is 0.19%. The sum of squared absolute error was 4.06e-20 and total sum of squares of actual values are 4.18e-17. This will give the coefficient of determination ,R^2, a very close value to 1 (0.99990). This shows the regression curve is a very close approximation to the actual value.

TABLE 7.5: Error ratio for Regression Equation for Fall at VDD=1.2V for NOR GATE

RES Ω	**Actual Value**	**Predicted Value**	**Absolute Error**	**Error Percentage**
0.00	2.04E-11	2.03192E-11	-8.08532E-14	0.40
100.00	2.18E-11	2.19634E-11	1.6342E-13	0.75
1.00k	3.35E-11	3.31833E-11	-3.16714E-13	0.95
5.00k	7.79E-11	7.73689E-11	-5.31151E-13	0.68
10.00k	1.3E-10	1.31672E-10	1.67194E-12	1.29
20.00k	2.33E-10	2.39032E-10	6.03189E-12	2.59
50.00k	5.4E-10	5.52365E-10	1.23647E-11	2.29
100.00k	1.05E-09	1.04745E-09	-2.54958E-12	0.24
200.00k	2.04E-09	1.94714E-09	-9.28632E-11	4.55
300.00k	2.87E-09	2.74351E-09	-1.26492E-10	4.41
400.00k	3.49E-09	3.4534E-09	-3.66048E-11	1.05
500.00k	3.97E-09	4.09016E-09	1.20161E-10	3.03

7.3.4 Deriving Delay Model of NOR gate with two fan-out and four fan-out

A similar setup as in Section 7.3.3 were used. For NOR gates with two and four fan-out, the number of output inverters was increased to two and four inverters respectively. For both cases the best fitting curve was NIST MGH09 with offset. The coefficients for both cases are shown in Table 7.6.

Table 7.6: Coefficient value for two and four fan-out of NOR gate

Coefficient	Two Fan-out	Four fan-out
a	9.3954614039374084E-09	5.0689954752511920E-09
b	2.7541555231155633E+02	5.4734337456823164E+05
c	1.1602410012328825E+06	2.9313666929908784E+05
d	1.7955386051198661E+08	2.7250910292711859E+11
e	2.0978543263559347E-11	3.8309501836100739E-11

The equations were then validated and the R^2 value for both cases were very close to 1.0.

7.3.5 Adjustment for Single Fault Assumptions

In the previous sections, we have derived the delay model for 2 basic gates. In the actual case, delay model has to be derived for all types of basic cell used in the circuit under test. The defect can occur at any of the gates. Under the single fault assumption, we can only have one fault in the system at one point of time. In order to use this assumption within our derived model, each gate will be assigned a unique identifier, k_i.

In equation 7.3, k_1, k_2 to k_n each represent different gates. Only one fault is activated at a time by setting one of the k_i values to '1' and the other values of k_i

are set to ’0’.

$$\begin{aligned}\text{Fall Delay} &= k_1(a_1R_1^2+b_1R_1+c_1)\\ &+k_2(a_2R_2^2+b_2R_2+c_2)\\ &+k_3(a_3R_3^2+b_3R_3+c_3)\\ &+\ldots\\ &+k_n(a_nR_n^2+b_nR_n+c_n)\end{aligned} \tag{7.3}$$

For example, let us say k_2 represents a particular NAND gate. To activate the fault at this gate, rest of the k_i will be set to 0. This will give the fall delay as

$$\text{Fall Delay} = k_2(a_2R_2^2+b_2R_2+c_2) \tag{7.4}$$

7.4 Resistive Short Defects

A resistive short can cause a circuit to have either an additional delay or a logical error. As for the resistive open faults the two main constraints that will impact the behaviour of the defect are the value of the resistance and the supply voltage. In this work, a simplified model of resistive short defects was studied by observing the behaviour of the circuit with a resistor connected to the ground signal. The simulation setup circuit is shown in Figure 7.8.

The primary input ’IN’ was toggled from HIGH to LOW to sensitise the path from ’IN’ to the primary output, ’OUT’. The delay was measured between ’in2’ and ’in3’.

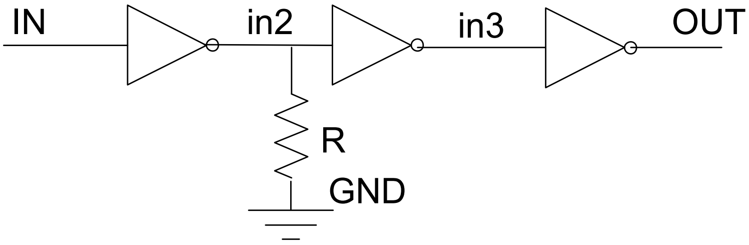


Figure 7.8: Transistor level simulation setup for resistive short

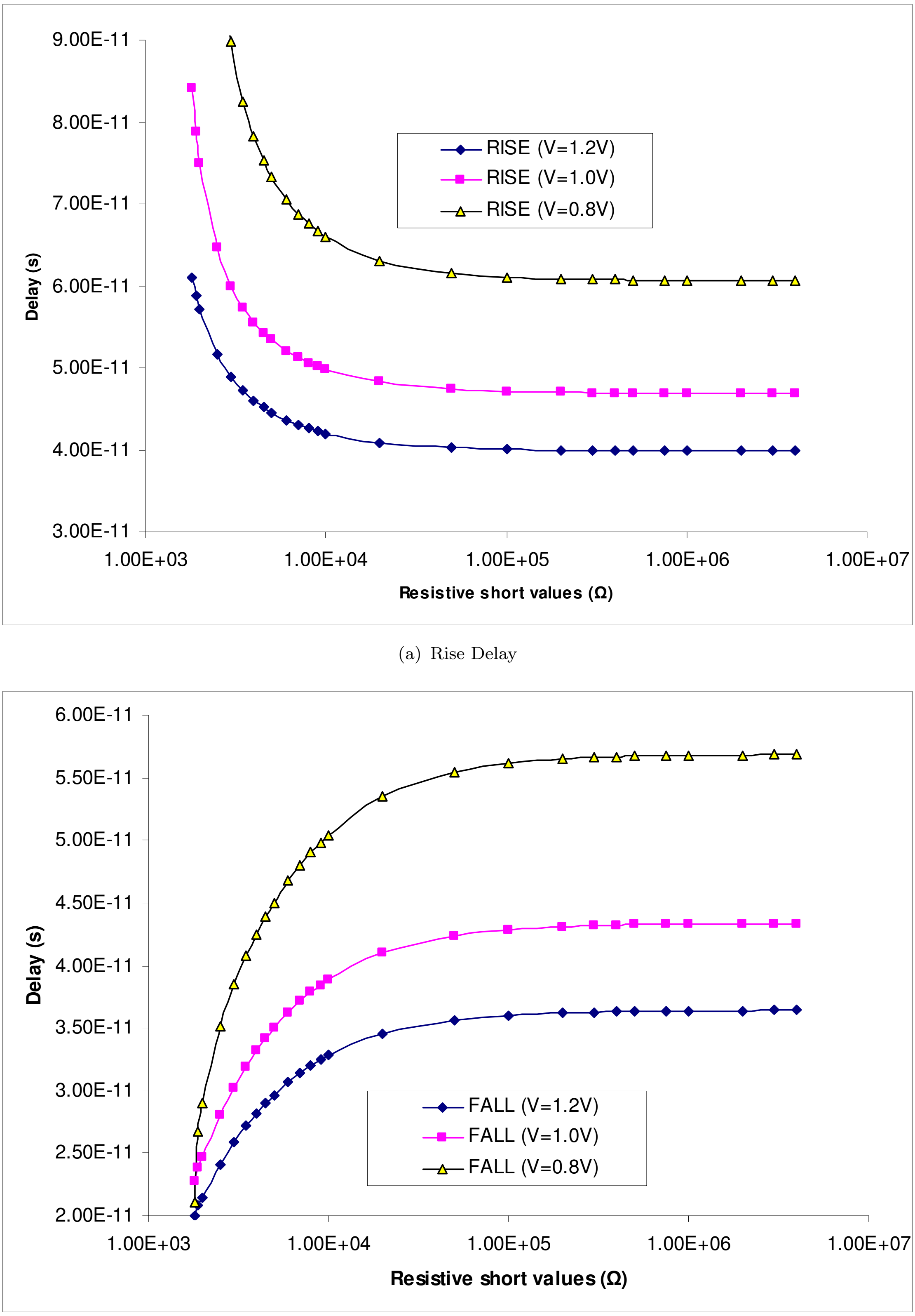
Figure 7.9 shows the delay results for rise and fall cases. For the rising case, 'in2' was set to rise which will cause 'in3' to fall. Since 'in2' is shorted to ground, this will cause an extra delay for the signal to rise from LOW to HIGH. For the falling case, 'in2' was set to fall and 'in3' will rise. For this case, there will not be an extra delay since the shorted resistor will be always pulling to the ground. As a result, only results from rising cases are useful for our studies.

Table 7.7 shows the relations between the delay at three different voltage settings for specific values of resistive shorts. The smaller and close to zero value of resistance represent a circuit shorted to ground and will have high impact. The larger value of resistance models the non-faulty conditions. It can be observed from Table 7.7, that as the value of the supply voltage decreases, resistance value that cause the circuit to exhibit stuck-at-fault (SF) behaviour increases. The values are 1100 Ω, 1300 Ω and 1700 Ω for 1.2V, 1.0V and 0.8V supply voltage respectively.

Table 7.7 also shows the ratio for all three rising cases. The ratio increasing for lower supply voltage. Thus, we have choose the rise case for 0.8V to model the delay for a resistive short. Using lowest sum of square relative error as the objectives, the best fitting regression curve were NIST MGH09 [145] .

The general curve equation is given as in equation 7.5 with y is the value of delay and x is the value of resistance.

$$y = \frac{a(x^2 + bx)}{(x^2 + cx + d)} \tag{7.5}$$



(a) Rise Delay

(b) Fall Delay

FIGURE 7.9: Relationship between delay and resistive open values for 3 different supply voltages

Table 7.7: Gate delay for INV gate for different VDD

	Vdd=1.2V		**Vdd=1.0V**		**Vdd=0.8V**	
RES **Ω**	**RISE DELAY**	**RISE RATIO**	**RISE DELAY**	**RISE RATIO**	**RISE DELAY**	**RISE RATIO**
1.10k	SF	-	SF	-	SF	-
1.20k	1.37E-10	3.42	SF	-	SF	-
1.30k	9.70E-11	2.43	SF	-	SF	-
1.40k	8.17E-11	2.05	1.79E-10	3.80	SF	-
1.50k	7.32E-11	1.83	1.24E-10	2.64	SF	-
1.60k	6.78E-11	1.70	1.03E-10	2.19	SF	-
1.70k	6.39E-11	1.60	9.15E-11	1.95	SF	-
1.80k	6.11E-11	1.53	8.41E-11	1.79	2.77E-10	4.55
1.90k	5.89E-11	1.47	7.89E-11	1.68	1.93E-10	3.17
2.00k	5.71E-11	1.43	7.50E-11	1.60	1.58E-10	2.60
3.00k	4.89E-11	1.22	5.99E-11	1.28	8.98E-11	1.47
5.00k	4.46E-11	1.12	5.34E-11	1.14	7.33E-11	1.20
7.00E+03	4.30E-11	1.08	5.13E-11	1.09	6.88E-11	1.13
9.00E+03	4.23E-11	1.06	5.02E-11	1.07	6.67E-11	1.09
10.00k	4.20E-11	1.05	4.98E-11	1.06	6.60E-11	1.08
50.00k	4.03E-11	1.01	4.74E-11	1.01	6.16E-11	1.01
200.00k	3.99E-11	1.00	4.70E-11	1.00	6.09E-11	1.00

The coefficients for equation 7.5 are:

a = 6.2265324054736917E-11

b = -1.0963230706417353E+03

c =-1.6623322343549310E+03

d = 3.6914302219389559E+04

Using equation 7.5, the expected values and predicted values are calculated and shown in Table 7.8. In addition, the absolute error and the error percentage are also presented. From this table, the largest error percentage is 2.49% and the average error percentage is 1.12%. The sum of squares absolute error is 6.0385e-22 and total sum of squares is 2.58e-19. This gives an R^2 value very close to 1.00. This validates the regression curve used in our model.

Table 7.8: Error ratio for Regression Equation for Fall at VDD=0.8V for resistive short

RES (Ω)	Actual Value	Predicted Value	Absolute Error	Error Percentage
1.80E+03	2.77E-10	2.77E-10	-1.03E-25	0.000
1.90k	1.93E-10	1.95E-10	1.64E-12	0.850
2.00k	1.58E-10	1.58E-10	8.22E-23	0.000
2.50k	1.05E-10	1.03E-10	-2.47E-12	2.352
3.00k	8.98E-11	8.78E-11	-2.00E-12	2.227
3.50k	8.25E-11	8.10E-11	-1.52E-12	1.842
4.00k	7.82E-11	7.70E-11	-1.16E-12	1.483
4.50k	7.54E-11	7.45E-11	-9.30E-13	1.233
5.00k	7.33E-11	7.27E-11	-6.36E-13	0.868
6.00k	7.05E-11	7.03E-11	-2.10E-13	0.298
7.00k	6.88E-11	6.88E-11	-2.15E-21	0.000
8.00k	6.76E-11	6.78E-11	1.77E-13	0.262
9.00k	6.67E-11	6.70E-11	3.31E-13	0.496
10.00k	6.60E-11	6.65E-11	4.63E-13	0.702
20.00k	6.31E-11	6.42E-11	1.08E-12	1.712
50.00k	6.16E-11	6.30E-11	1.39E-12	2.256
100.00k	6.11E-11	6.26E-11	1.52E-12	2.488

7.5 Simulation Results

In this section the derived delay models were used in different circuits to verify the accuracy of the models. The models were used in mixed-signal VHDL-AMS simulations. These results were then compared with results obtained from circuit level SPICE simulation. Appendix C shows the VHDL-AMS code used for this simulation.

7.5.1 Verifying Resistive opens for NAND gate with one fan-out

A full adder circuit with resistive opens, R as in Figure 7.10, was simulated both at circuit level and mixed-signal VHDL-AMS level.

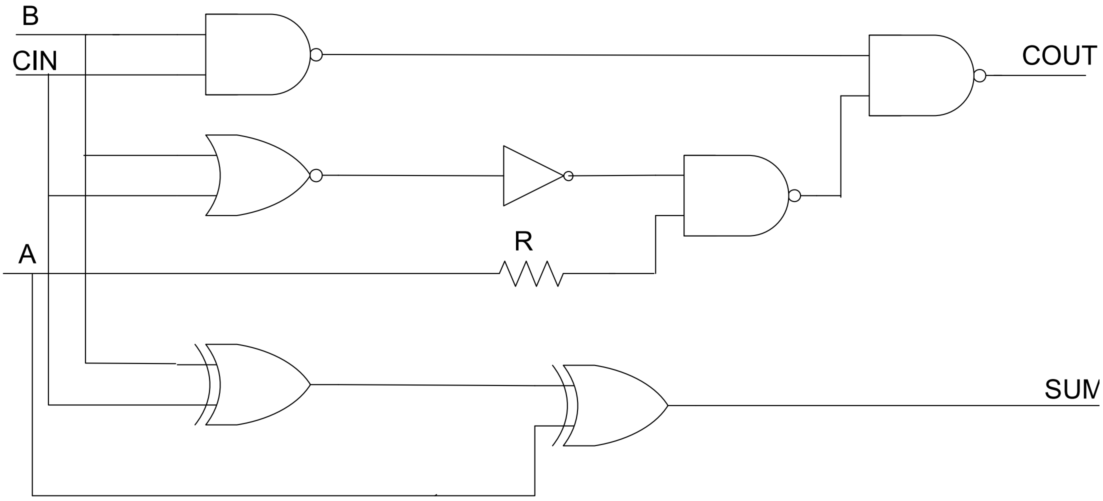


Figure 7.10: Full Adder with Resistive Open

At the SPICE level, the actual path delays between input A and output COUT were measured for different values of resistive opens. In other words, the combined delays due to the individual gate delays, together with any loading effects were measured.

In the VHDL-AMS model, the delay model for the NAND gate from Section 7.3.1 was used and the path delay was measured. The actual path delay is measured from the time when the input signals reach 50% of supply voltage to the time the output signals reaches 50% of supply voltage.

Figure 7.11 shows the path delay for the full adder circuit for the FALL case at 1.2V. It is observed that the delay generated from the VHDL-AMS model corresponds well to the actual measured delay. The detailed values with the error percentage are shown in Table 7.9. The error percentage is between 0.06% and 12.73%. The sum of squares absolute error is 5.64e-20 and total sum of squares of delay from spice is 3.07e-17. This gives R^2 value of 0.9982. The R^2 value very close to 1.00 shows that the values are a very close approximation to the actual delay values.

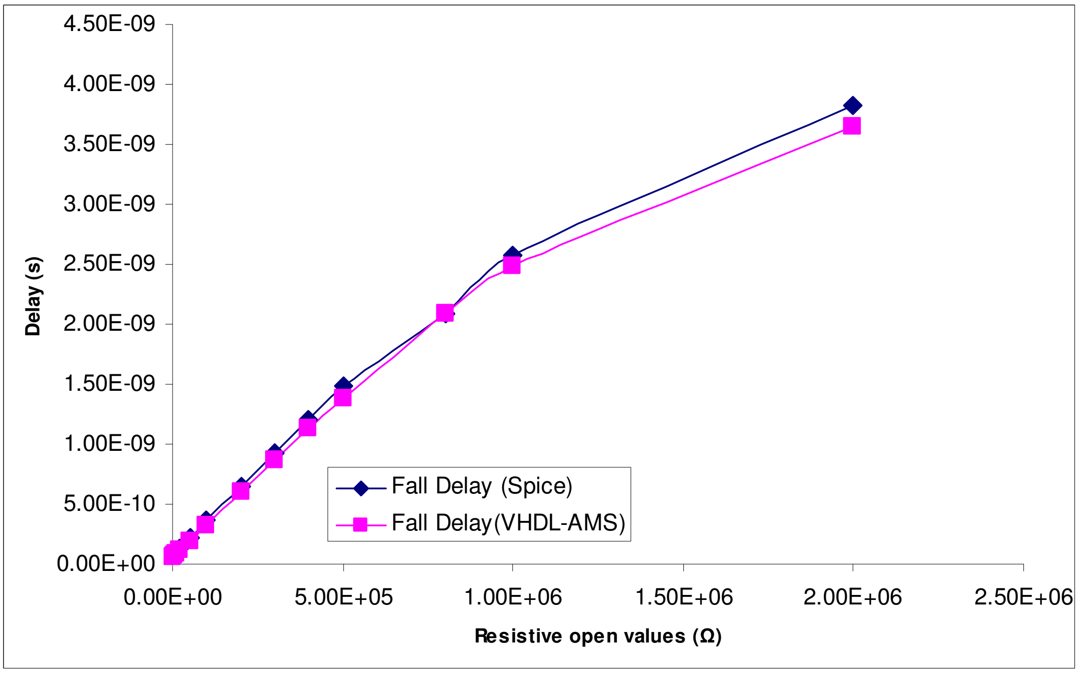


FIGURE 7.11: Path delay from SPICE and VHDL-AMS for full adder with resistive open at NAND gate

7.5.2 Verifying Resistive open for NAND gate with two fan-out and four fan-out

For the case of a NAND gate with two fan-outs, we have used the ISCAS85 c17 benchmark circuit. The circuit with the resistive open fault is shown in Figure 7.12.

Similar to Section 7.5.1, results from both SPICE simulations and VHDL-AMS simulations were compared. Table 7.10 shows the results from both simulations and the error percentage. The results were then plotted and can be seen in Figure 7.13. It has been observed that the error percentage gets larger as the value of the resistance gets bigger. As the value of the resistance gets closer to an open circuit, the behaviour of the circuit is close to a stuck-at fault circuit. As our model was derived to handle delay faults only , this causes a large difference between the actual and model based simulations.

Table 7.9: Error difference for adder with faulty NAND gate: one fan-out

RES	**Delay (SPICE)**	**Delay(VHDL-AMS)**	**Error%**
0.00	6.63E-11	5.81E-11	12.41
100.00	6.66E-11	5.81E-11	12.73
1.00k	6.89E-11	6.17E-11	10.42
5.00k	8.00E-11	7.26E-11	9.33
10.00k	9.50E-11	8.61E-11	9.44
20.00k	1.26E-10	1.13E-10	10.03
50.00k	2.15E-10	1.94E-10	9.69
100.00k	3.61E-10	3.29E-10	8.84
200.00k	6.45E-10	6.02E-10	6.77
300.00k	9.27E-10	8.70E-10	6.13
400.00k	1.21E-09	1.13E-09	5.92
500.00k	1.48E-09	1.39E-09	6.00
800.00k	2.09E-09	2.09E-09	0.06
1.00M	2.57E-09	2.49E-09	2.93
2.00M	3.82E-09	3.65E-09	4.58

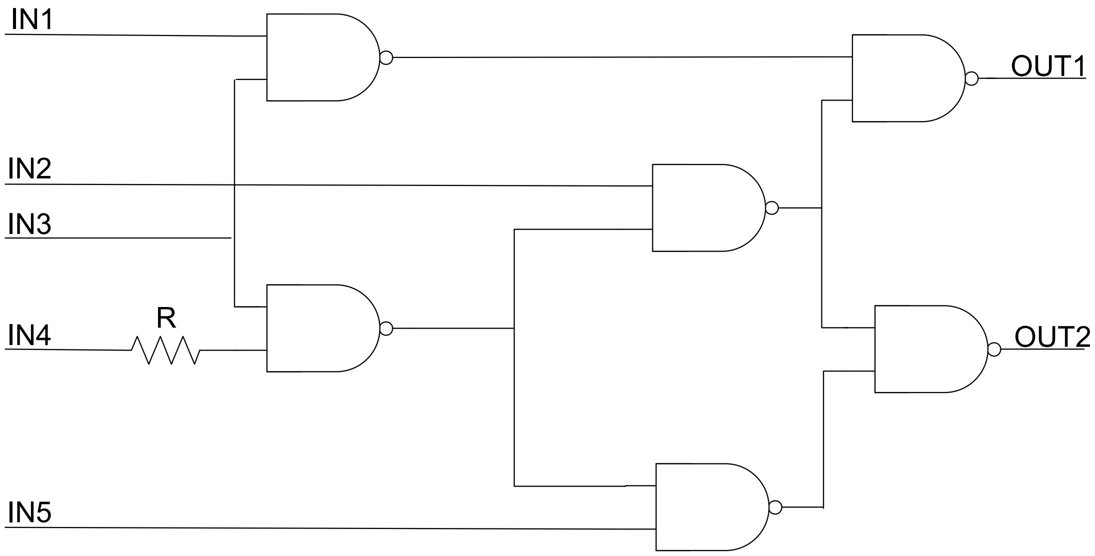


Figure 7.12: C17 with resistive open

For the case of four fan-outs, the standard C17 benchmark circuit was modified as shown in Figure 7.14. This modified C17 gates now has four fan-outs to enable the simulation of the NAND gate with 4 fan-outs.

Detailed results for the four fan-out case are shown in Table 7.11. As for the case of two fan-outs, the error difference get larger at larger values of resistive opens for

Table 7.10: Error difference for C17 circuit with faulty NAND gate

RES	Delay (SPICE)	Delay(VHDL-AMS)	Error%
0.00	7.67E-11	8.14E-11	6.21
100.00	7.70E-11	8.25E-11	7.18
1.00k	7.99E-11	8.20E-11	2.59
5.00k	9.20E-11	8.98E-11	2.40
10.00k	1.07E-10	1.11E-10	4.55
20.00k	1.37E-10	1.35E-10	1.39
50.00k	2.26E-10	2.21E-10	2.61
100.00k	3.71E-10	3.64E-10	1.99
200.00k	6.47E-10	6.41E-10	0.90
300.00k	9.11E-10	9.08E-10	0.29
400.00k	1.12E-09	1.17E-09	4.04
500.00k	1.30E-09	1.42E-09	8.77
750.00k	1.58E-09	1.99E-09	25.95
1.00M	1.66E-09	2.53E-09	51.78
2.00M	9.27E-09	3.66E-09	60.51

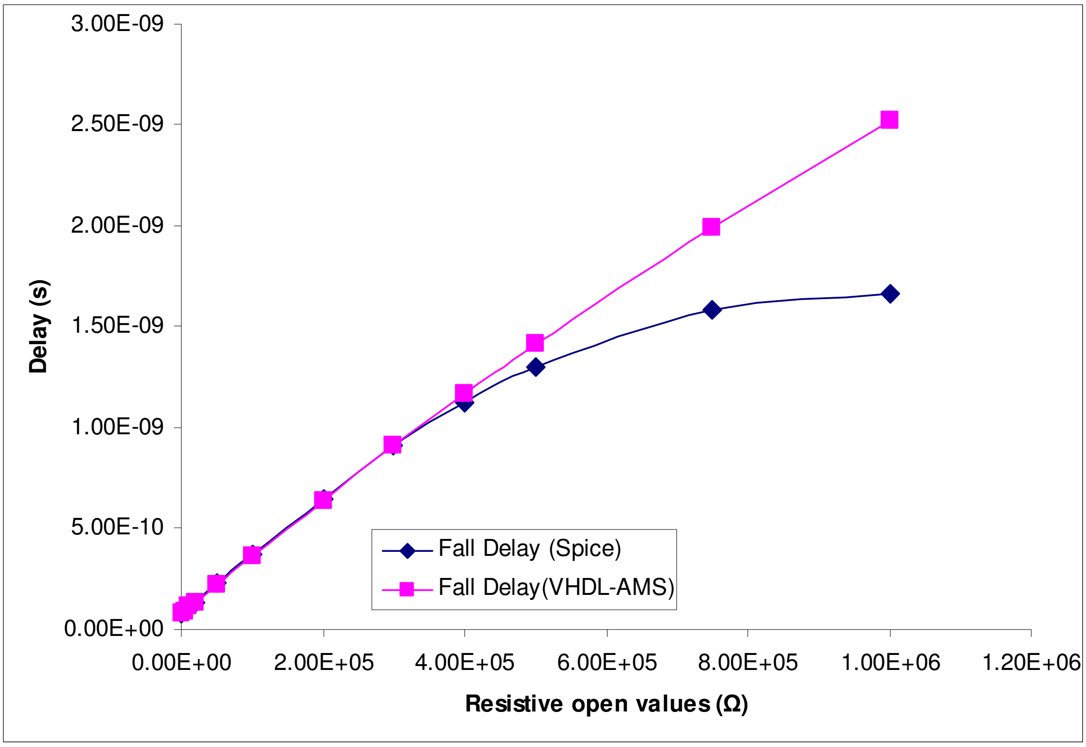


Figure 7.13: Path delay from SPICE and VHDL-AMS for C17 circuit

the same reasons as for the case of NAND gate with 2 fan-out. The model gives acceptable results up to 500K Ω. Beyond that value, the results deviate. However,

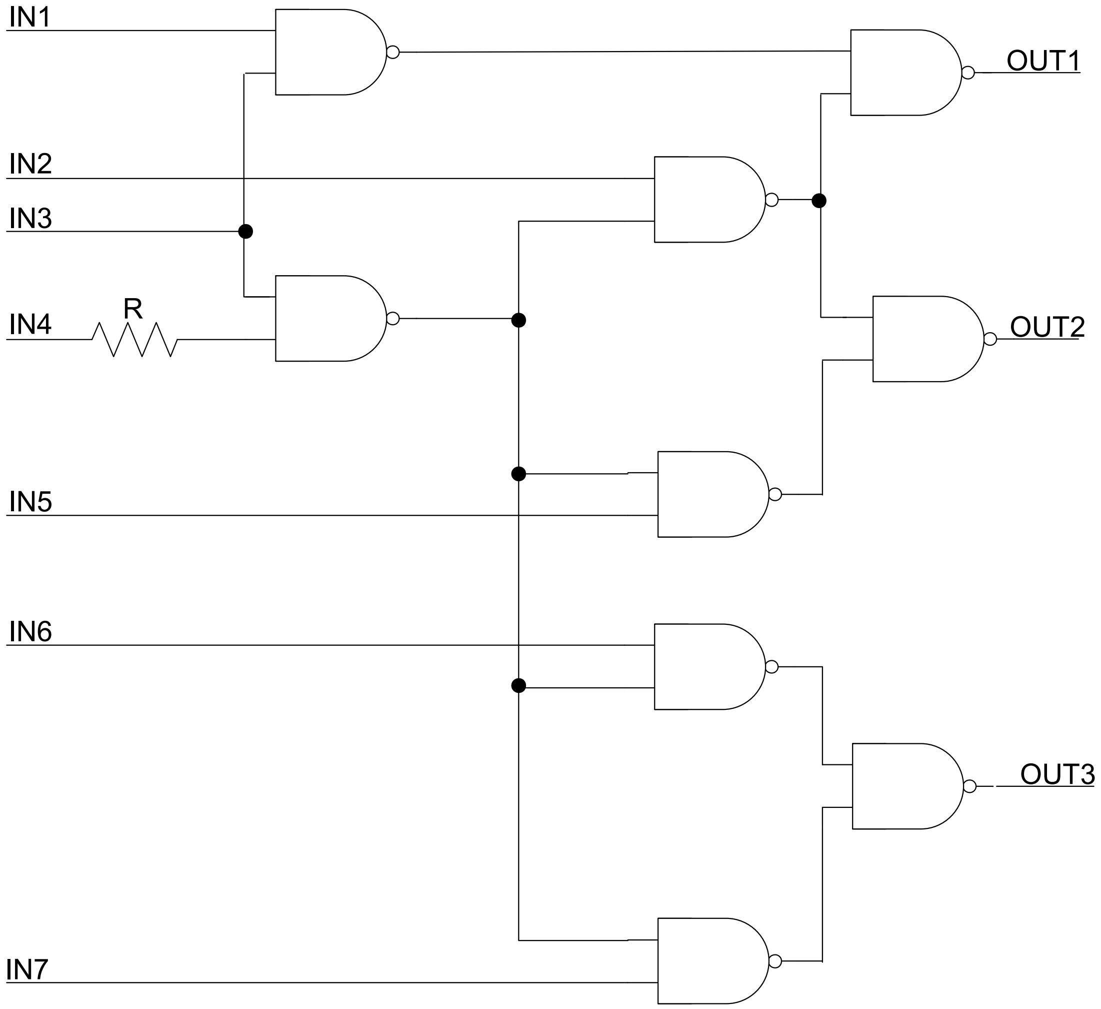


Figure 7.14: Modified C17 Circuit with 4 fan-out

in reality the fault will be regarded as stuck-at-fault.

7.5.3 Verifying Resistive open for NOR gate

Similar to the process for verifying the models for the NAND gate, the verification were done for defects at NOR gate. The same location of defect was injected at SPICE circuit and results were observed. Figure 7.15 shows the difference between the results from SPICE simulations and VHDL-AMS simulations.

Due to similarity of the setup, we have moved to the verification for NOR gate with 4 fan-out. The structure of the modified benchmark circuit C17 as shown in

Table 7.11: Error difference for modified C17 circuit with four fan-out NAND gate

RES	**Delay (SPICE)**	**Delay(VHDL-AMS)**	**Error%**
0.00	9.29E-11	9.94E-11	6.99
100.00	9.32E-11	9.98E-11	7.05
1.00k	9.62E-11	1.01E-10	5.17
5.00k	1.08E-10	1.13E-10	4.60
10.00k	1.23E-10	1.29E-10	4.26
20.00k	1.54E-10	1.56E-10	1.60
50.00k	2.46E-10	2.42E-10	1.40
100.00k	3.94E-10	3.88E-10	1.34
200.00k	6.73E-10	6.71E-10	0.28
300.00k	9.39E-10	9.50E-10	1.21
400.00k	1.15E-09	1.22E-09	6.17
500.00k	1.33E-09	1.48E-09	11.27
750.00k	1.61E-09	2.08E-09	28.89
1.00M	1.69E-09	2.58E-09	52.42
2.00M	9.32E-09	3.78E-09	59.48

Figure 7.14 were used. However all the NAND gates were replaced with 2 input NOR gates and simulation was conducted at SPICE level as well in VHDL-AMS. Results from both levels were compared and can be observed in Figure 7.16.

It can be observed from both verification processes that the models give very good approximations for resistor values up to 125K Ω. However, when the value of resistive open are larger than 150k Ω, the difference between actual SPICE simulations and results from the model was getting larger. The large resistance eventually cause the circuit to show stuck-at-fault (SF) behaviour. On the other hand, the model is a good approximation up to delay of 1.50E-09s which is equivalent to a frequency of 667MHz. For speeds below 667MHz, further refined delay models are required.

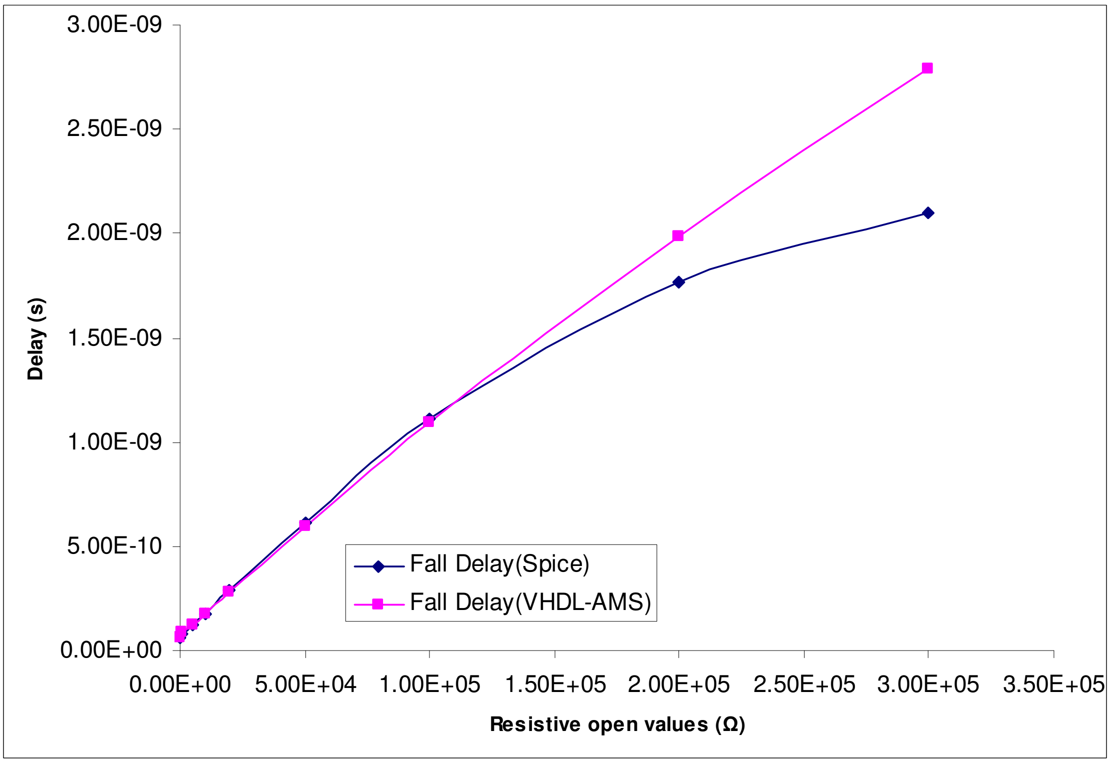


FIGURE 7.15: Path delay from SPICE and VHDL-AMS for full adder with resistive open at NOR gate with one fan-out

7.5.4 Verifying Resistive Short

The model derived for resistive short was verified using an adder circuit. The location of the fault is shown in Figure 7.17.

The circuit was simulated at transistor level. The derived model from Section 7.4 was used in VHDL-AMS level simulation. Both results were then compared. The results from VHDL-AMS correspond well with results obtained from SPICE simulation for larger value of resistance. This can be observed in Figure 7.18.

There are significant differences at smaller value of resistance. As we have limited our model to handle delay faults only, the resistance that might cause stuck-at-fault behaviour have resulted larger discrepancy between actual simulation and behavioural simulations. As the value of resistance increases, the importance of delay fault testing increases and we can see very close results in this area of interest.

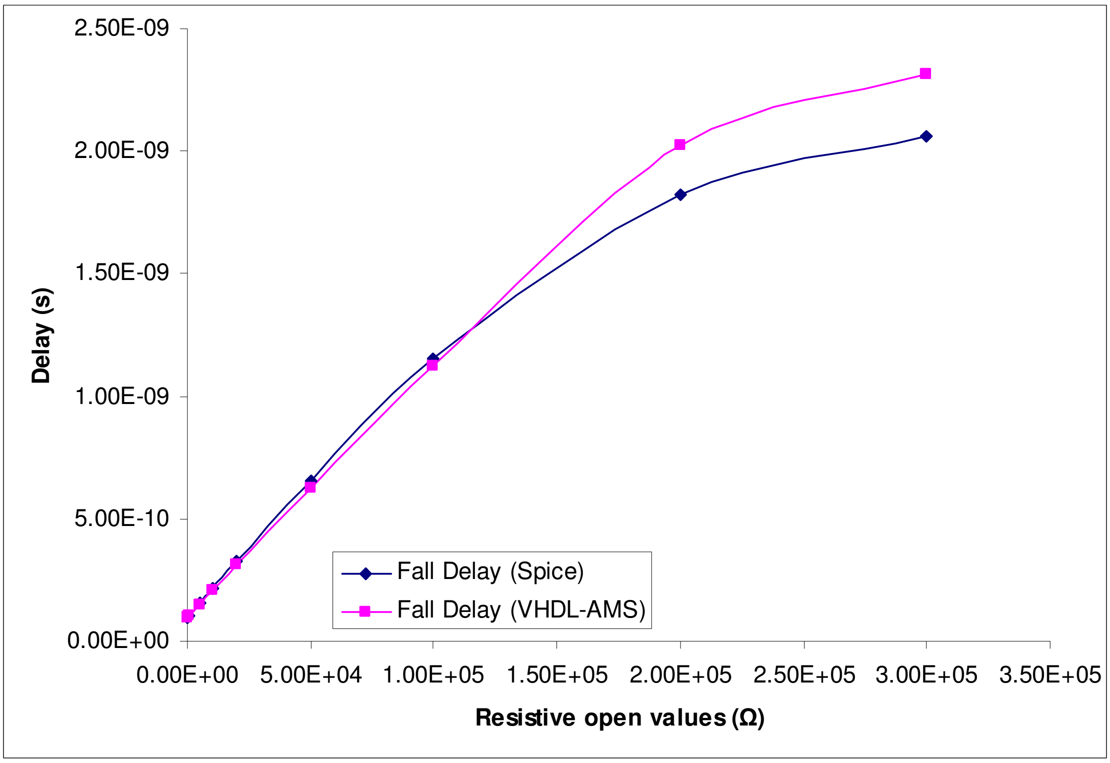


FIGURE 7.16: Path delay from SPICE and VHDL-AMS for NOR gate with 4 fan out

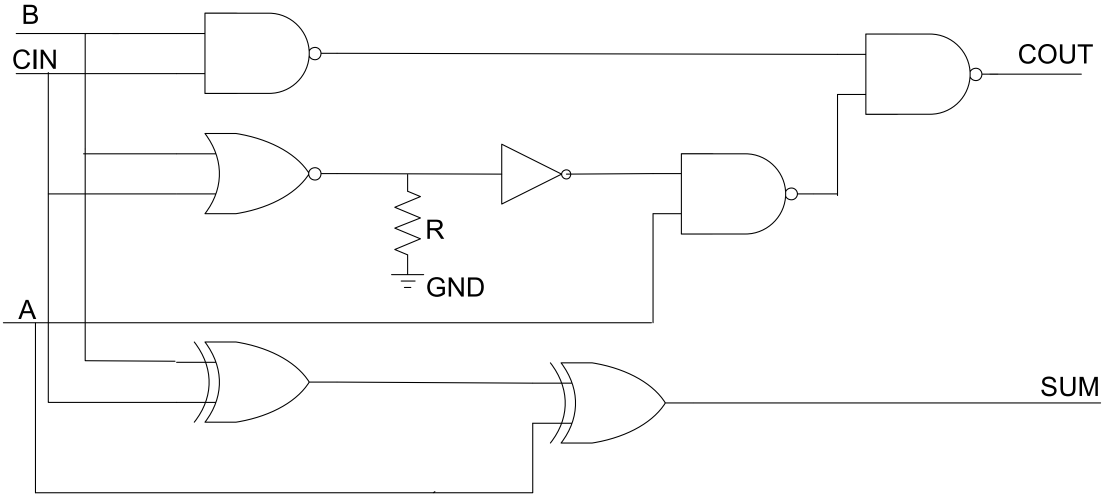


FIGURE 7.17: Adder circuit with resistive short

7.5.5 Verifying Resistive open for NAND gate with one fan-out for larger circuit

Even though the model derived has been verified on small circuits, how good is the model when it is used on a larger circuit? To answer this question, the derived

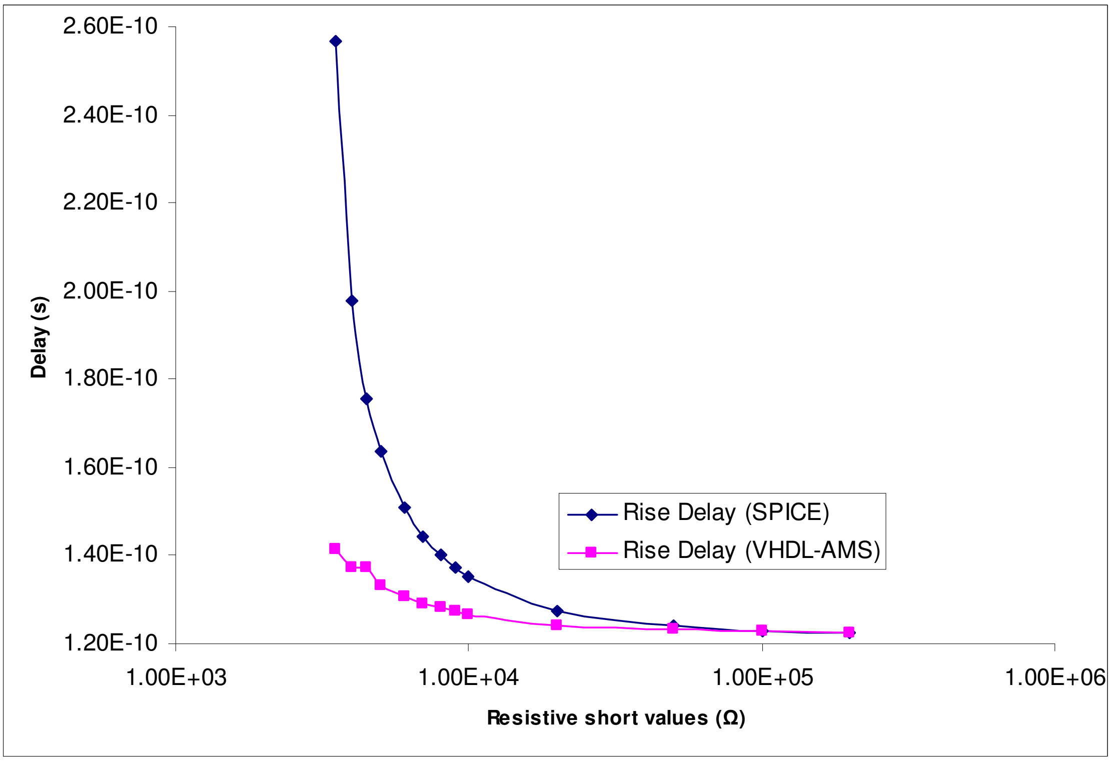


FIGURE 7.18: Path delay from SPICE and VHDL-AMS for adder with resistive short

model from the NAND gate was used on a 8 bit full adder circuit. The defective NAND gate was inserted on the first bit of the adder. The 8 bit adder was first simulated at transistor level using SPICE simulations. Then, the 8 bit adder was simulated using VHDL-AMS. As for the previous simulations, the fall case at highest voltage, 1.2 volts was used. The first four columns in Table 7.12 show the simulation results. The second column labelled **Delay (SPICE)** is the transistor level SPICE simulation results. The third column labelled **Delay (VHDL-AMS with accurate gate delay)** is the delay measured from VHDL-AMS simulations. The error percentage between the second column and third column is labelled as $Error^a$ in fourth column.

One of the important elements that contributes to the circuit delay is individual gate delay. The gate delay along the sensitize path will add up to the total circuit delays. If these individual gate delays are not included in VHDL-AMS simulation, the total calculated circuit delay will be different than actual delay measured from

TABLE 7.12: Error difference for 8 bit adder with faulty NAND gate

RES (Ω)	**Delay (SPICE)**	**Delay (VHDL-AMS with accurate gate delay)**	$Error^a$ (%)	**Delay (VHDL-AMS after correcting) factor)**	$Error^b$ (%)
0.00	9.33E-10	9.32E-10	0.14	9.33E-10	0.02
100.00	9.33E-10	9.34E-10	0.06	9.26E-10	0.75
1.00k	9.36E-10	9.41E-10	0.57	9.33E-10	0.25
5.00k	9.45E-10	9.46E-10	0.08	9.44E-10	0.14
10.00k	9.56E-10	9.61E-10	0.47	9.60E-10	0.41
20.00k	9.79E-10	9.80E-10	0.06	9.86E-10	0.65
50.00k	1.05E-09	1.06E-09	1.19	1.06E-09	1.25
75.00k	1.11E-09	1.14E-09	2.27	1.14E-09	2.36
100.00k	1.17E-09	1.20E-09	2.78	1.20E-09	2.57
200.00k	1.40E-09	1.47E-09	5.30	1.47E-09	5.12
300.00k	1.63E-09	1.74E-09	6.84	1.74E-09	7.03
400.00k	1.85E-09	2.00E-09	8.37	2.01E-09	8.56
500.00k	2.06E-09	2.26E-09	9.82	2.26E-09	9.82
750.00k	2.51E-09	2.87E-09	14.03	2.85E-09	13.55
1.00M	2.87E-09	3.37E-09	17.20	3.37E-09	17.24
2.00M	3.81E-09	4.52E-09	18.65	4.53E-09	18.87

circuit level SPICE simulation. This discrepancy will further add to the error ratio between VHDL-AMS and circuit level simulation.

In our previous simulations, these fault free gate delays have been individually measured and added to the gate descriptions at VHDL-AMS level. In practice, nominal gate delays can be obtained from data sheets of the technology file for standard cells. If standard cells are modified, the nominal delay might also have changed, therefore delays have to be measured for each modified cell.

Another way to find the total gate delays along the sensitise path is by obtaining the total path delay through circuit level SPICE simulation. The total path delay measured from fault free circuit will include the individual gate delay as well as the interconnect delays. The measurement from circuit level only needs to be done once for every different sensitise path. Once the lumped path delay is obtained and added to the circuit, the individual gate delay can be set to zero. We have named the lumped circuit delay as Correcting Factor.

The lumped delay, δ_1 which will be mainly due to gate delay and interconnect delays will be obtained from the fault free circuit. The similar fault free circuit has to be simulated on VHDL-AMS and the delay δ_2 is calculated. In obtaining δ_2, all the gate delay and interconnect delays are ignored and assumed to be zero.

The difference between δ_1 and δ_2 will be recorded as Correction Factor, δ_3. δ_3 are then added to the simulations in VHDL-AMS. By integrating the actual total gate delays in VHDL-AMS, more accurate results can be expected.

For the 8 bit adder case, we have found that the delay from SPICE for fault-free circuit was 9.33e-10s which is δ_1. Delay from VHDL-AMS for the same fault free circuit was 3.75e-10, which is recorded as δ_2. So the correction factor δ_3 is 5.58e-10. δ_3 needs to be added to all measurements taken from VHDL-AMS in order to obtain the actual delay values. The last two columns in Table 7.12 show the results after adding the correction factor. The fifth column labelled **Delay (VHDL-AMS after correcting factor)** is the delay measured from VHDL-AMS simulation after adding the correcting factor of δ_3. The error percentage between actual SPICE simulation (column 2) and delay from fifth column is shown as $Error^b$ in the last column of the table.

Figure 7.19 shows the difference of the delay from three different measurements. The actual delay measured (Fall Delay Spice) is compared with the two other delay measured using different methods. It can be clearly observed that the delay measured with the correction factor closely match the delay measured with accurate gate delay.

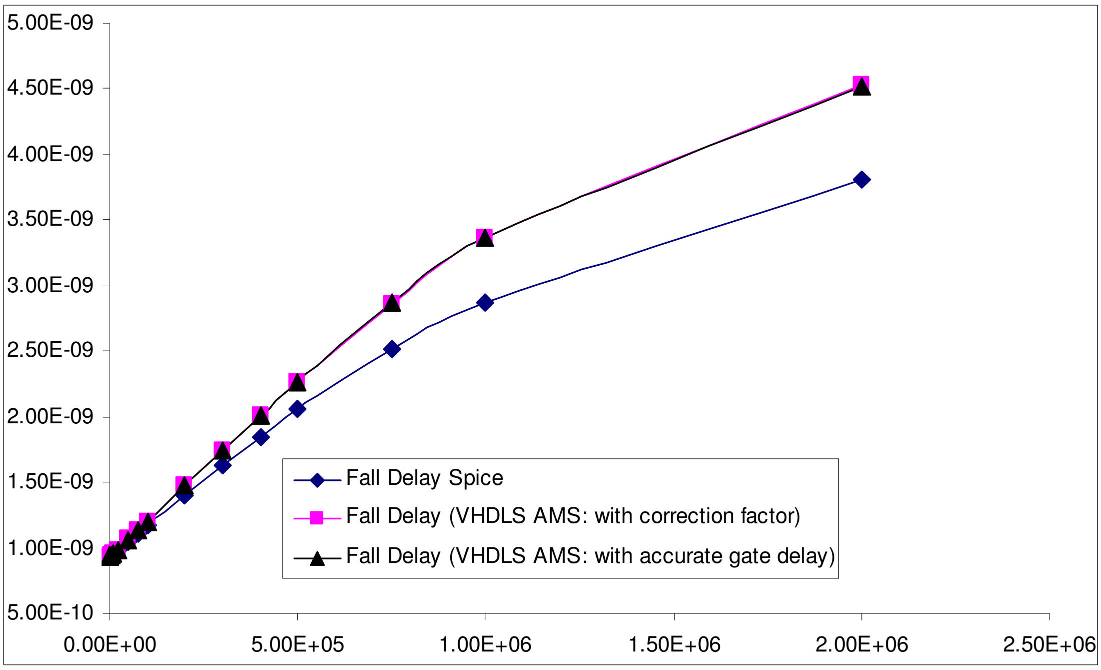


Figure 7.19: Delay measurement from three different methods

7.6 Analysis

Simulation results for resistive opens and resistive shorts show that the derived model for basic cells gives acceptable accuracy in comparison to the results obtained from circuit level SPICE simulations. For resistive open, results from higher supply voltage were used to derive the model since higher supply voltage gives the largest fault coverage. On the other hand, results from lowest supply voltage were used in the case of resistive shorts.

For resistive opens, as the value of the resistance gets larger, the accuracy of the results gets worse. This is because a larger resistance will push the circuit to stuck-at-fault behaviour. These large values of resistance will be detected by means of static stuck-at-fault testing. For resistor shorts, large values of resistance, which will generally cause performance degradation, have better accuracy in our model. The smaller values of resistance which are‘ closer to the non resistive short have significant difference between the model and actual circuit level simulations. However, these faults will be generally detected at static voltage testing.

For improved accuracy, the delay model can be further optimised by adding other features. Among the possible optimisations is the accurate modelling of the interconnects as well as segmented modelling of the delay curve. However, it should be noted that by adding these features, there will be penalty in term of processing speed. It will be a decision to be made by the test engineer on the level of accuracy needed for the behavioural level simulations to get the best trade-off between the speed and accuracy. The procedure flow of the process can be simplified as in Figure 7.20.

7.7 Summary

The method proposed enables the time used for fault simulation to be reduced significantly. The delay models were derived from basic cells. SPICE simulations at transistor level are used for the derivation of the model. The derived models are then used in more complex circuits and the simulations are conducted using behavioural fault simulations with VHDL-AMS. The simplified VHDL-AMS model used has resulted in acceptable accuracy for most cases. For larger circuits one has the option either to obtain the individual gate delays for all the cells being used or the circuits can be simulated once at transistor level to obtain the correction factor. Further optimisation for delay model is possible by increasing the complexity of the simplified delay model. The optimisation parameters have to be carefully added to get the best trade-off between speed and accuracy.

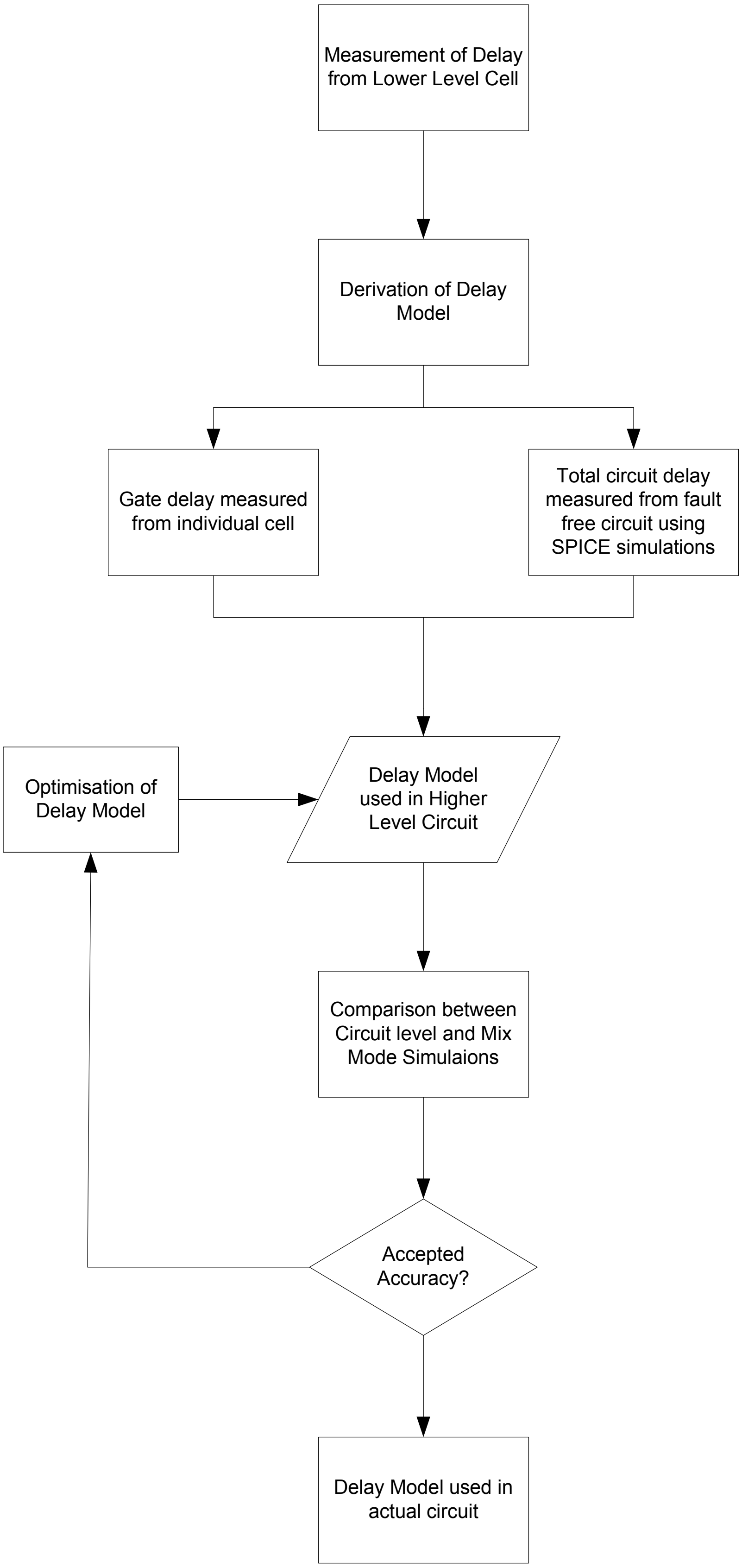

Measurement of Delay from Lower Level Cell
Derivation of Delay Model
Gate delay measured from individual cell
Total circuit delay measured from fault free circuit using SPICE simulations
Optimisation of Delay Model
Delay Model used in Higher Level Circuit
Comparison between Circuit level and Mix Mode Simulaions
Accepted Accuracy?
Delay Model used in actual circuit

Figure 7.20: Design flow of fault modelling and simulations

Chapter 8

Conclusions and future work

It is very likely that the use of Multi Voltage Design(MVD) as a method to achieve efficient power management technique particularly in consumer electronics, will continue to increase. Developing efficient test methodology for MVD design is very essential in order to deliver highly reliable products to consumers. At the same time, industry cannot afford to conduct testing in naive approach as it will be very uneconomical . The aim of the present work is to suggest efficient delay fault testing techniques for MVD systems.

This last chapter comprises two sections. Section 8.1 gives the conclusion of this work and Section 8.2 proposes ideas for future work.

8.1 Conclusion

Todate the testing methodology for circuit working at more than one voltage has not been fully presented. We have looked on some of the important issues in relation to testing the multi voltage design circuits.

The work can be separated in 3 main sections

1. **Voltage requirement for delay fault testing**

Previous work on low voltage testing gave a general idea on behaviour of the defects at different operating condition. We have studied the behaviour of two main classes of defects : resistive open and resistive bridging fault. The resistance as well as the supply voltage were varied and responses were observed.

Resistive open defects have better detection ratio at higher supply voltage. On the other hand, lower supply voltage gives better detection ratio for resistive bridging faults. Results show that for acceptable fault coverage, it is necessary to select the number of voltage-specific delay fault test. This will be in addition to the voltage dependant stuck-at-fault test. The initial finding of this work titled ***"Dynamic Voltage Scaling Aware Delay Fault Testing"*** has been published in IEEE European Test Symposium, conducted at Southampton in May 2006.

2. **Testing of Level Shifter for Multi Voltage Design**

Level shifters play an important role in Multi Voltage Designs. A defective level shifter can cause performance degradation as well as logical error. Other faults such as bridging faults will be amplified in the presence of a defective level shifter.

Through extensive simulation we have shown that the level shifter can be tested using a single supply voltage. This reduces the need for extra test setup as well as the requirement for additional test vectors. We have used two different defects i.e resistive opens and restive shorts. The delay ratio for both classes of defects are in line with results in part 1 of this work. This work with the title ***"Testing of Level Shifters in Multiple Voltage Designs"*** has been published in the 14th IEEE International Conference on Electronics, Circuits and Systems at Morocco in December 2007.

3. **Delay Fault Modelling/Simulation using VHDL-AMS in Multi Voltage Design**

Fault simulation is conventionally done at gate or transistor level. Fault simulation using mix-mode language such as VHDL-AMS will reduce the fault simulation time. We have proposed a method to run the fault simulation by deriving the fault delay model at transistor level and using the model in gate level. The delay model was derived from basic cells. SPICE simulation at transistor level is used for the derivation. The derived model is then used in more complex circuits and the simulations are conducted using behavioural fault simulations using VHDL-AMS. The simplified VHDL-AMS model used has resulted in acceptable accuracy for most cases. For larger circuits one has the options either to obtain the individual gate delays for all the cells being used or the circuits can be simulated once at transistor level to obtain the correction factor. Part of findings from this work has been published in the 26th IEEE International Conference on Microelectronics conducted at Nis, Serbia in May 2008. The paper title is ***"Delay Fault Modelling/Simulation using VHDL-AMS in Multi-Vdd Systems"***.

Overall, from the point of view of testability, we have suggested key points that will greatly assist in testing procedures of a Multi Voltage Design. It is expected that the findings of this work will expedite the testing process without sacrificing the test quality.

8.2 Future Work

Future work suggested in this chapter is proposed to improve the testability of Multi Voltage Design circuits. The three areas that can be explored are test point insertion for improving delay fault testing of Multi Voltage Design, handling specific issues in relation to different types of Multi Voltage Design techniques and integration of static fault behaviour to our behavioural fault simulations.

The aim of the proposed future work is to reduce the number of supply voltage settings required for testing. The idea is based on work done by Ingelsson et al. [57]. In their work, test point insertions were used to reduce the number of supply voltages required for bridging fault detection. The idea can be extended for delay fault testing for Multi Voltage Design. The insertion of test point will result in area overhead to implement the test point structure. The challenge will be to find the best compromise between increase in silicon area and reduction in number of supply voltages to achieve required fault coverage.

In this thesis, we have looked at problem generic to any multi voltage design. There are specific issues to particular types of MVD. For an example, in Dynamic Voltage and Frequency Scaling (DVFS), in addition to voltage, the frequency was also adjusted. It is common that these voltage/frequency pairs are fixed [125]. Further research is required to see the defect behaviour within this fixed voltage/frequency pair. Even though our results indicate these fixed pairs will not have a large impact, these have to be verified by extensive simulations.

In the area of behavioural simulations using VHDL-AMS, we have observed some limitation in which the model did not give accurate results. This inaccuracy is observed at the area close to the critical resistance. This indicates that if our model is integrated to handle static fault, the accuracy of the model will be much improved. In addition, the delay model can be further improved by segmenting the delay curve into multiple segments.

Appendix A

Multi Voltage Testing for Lower Geometry Circuit

This appendix briefly presents results and discussion for multi voltage testing conducted at 0.35 micron technology. As the actual results database were destroyed in Mountbatten fire, only partial results from printed copy are used in this discussions.

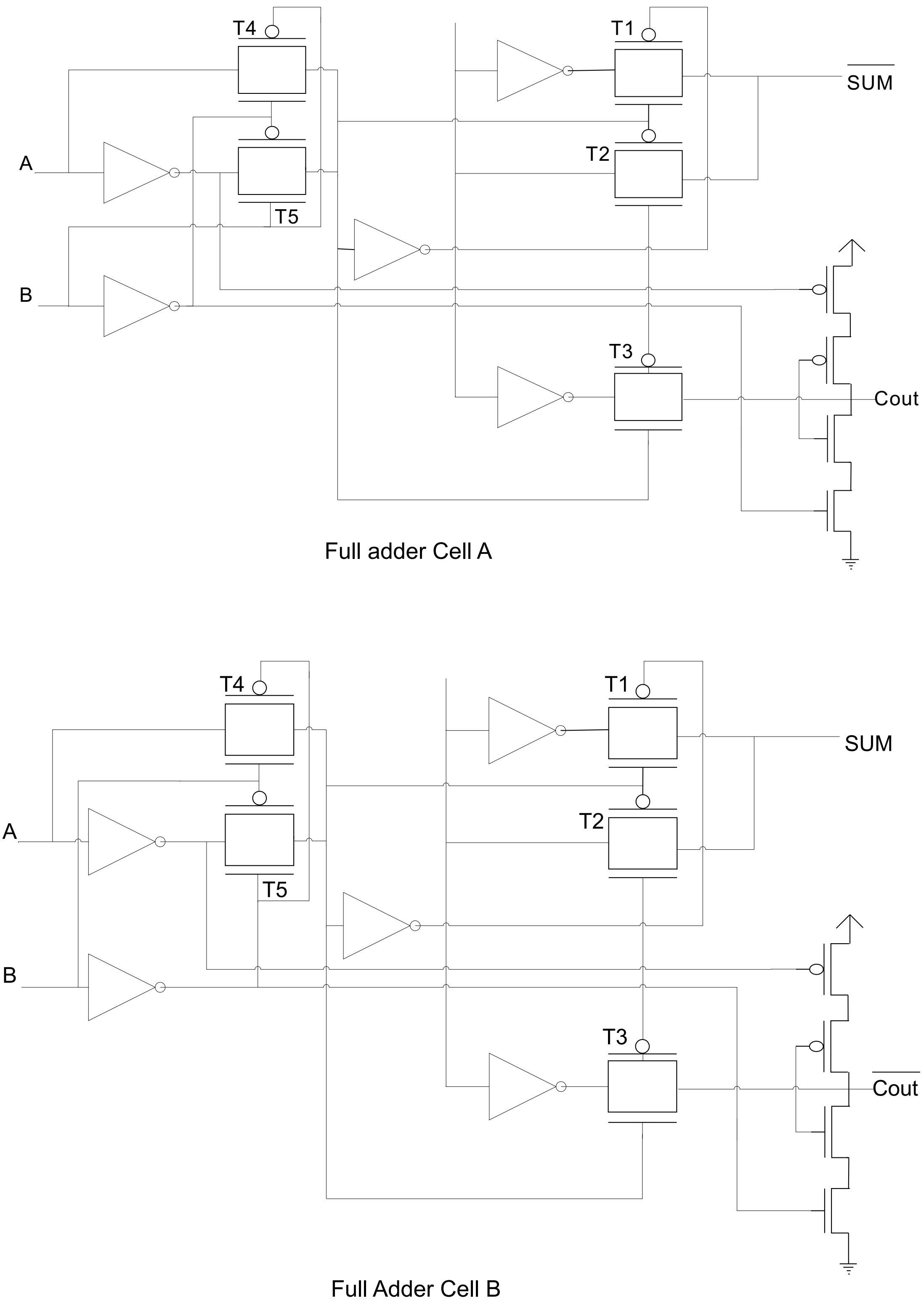
A.1 Introduction

We have studied two defects that can cause timing failures. These are transmission gate opens and resistive opens. We have chosen these defects since both will cause increases in propagation delays. It is important to note that certain faults such as NMOS gate-to-source shorts and NMOS gate-to-drain shorts will cause reductions in delay values. Transmission gate opens were thoroughly studied in [40] using 0.8 and 0.6 μm technology. In order to determine whether the fault effects become more significant with changing feature sizes, we have revisited the examples using 0.35 μm technology and voltage steps from an actual DVS processor. The voltage steps are 3.3, 3.0, 2.7, 2.5 and 2.0 Volts. These voltage ranges are consistent with the voltage range of StrongArm SA1100 DVS processors [147].

A.2 Transmission Gate Opens

Transmission gate opens were simulated using a similar setup to that in [40]. The circuit is part of a multiplier consisting of 4 levels of carry-save adders. The circuit uses two different pass transistor logic implementations for full adder cells which is shown in figure A.1. Interconnections between the adder's cells are shown in figure A.2. CSA11, CSA12, CSA13, CSA31, CSA32 and CSA33 use adder cell A. CSA01, CSA02, CSA03, CSA21, CSA22 and CSA23 use adder cell B. Each adder has five transmission gates. If one of the transistors, either PMOS or NMOS, is stuck open, the output will be degraded. A degraded signal is defined as one in which V_{IH} is lower then the supply voltage or V_{IL} is higher than the ground signal.

Faults were injected at two different locations. Both of the faults are NMOS opens. For the 0.35 μm technology, $|Vtp|$ is higher than V_{tn}. This will result in NMOS transistors having a higher driving strength and an open in a NMOS gate will create a longer delay than an open in a PMOS gate. The first fault is an NMOS open at the output transmission gate of CSA11. This open will cause a degraded signal to be passed to input of CSA21. The second fault is an open at the output transmission gate of CSA22. The resulting degraded signal will be fed to input B of CSA32. The signal paths for the faults are shown in table A.7 The faults were injected in two different types of cell to show how the supply voltages affect different fault locations. Inverters were used as buffers at all inputs and outputs of the circuit under test.



Full adder Cell A

Full Adder Cell B

Figure A.1: Adder Cell A and B

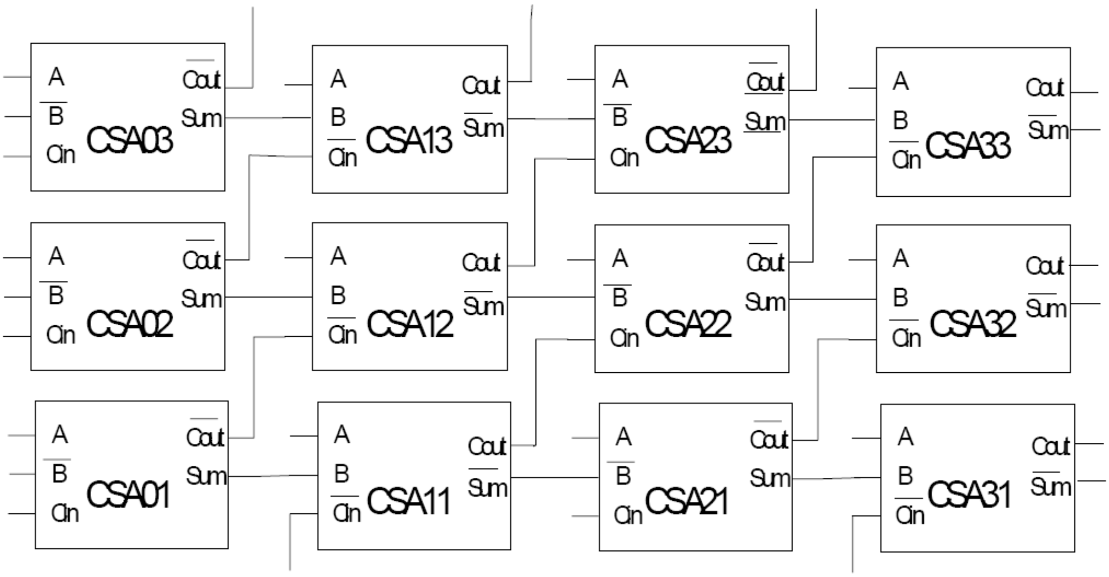


FIGURE A.2: Interconnection between adder cells for simulation setup

Fault	Fault Site	Signal Processing Path
A	CSA11 NMOS Open	CSA01(A)-CSA11(B)-CSA21 (B)-CSA32(Cin) CSA32(Cout)
B	CSA22 NMOS Open	CSA01(A)-CSA11(B)-CSA22 (Cin)-CSA32(B)-CSA32(Cout)

TABLE A.1: Signal propagating path

A.2.1 Simulation Results

Figure A.3 shows how the path delay changes for the fault-free and transmission gate fault cases, with varying supply voltage. In all three examples, the fault-free case is shown as a solid trace, while the faulty delay is shown dashed. It can be seen that the delay increases as the supply voltage falls, until at 2.5V, the fault behaves as a stuck fault, after the initial transient.

Table A.2 shows the path delay ratio and gate delay ratios between the faulty circuits and the fault free circuit. The path delay ratio is measured as the ratio of the signal propagating path of the faulty circuit to the same path of the fault free circuit. The gate delay ratio is the ratio between the delay of the faulty adder cell and the fault free adder cell. The entries in Table 2 shown as "SAF" mean the fault causes the circuit to show a stuck-at-fault error. In [40] Chang reported that the stuck fault happens at lower voltages for larger geometry process. Our results

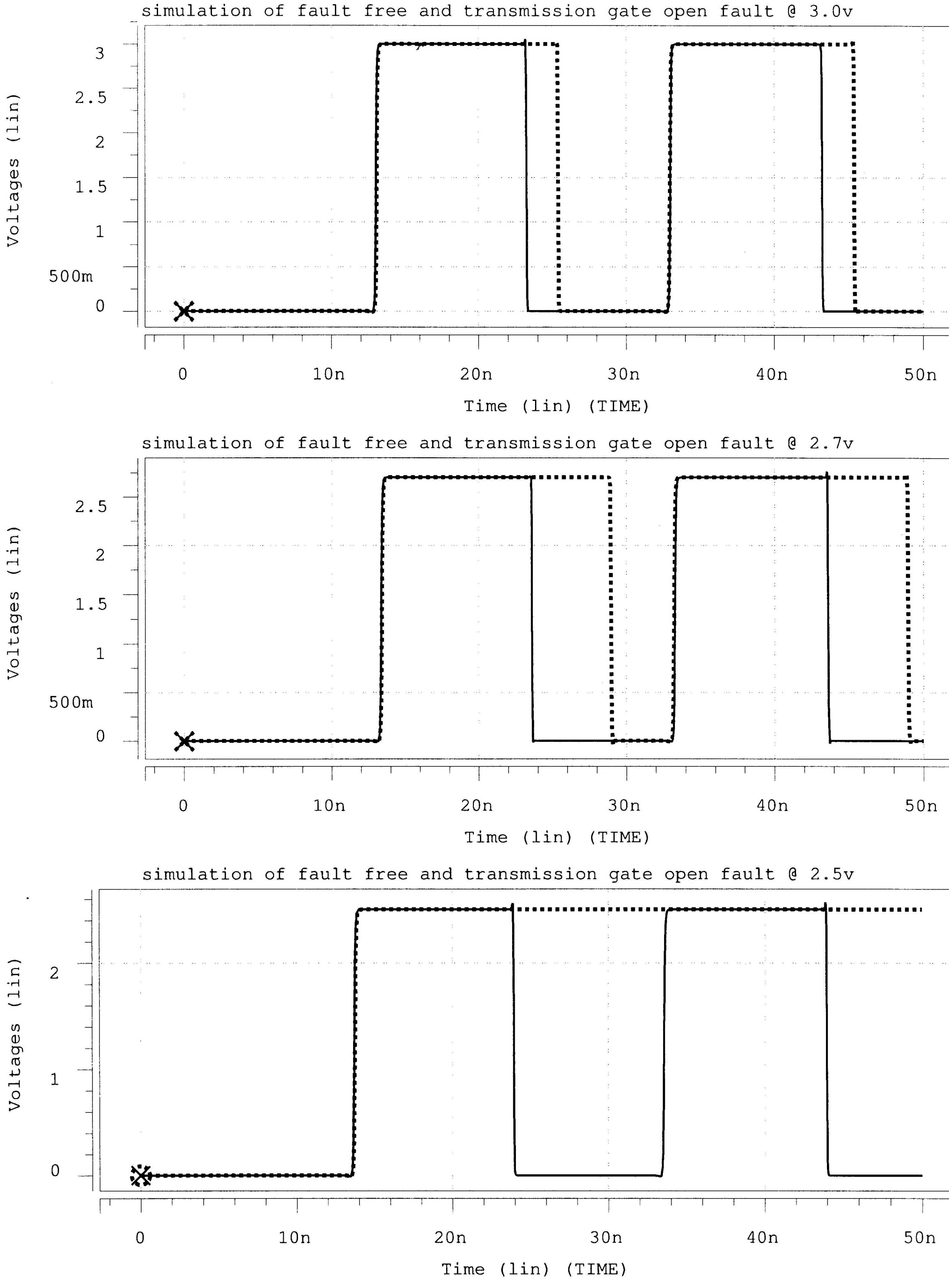


Figure A.3: Change in delay for fault-free and transmission gate open fault at different supply voltages

shows similar patterns with the delay faults manifesting themselves as stuck faults at higher voltages. As expected, both the gate delay ratio and the path delay ratio increase as we reduce the voltage. Another important point to note is that for the path delay ratio, the highest voltage shows a minimum ratio of 36%. With the increasing accuracy of the Automated Test Equipments (ATE), these delay faults can be captured even at much higher voltages.

Vdd	Fault A		Fault B	
	Path Delay Ratio	Gate Delay Ratio	Path Delay Ratio	Gate Delay Ratio
3.3	1.36	1.02	1.43	1.05
3.0	1.56	1.05	1.68	1.10
2.7	2.27	1.12	2.51	1.19
2.5	SAF	SAF	SAF	SAF
2.0	SAF	SAF	SAF	SAF

TABLE A.2: Path delay ratio and gate delay ratio for fault A and fault B

A.3 Resistive Open Defects

A resistive open defect can be modelled as a resistance between two nodes. This is depicted in Figure A.4. Previous research, [15], categorises open faults into strong-opens (>10MΩ) and weak opens (≤10MΩ). Strong-opens will cause stuck-at faults and can be detected using standard stuck-at patterns. Weak opens are difficult to detect because they have timing-dependent test results. This implies that the test results change with test speed. We have used 2 different circuits to evaluate the effect of the power supply on gate and path delay ratios for circuits with weak opens.

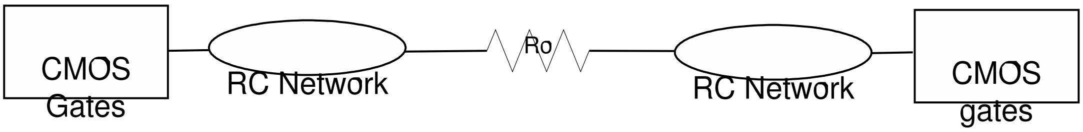


FIGURE A.4: Resistive open fault model

A buffer chain of 6 inverters, as shown in Figure A.5, was simulated to characterise the behaviour of a circuit with weak resistive opens. The resistor open defects were injected between the second and third inverters. The path delay was measured between the input of the second inverter and output of the fifth inverter and the gate delay was measured between the same input and the output of third inverter. Table A.3 and A.4 list the delay ratios for the circuit.

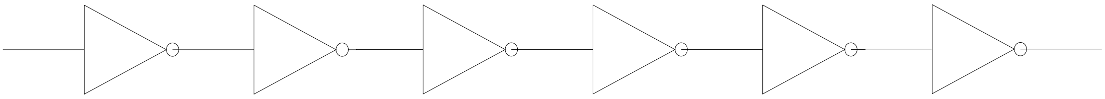


Figure A.5: Buffer chain of 6 inverters

Vdd	R_0=500 Ω	R_0=10k Ω	R_0=25k Ω	R_0=250k Ω	R_0=1M Ω
3.30	1.01	1.09	1.20	3.20	9.25
3.00	1.00	1.07	1.18	3.04	8.50
2.70	1.00	1.06	1.16	2.75	7.65
2.50	1.00	1.06	1.16	2.70	7.22
2.00	1.00	1.05	1.13	2.25	5.59

Table A.3: Path delay ratio for buffer chain

Vdd	R_0=500 Ω	R_0=10k Ω	R_0=25k Ω	R_0=250k Ω	R_0=1M Ω
3.30	1.01	1.15	1.40	5.38	17.46
3.00	1.01	1.15	1.36	5.01	16.01
2.70	1.01	1.11	1.32	4.48	14.20
2.50	1.01	1.11	1.32	4.36	13.36
2.00	1.00	1.09	1.24	3.38	9.91

Table A.4: Gate delay ratio for buffer chain

To observe the impact of resistive opens in a more complex circuit, weak resistive open faults were injected into the circuit of Figure A.2. The resistive open defects were injected at 2 locations in the multiplier cell. The first location (A) is between CSA22 and CSA 32. The second location (B) is between CSA11 and CSA21. Table A.5 and A.6 list the delay ratios.

A.3.1 Simulation Results

Figure A.6 and figure A.7 shows the circuit waveforms for the inverter chain under fault-free conditions (solid) and with the resistive open fault inserted (dashed),

Vdd	R_0=25k Ω		R_0=250kΩ		R_0=1MΩ	
	Fault Location					
	A	B	A	B	A	B
3.30	1.02	1.02	1.18	1.13	1.67	1.48
3.00	1.01	1.02	1.17	1.12	1.61	1.44
2.70	1.01	1.02	1.15	1.11	1.56	1.39
2.50	1.01	1.01	1.12	1.10	1.50	1.36
2.00	1.01	1.01	1.11	1.07	1.38	1.26

Table A.5: Path delay ratio for multiplier circuit for resistive open faults

Vdd	R_0=25k Ω		R_0=250kΩ		R_0=1MΩ	
	Fault Location					
	A	B	A	B	A	B
3.30	1.00	1.01	1.32	1.16	2.07	1.55
3.00	1.00	1.02	1.30	1.15	1.97	1.54
2.70	1.00	1.01	1.29	1.15	1.95	1.51
2.50	1.00	1.01	1.28	1.15	1.86	1.50
2.00	1.00	1.01	1.25	1.12	1.74	1.41

Table A.6: Gate delay ratio for multiplier circuit for resistive open faults

at 3.3 V and 2.0 V respectively. In both cases, the waveform is observed one inverter after the fault site. It can be seen that as the supply voltage falls, the absolute delay increases for both the fault-free and faulty cases. However, the relative delay for the faulty cases decreases compared with the fault-free delay. From table A.5 and A.6, it can be seen that the reduction of the delay ratios with decreasing voltage for both path and gate delays is significant. The delay ratio pattern is inverse to that for transmission gate open faults. As we reduce the supply voltage, the time taken for gate signals to settle grows. This is true for both faulty and fault free circuits. Note however, that the delay in the faulty circuit does not change proportionally to that in the fault free circuit. Table 7 shows the absolute delay of the circuit for the fault free case and for a resistive open fault of 1MΩ. The delay almost doubles (1.83) when the supply voltage is reduced from 3.3V to 2.0V for fault free circuit. For the faulty circuit the increase in delay is only 1.04 times. For small resistive open fault values, the increase in delay due to the fault is not significant at both ends of the voltage range. However, as the resistor value increases, the impact of the voltage reduction is marked. For the more complex circuit, the impact of the resistance is not as significant as in the smaller circuit. This is because the delayed signals get restored at the next adder cell. The statistical distribution of resistive open faults of 1MΩ to 10MΩ is similar to that in the range of 100KΩ to 1MΩ[102]. The probability of having a resistive open of 1MΩ and above is as high as having a lower value of resistance. In this scenario it is better to run the test at the highest voltage since not only will a fault give a larger delay ratio, the test application time will be significantly reduced.

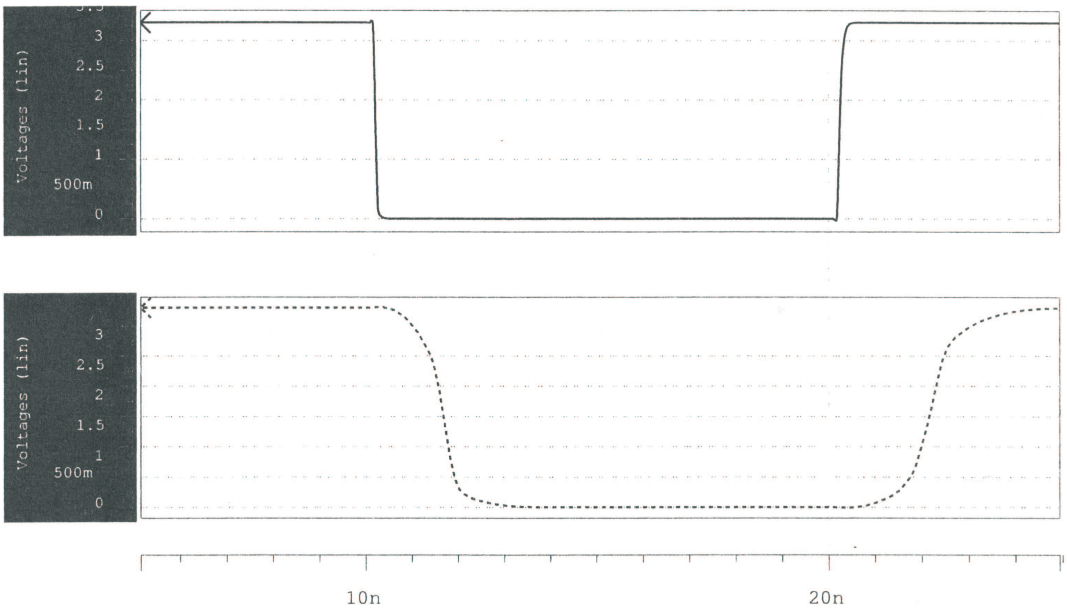


FIGURE A.6: Fault-free (solid) and resistive open delays(dashed) at 3.3V

Vdd	Gate Delay for fault free circuit	Gate Delay for resistive open of 1MΩ
3.3	1.19E-10	2.07E-09
2.0	2.18E-10	2.16E-09

TABLE A.7: Signal propagating path

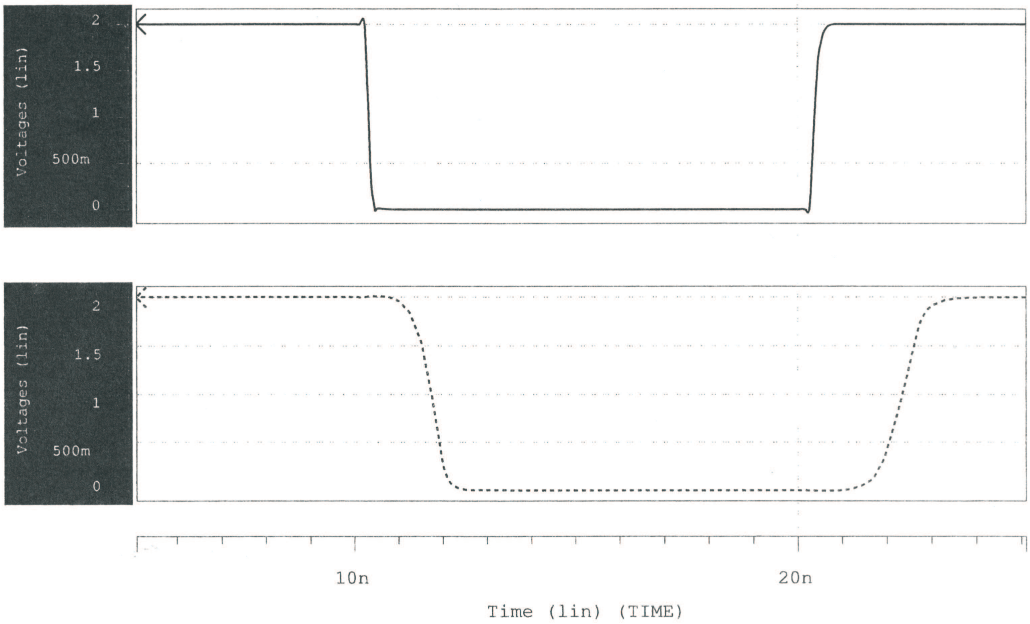


FIGURE A.7: Fault-free (solid) and resistive open delays(dashed) at 2.0V

A.4 Summary

This section presents an approach to a testing strategy for delay faults in Dynamic Voltage Scaling systems. Our study shows that we do not need to use the lowest operating voltages to detect certain types of faults. From the transmission gate open simulation it is evident that low to mid-range voltages give sufficient fault coverage. In general testing at lower operating voltages is only required for certain types of faults such as transmission gate opens and bridging faults. On the other hand, weak resistive opens that cause delay faults are best tested at higher operating voltages. Simulation results both of a simple inverter chain circuit and a more complicated multiplier circuits support our conclusions. The overall conclusion is that in order to guarantee the test quality of DVS systems, it will be necessary to select a number of voltage-specific delay fault tests, in addition to voltage-independent stuck-fault tests. Initial testing can be done at the highest operating voltage and this will reduce the time and cost of the test. The escaped defects can be detected at lower mid-range voltages without the need to go to the lowest voltages. Future work will aim to find an optimal set of voltage/fault test pairs.

Appendix B

Spice Netlist for Top Level Multiplier Array Circuit for Figure 5.4

```
// Generated for: spectre
// Generated on: Apr 28 23:13:09 2008
// Design library name: year4
// Design cell name: multiplier_defectB
// Design view name: schematic
simulator lang=spectre
global 0
include "models.scs"
parameters prd_x0=8e-09 prd_x1=8e-09 prd_y0=8e-09 prd_y1=8e-09 \
    prd_y2=8e-09 prd_y3=8e-09 pw_x0=4e-09 pw_x1=4e-09 pw_y1=4e-09 \
    pw_y2=4e-09 pw_y3=4e-09 res=5000000.0 vsupply=0.8 prd_x2=8e-09 \
    pw_x2=4e-09 pw_y0=4e-09 set0=0 set1=0 set10=0 set11=0 set12=0 set13=0 \
    set14=0 set15=0 set2=0 set3=0 set4=0 set5=0.8 set6=0 set7=0 set8=0 \
    set9=0

// Library name: HCMOS9GP_Dig
// Cell name: MC_ND2HS
// View name: schematic
subckt MC_ND2HS A B dvdd dvss out
    M3 (out B dvdd dvdd) EPHSGP_BS3JU w=0.77u l=0.13u nfing=1 ncrsd=1 \
        number=1 srcefirst=1 ngcon=1 mismatch=1 po2act=-1 nbti=0 lpe=0
    M2 (out A dvdd dvdd) EPHSGP_BS3JU w=0.77u l=0.13u nfing=1 ncrsd=1 \
        number=1 srcefirst=1 ngcon=1 mismatch=1 po2act=-1 nbti=0 lpe=0
    M0 (out A net9 dvss) ENHSGP_BS3JU w=0.64u l=0.13u nfing=1 ncrsd=1 \
        number=1 srcefirst=1 ngcon=1 mismatch=1 po2act=-1 lpe=0
    M1 (net9 B dvss dvss) ENHSGP_BS3JU w=0.64u l=0.13u nfing=1 ncrsd=1 \
        number=1 srcefirst=1 ngcon=1 mismatch=1 po2act=-1 lpe=0
ends MC_ND2HS
// End of subcircuit definition.

// Library name: HCMOS9GP_Dig
// Cell name: MC_IVHS
// View name: schematic
subckt MC_IVHS dvdd dvss in out
    M1 (out in dvdd dvdd) EPHSGP_BS3JU w=2.1u l=0.13u nfing=1 ncrsd=1 \
        number=1 srcefirst=1 ngcon=1 mismatch=1 po2act=-1 nbti=0 lpe=0
    M0 (out in dvss dvss) ENHSGP_BS3JU w=1.17u l=0.13u nfing=1 ncrsd=1 \
        number=1 srcefirst=1 ngcon=1 mismatch=1 po2act=-1 lpe=0
ends MC_IVHS
// End of subcircuit definition.

// Library name: year3
// Cell name: NOR_NB
// View name: schematic
subckt NOR_NB A B dvdd dvss out
    M5 (dvss B out dvss) ENHSGP_BS3JU w=1.28u l=0.13u nfing=1 ncrsd=1 \
        number=1 srcefirst=1 ngcon=1 mismatch=1 po2act=-1
    M4 (out A dvss dvss) ENHSGP_BS3JU w=1.28u l=0.13u nfing=1 ncrsd=1 \
        number=1 srcefirst=1 ngcon=1 mismatch=1 po2act=-1
    M7 (net54 A dvdd dvdd) EPHSGP_BS3JU w=2.28u l=0.13u nfing=1 ncrsd=1 \
        number=1 srcefirst=1 ngcon=1 mismatch=1 po2act=-1 nbti=0
    M8 (net54 B out dvdd) EPHSGP_BS3JU w=2.28u l=0.13u nfing=1 ncrsd=1 \
        number=1 srcefirst=1 ngcon=1 mismatch=1 po2act=-1 nbti=0
ends NOR_NB
// End of subcircuit definition.

// Library name: year3
// Cell name: XOR_NB
// View name: schematic
subckt XOR_NB A B X vdd vss
    I5 (vdd vss B B1) MC_IVHS
    I2 (vdd vss A A1) MC_IVHS
    M6 (0 B1 net5 0) ENHSGP_BS3JU w=1.28u l=0.13u nfing=1 ncrsd=1 number=1 \
        srcefirst=1 ngcon=1 mismatch=1 po2act=-1
    M5 (net5 A1 X 0) ENHSGP_BS3JU w=1.28u l=0.13u nfing=1 ncrsd=1 number=1 \
        srcefirst=1 ngcon=1 mismatch=1 po2act=-1
    M1 (net16 B 0 0) ENHSGP_BS3JU w=1.28u l=0.13u nfing=1 ncrsd=1 number=1 \
        srcefirst=1 ngcon=1 mismatch=1 po2act=-1
    M4 (X A net16 0) ENHSGP_BS3JU w=1.28u l=0.13u nfing=1 ncrsd=1 number=1 \
        srcefirst=1 ngcon=1 mismatch=1 po2act=-1
    M3 (net23 A1 X vdd) EPHSGP_BS3JU w=2.28u l=0.13u nfing=1 ncrsd=1 \
        number=1 srcefirst=1 ngcon=1 mismatch=1 po2act=-1 nbti=0
    M2 (vdd B net23 vdd) EPHSGP_BS3JU w=2.28u l=0.13u nfing=1 ncrsd=1 \
        number=1 srcefirst=1 ngcon=1 mismatch=1 po2act=-1 nbti=0
    M0 (X A net30 vdd) EPHSGP_BS3JU w=2.28u l=0.13u nfing=1 ncrsd=1 \
        number=1 srcefirst=1 ngcon=1 mismatch=1 po2act=-1 nbti=0
    M7 (net30 B1 vdd vdd) EPHSGP_BS3JU w=2.28u l=0.13u nfing=1 ncrsd=1 \
        number=1 srcefirst=1 ngcon=1 mismatch=1 po2act=-1 nbti=0
ends XOR_NB
// End of subcircuit definition.

// Library name: year4
// Cell name: adder_for_multi_defectB
// View name: schematic
subckt adder_for_multi_defectB A B CIN COUT SUM vdd vss
    R1 (A net033) resistor r=res
    I19 (net40 net45 vdd vss COUT) MC_ND2HS
    I18 (B CIN vdd vss net40) MC_ND2HS
    I20 (net49 A vdd vss net45) MC_ND2HS
    I5 (vdd vss net54 net49) MC_IVHS
    I16 (B CIN vdd vss net54) NOR_NB
    I15 (net62 net033 SUM vdd vss) XOR_NB
    I14 (B CIN net62 vdd vss) XOR_NB
ends adder_for_multi_defectB
// End of subcircuit definition.

// Library name: year4
// Cell name: adder_for_multi
// View name: schematic
subckt adder_for_multi A B CIN COUT SUM vdd vss
    I19 (net40 net45 vdd vss COUT) MC_ND2HS
    I18 (B CIN vdd vss net40) MC_ND2HS
    I20 (net49 A vdd vss net45) MC_ND2HS
    I5 (vdd vss net54 net49) MC_IVHS
    I16 (B CIN vdd vss net54) NOR_NB
    I15 (net62 A SUM vdd vss) XOR_NB
    I14 (B CIN net62 vdd vss) XOR_NB
```

```
ends adder_for_multi
// End of subcircuit definition.

// Library name: year4
// Cell name: multiplier_defectB
// View name: schematic
I32 (y1x1 y2x0 vss net45 net49 vdd vss) adder_for_multi_defectB
V2 (y2x1 0) vsource type=pulse val0=0.0 val1=set9 period=prd_x2 delay=0.0 \
        rise=50p fall=50p width=pw_x2
V4 (y2x2 0) vsource type=pulse val0=0.0 val1=set10 period=prd_x1 delay=0.0 \
        rise=50p fall=50p width=pw_x1
V6 (y3x2 0) vsource type=pulse val0=0.0 val1=set14 period=prd_y1 delay=0 \
        rise=50p fall=50p width=pw_y1
V8 (y2x0 0) vsource type=pulse val0=0.0 val1=set8 period=prd_y3 delay=0.0 \
        rise=50p fall=50p width=pw_y3
V24 (y0x1 0) vsource type=pulse val0=0.0 val1=set1 period=prd_x2 delay=0.0 \
        rise=50p fall=50p width=pw_x2
V19 (y0x3 0) vsource type=pulse val0=0.0 val1=set3 period=prd_x0 delay=0.0 \
        rise=50p fall=50p width=pw_x0
V15 (y0x2 0) vsource type=pulse val0=0.0 val1=set2 period=prd_x1 delay=0.0 \
        rise=50p fall=50p width=pw_x1
V9 (y3x1 0) vsource type=pulse val0=0.0 val1=set13 period=prd_y2 delay=0.0 \
        rise=50p fall=50p width=pw_y2
V3 (y2x3 0) vsource type=pulse val0=0.0 val1=set11 period=prd_x0 delay=0.0 \
        rise=50p fall=50p width=pw_x0
V23 (y1x3 0) vsource type=pulse val0=0.0 val1=set7 period=prd_y0 delay=0 \
        rise=50p fall=50p width=pw_y0
V22 (y1x2 0) vsource type=pulse val0=0.0 val1=set6 period=prd_y1 delay=0 \
        rise=50p fall=50p width=pw_y1
V20 (y1x0 0) vsource type=pulse val0=0.0 val1=set4 period=prd_y3 delay=0.0 \
        rise=50p fall=50p width=pw_y3
V17 (y0x0 0) vsource type=pulse val0=0.0 val1=set0 period=prd_y3 delay=0.0 \
        rise=50p fall=50p width=pw_y3
V7 (y3x0 0) vsource type=pulse val0=0.0 val1=set12 period=prd_y3 delay=0.0 \
        rise=50p fall=50p width=pw_y3
V21 (y1x1 0) vsource type=pulse val0=0.0 val1=set5 period=prd_y2 delay=0.0 \
        rise=50p fall=50p width=pw_y2
V5 (y3x3 0) vsource type=pulse val0=0.0 val1=set15 period=prd_y0 delay=0 \
        rise=50p fall=50p width=pw_y0
V1 (vss 0) vsource dc=0 type=dc
V0 (vdd 0) vsource dc=vsupply type=dc
I20 (y0x1 y1x0 vss net38 P1 vdd vss) adder_for_multi
I17 (net073 y3x3 net050 P7 P6 vdd vss) adder_for_multi
```

```
I16 (vss net056 net057 net054 P4 vdd vss) adder_for_multi
I15 (net054 net0135 net043 net073 P5 vdd vss) adder_for_multi
I14 (y0x3 net038 net031 net057 P3 vdd vss) adder_for_multi
I7 (y0x2 net49 net38 net031 P2 vdd vss) adder_for_multi
I13 (y1x3 net042 net034 net043 net056 vdd vss) adder_for_multi
I9 (y2x2 y3x1 net041 net010 net042 vdd vss) adder_for_multi
I8 (y1x2 net036 net45 net034 net038 vdd vss) adder_for_multi
I12 (y2x3 y3x2 net010 net050 net0135 vdd vss) adder_for_multi
I2 (y2x1 y3x0 vss net041 net036 vdd vss) adder_for_multi
simulatorOptions options reltol=1e-3 vabstol=1e-6 iabstol=1e-12 temp=27 \
    tnom=27 scalem=1.0 scale=1.0 gmin=1e-12 rforce=1 maxnotes=5 maxwarns=5 \
    digits=5 cols=80 pivrel=1e-3 ckptclock=1800 \
    sensfile="../psf/sens.output" checklimitdest=psf
tran tran stop=30n write="spectre.ic" writefinal="spectre.fc" \
    annotate=status maxiters=5
finalTimeOP info what=oppoint where=rawfile
modelParameter info what=models where=rawfile
element info what=inst where=rawfile
outputParameter info what=output where=rawfile
designParamVals info what=parameters where=rawfile
primitives info what=primitives where=rawfile
subckts info what=subckts  where=rawfile
saveOptions options save=allpub subcktprobelvl=2
```

Appendix C

VHDL-AMS Code for Simulations in Chapter 7

C.1 8 Bit Adder Top Level

```
-- VHDL-AMS code of 8 Bit Adder with one cell of adder having faulty component



library IEEE;
use IEEE.math_real.all;
use IEEE.electrical_systems.all;
use IEEE.std_logic_1164.all;
use work.all;


entity adder_nb_8bit is
port(
        terminal inputA: electrical_vector(7 downto 0);
        terminal inputB: electrical_vector(7 downto 0);
        terminal inputCIN: electrical_vector(7 downto 0);
        terminal SUM: electrical_vector(7 downto 0);
        terminal COUT: electrical_vector(7 downto 0)
);
end adder_nb_8bit;
```

```
architecture struct of adder_nb_8bit is


        component adder_nb_non_faulty is
    port(
        terminal inputA: electrical;
        terminal inputB: electrical;
        terminal inputCIN: electrical;
        terminal SUM: electrical;
        terminal COUT: electrical);

end component adder_nb_non_faulty;




    component adder_nb_faulty is
    port(
        terminal inputA: electrical;
        terminal inputB: electrical;
        terminal inputCIN: electrical;
        terminal SUM: electrical;
        terminal COUT: electrical);

end component adder_nb_faulty;


   terminal int1, int2, int3, int4, int5: electrical;           -- signal just like wire

begin

Gate1: adder_nb_non_faulty port map (inputA=>inputA(0), inputB=>inputB(0), inputCIN=>inputCIN(0)
Gate2: adder_nb_non_faulty port map (inputA=>inputA(1), inputB=>inputB(1), inputCIN=>inputCIN(1)
Gate3: adder_nb_faulty port map (inputA=>inputA(2), inputB=>inputB(2), inputCIN=>inputCIN(2), SU
Gate4: adder_nb_non_faulty port map (inputA=>inputA(3), inputB=>inputB(3), inputCIN=>inputCIN(3)
Gate5: adder_nb_non_faulty port map (inputA=>inputA(4), inputB=>inputB(4), inputCIN=>inputCIN(4)
Gate6: adder_nb_non_faulty port map (inputA=>inputA(5), inputB=>inputB(5), inputCIN=>inputCIN(5)
Gate7: adder_nb_non_faulty port map (inputA=>inputA(6), inputB=>inputB(6), inputCIN=>inputCIN(6)
Gate8: adder_nb_non_faulty port map (inputA=>inputA(7), inputB=>inputB(7), inputCIN=>inputCIN(7)


end struct;
```

C.1.1 Adder Slice

VHDL-AMS Code For Fault Free Adder

```
library IEEE;
use IEEE.math_real.all;
use IEEE.electrical_systems.all;
use IEEE.std_logic_1164.all;
use work.all;



entity adder_nb_non_faulty is
port(
        terminal inputA: electrical;
        terminal inputB: electrical;
        terminal inputCIN: electrical;
        terminal SUM: electrical;
        terminal COUT: electrical
);
end adder_nb_non_faulty;

architecture struct of adder_nb_non_faulty is



    component nand_nb is
    port (terminal a, b, c : electrical);
        end component nand_nb;

        component nor_nb is
    port (terminal a, b, c : electrical);
        end component nor_nb;

        component xnor_nb is
    port (terminal a, b, c : electrical);
        end component xnor_nb;



        component not_nb is
    port (terminal a, b : electrical);
        end component not_nb;
```

```
  terminal int1, int2, int3, int4, int5: electrical;          -- signal just like wire

begin

    Gate1: nand_nb port map (a=>inputB, b=>inputCIN, c=>int1);
    Gate2: nand_nb port map (a=>int1, b=>int4, c=>COUT);
        Gate3: nor_nb port map  (a=>inputB, b=>inputCIN, c=>int2);
    Gate4: not_nb port map  (a=>int2, b=>int3);
    Gate5: nand_nb port map (a=>int3, b=>inputA, c=>int4);
        Gate6: xnor_nb port map  (a=>inputB, b=>inputCIN, c=>int5);
    Gate7: xnor_nb port map  (a=>int5, b=>inputA, c=>SUM);

end struct;
```

VHDL-AMS Code For Faulty Adder having a component (NAND Gate) with resistive open

```
library IEEE;
use IEEE.math_real.all;
use IEEE.electrical_systems.all;
use IEEE.std_logic_1164.all;
use work.all;


entity adder_nb is
port(
        terminal inputA: electrical;
        terminal inputB: electrical;
        terminal inputCIN: electrical;
        terminal SUM: electrical;
        terminal COUT: electrical
);
end adder_nb;

architecture struct of adder_nb is


    component nand_nb is
    port (terminal a, b, c : electrical);
        end component nand_nb;
        component nor_nb is
    port (terminal a, b, c : electrical);
        end component nor_nb;

        component xnor_nb is
    port (terminal a, b, c : electrical);
        end component xnor_nb;

        component nand_faulty is
    port (terminal a, b, c : electrical);
        end component nand_nb2;

        component not_nb is
    port (terminal a, b : electrical);
        end component not_nb;




   terminal int1, int2, int3, int4, int5: electrical;           -- signal just like wire

begin

    Gate1: nand_nb port map (a=>inputB, b=>inputCIN, c=>int1);
    Gate2: nand_nb port map (a=>int1, b=>int4, c=>COUT);
        Gate3: nor_nb port map  (a=>inputB, b=>inputCIN, c=>int2);
    Gate4: not_nb port map  (a=>int2, b=>int3);
    Gate5: nand_faulty port map (a=>int3, b=>inputA, c=>int4);
        Gate6: xnor_nb port map  (a=>inputB, b=>inputCIN, c=>int5);
    Gate7: xnor_nb port map  (a=>int5, b=>inputA, c=>SUM);
```

C.1.2 Faulty Component : NAND Gate

VHDL-AMS Code For Faulty NAND Gate: resistive open

```
library IEEE;
use IEEE.math_real.all;
use IEEE.electrical_systems.all;
use IEEE.std_logic_1164.all;
use work.all;
```

```
entity nand_faulty is

        generic (td : real := 0.0;
                   tt : real := 0.0;
                   res: real :=2.0e+06;
                   coeff_a:real :=1.3442817920651515E-09;
                   coeff_b:real :=1.1339297457734438E+07;
                   coeff_c:real :=1.7259136506074388E+05;
                   coeff_d:real :=5.6456895070085156E+12;
                   coeff_e:real :=3.0249312551576300E-11;

                   thres : voltage := 0.35);

        port (terminal a, b, c : electrical);

end entity nand_faulty;

architecture abm of nand_faulty is
  constant vl:real:=0.0;
  constant vh:real:=1.2;
  signal vstate:real:=0.0;

  constant df: real :=(coeff_a*((res*res)+(coeff_b*res)))/((res*res)+(coeff_c*res)+coeff_d)+coef


  function real2time(tt: REAL) RETURN TIME IS
    begin
    return real(tt * 1.0e15) * 1 fs;
    end real2time;

  quantity va across a to electrical_ref;
  quantity vb across b to electrical_ref;
  quantity vc across ic through c to electrical_ref;
  quantity vb_d:voltage;

begin
vb_d==vb'delayed(df);



  -- purpose: Detect threshold crossing and assign event on output (d)
  -- type   : combinational
  -- inputs : vin'above(thres)
  -- outputs: pulse_signal
  process (va'above(thres), vb_d'above(thres)) is
  begin  -- PROCESS
    if va'above(thres) and vb_d'above(thres)then
                    vstate <=vl;
                else
                vstate <=vh;
                        end if;
  end process;



--vint2<=vstate'delayed (real2time(df));

vc==vstate'ramp(0.0225e-9);

end architecture abm;
```

C.2 Test Bench for 8 Bit Adder

```
library IEEE;
use IEEE.electrical_systems.all;
use IEEE.std_logic_1164.all;
use work.all;



entity testbench is
end entity testbench;



architecture abm of testbench is


component adder_nb_8bit is
port(
        terminal inputA: electrical_vector(7 downto 0);
        terminal inputB: electrical_vector(7 downto 0);
        terminal inputCIN: electrical_vector(7 downto 0);
        terminal SUM: electrical_vector(7 downto 0);
        terminal COUT: electrical_vector(7 downto 0)
);
end component adder_nb_8bit;
```

```
        terminal inputA_t: electrical_vector(7 downto 0);
        terminal inputB_t: electrical_vector(7 downto 0);
        terminal inputCIN_t: electrical_vector(7 downto 0);
        terminal SUM_t: electrical_vector(7 downto 0);
        terminal COUT_t: electrical_vector(7 downto 0);


  --quantity va across ia through inputA_t to electrical_ref;
  --quantity vb across ib through inputB_t to electrical_ref;
  --quantity vc across ic through inputCIN_t to electrical_ref;


  signal a_sig : real_vector(7 downto 0):=(1.2, 1.2, 1.2, 1.2, 1.2, 1.2, 1.2, 1.2);
  signal b_sig : real_vector(7 downto 0):=(1.2, 1.2, 1.2, 1.2, 1.2, 1.2, 1.2, 1.2);
  signal c_sig : real_vector(7 downto 0):=(1.2, 1.2, 1.2, 1.2, 1.2, 1.2, 1.2, 1.2);

begin


--va(0) == a_sig(0);
--va(1) == a_sig(1);
--va(2) == a_sig(2);
--va(3) == a_sig(3);
--va(4) == a_sig(4);
--va(5) == a_sig(5);
--va(6) == a_sig(6);
--va(7) == a_sig(7);
--vb == b_sig;
--vc == c_sig;


--vc == c_sig;

D0: adder_nb_8bit port map (
        inputA(7 downto 0)=>inputA_t(7 downto 0),
        inputB(7 downto 0)=>inputB_t(7 downto 0),
        --
        inputCIN(7 downto 0)=>inputCIN_t(7 downto 0),
        SUM(7 downto 0)=>SUM_t(7 downto 0),
        COUT(7 downto 0)=>COUT_t(7 downto 0)

        );
end architecture abm;
```

References

[1] 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*, vol. 25, no. 10, pp. 2181–2192, Oct. 2006.

[2] P. Engelke, I. Polian, M. Renovell, S. Kundu, B. Seshadri, and B. Becker, "On detection of resistive bridging defects by low-temperature and low-voltage testing," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 27, no. 2, pp. 327–338, Feb. 2008.

[3] S. Kundu, P. Engelke, I. Polian, and B. Becker, "On detection of resistive bridging defects by low-temperature and low-voltage testing," *14th Asian Test Symposium*, pp. 266–271, Dec. 2005.

[4] 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," *VLSI Test Symposium*, pp. 171–178, 2004.

[5] *Intel XScale Core Developers Manual, 2003*, Intel, Inc, June 2006, http://www.intel.com/design/intelxscale/.

[6] *Crusoe processor documentation*, Transmeta Corporation, April 2002, http://www.transmeta.com/.

[7] *ARM IEM Technology*, ARM, Ltd, June 2006, http://www.arm.com/products/esd/iem_home.html/.

[8] M. Abramovici, M. A. Breuer, and A. D. Friedman, *Digital Systems Testing and Testable Design.* Jaico Publishing House, 2001.

[9] T. Hiraide, K. O. Boateng, H. Konishi, K. Itaya, M. Emori, H. Yamanaka, and T. Mochiyama, "Bist-aided scan test - a new method for test cost reduction," *21st VLSI Test Symposium*, 27 April-1 May 2003.

[10] 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 System*, vol. 27, no. 6, pp. 1117–1127, 2008.

[11] N. B. Z. Ali, M. Zwolinski, B. M. Al-Hashimi, and P. Harrod, "Dynamic voltage scaling aware delay fault testing," *Eleventh IEEE European Test Symposium*, pp. 1530–1877, 2006.

[12] C.-I. H. Chen, "Behavioral test generation/fault simulation," *IEEE Potentials*, vol. 22, no. 1, pp. 27–32, 2003.

[13] C.-I. H. Chen and S. Perumal, "Analysis of the gap between behavioral and gate level fault simulation," *Sixth Annual IEEE International Conference and Exhibition on ASIC*, pp. 144 – 147, 1993.

[14] S. Ghosh, "Behavioral-level fault simulation," *IEEE Design and Test of Computers*, vol. 5, no. 3, pp. 31–42, 1988.

[15] M. Zwolinski and A. D. Brown, "Behavioural modelling of analogue faults in VHDL-AMS - a case study," *International Symposium on Circuits and Systems*, pp. 632–635, 2004.

[16] Z. Li, X. Lu, W. Qiu, W. Shi, and D. M. H. Walker, “A circuit level fault model for resistive bridges,” *ACM Transactions on Design Automation of Electronic Systems*, vol. 8, no. 4, pp. 546–559, 2003.

[17] V. Litovski, M. Andrejevic, and M. Zwolinski, “Behavioural modelling, simulation, test and diagnosis of mems using anns,” *International Symposium on Circuits and Systems*, pp. 5182–5185, 2005.

[18] *International Technology Roadmap for Semiconductors, 2007 Edition: Test and test equipments*, ITRS, May 2008, http://www.itrs.net/Links/2007ITRS/2007_Chapters/2007_Test.pdf/.

[19] R. Garcia, “Rethink fault models for submicron-ic test,” *Test and Measurement World*, pp. 1–6, 2001.

[20] *The National Technology Roadmap for Semiconductors*, Semiconductor Industry Association, May 2008, http://www.sematech.org/.

[21] R. C. Tekumalla, “On test set generation for efficient path delay fault diagnosis,” *18th IEEE VLSI Test Symposium*, pp. 343–348, 2000.

[22] M. Sivaraman and A. J. Strojwas, “Path delay fault diagnosis and coverage-a metric and an estimation technique,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 20, no. 3, pp. 440–457, Mar 2001.

[23] D. M. Walker, “Architectures for catastrophic and delay fault tolerance,” *Fifth Annual IEEE International Conference on Wafer Scale Integration*, pp. 47–56, 1993.

[24] T. D. Smith, T. A. DeLong, B. W. Johnson, and J. A. Profeta, “An algorithm based fault tolerance technique for safety-critical applications,” *Annual Reliability and Maintainability Symposium*, pp. 278–285, Jan 1997.

[25] T. S. Perraju, S. P. Rana, and S. P. Sarkar, "Specifying fault tolerance in mission critical systems," *IEEE High-Assurance Systems Engineering Workshop*, pp. 24–31, Oct 1996.

[26] M. Peercy and P. Banerjee, "Fault tolerant vlsi systems," *IEEE Proceedings*, vol. 81, no. 5, pp. 745–758, May 1993.

[27] S. Hronik, "Industry acceptance of boundary scan," *WESCON/94. 'Idea/Microelectronics'*, pp. 518–524, 1994.

[28] R. Dellecker, "Boundary-scan bursts into the modern production facility," *IEEE AUTOTESTCON Proceedings*, pp. 184–186, 2000.

[29] *Test Technology Overview*, Center for Electronic Design, Communications, and Computing, PSU, July 2008, http://www.cedcc.psu.edu/ee497i/rassp_43/sld001.htm.

[30] B. Eklow, "What do you mean my board test stinks?" *International Test Conference*, p. 1423, Oct. 2004.

[31] B. Nadeau-Dostie, *Design for at-speed test, Diagnosis and Measurement.* Kluwer Academic Publishers, 2000.

[32] I. Pomeranz and S. M. Reddy, "Improving the stuck-at fault coverage of functional test sequences by using limited-scan operations," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 12, no. 7, pp. 780–788, July 2004.

[33] M. Mayberry, J. Johnson, N. Shahriari, and M. Tripp, "Realizing the benefits of structural test for intel microprocessors," *International Test Conference*, pp. 456–463, 2002.

[34] B. Kruseman, A. Majhi, C. Hora, S. Eichenberger, and J. Meirlevede, "Systematic defects in deep sub-micron technologies," *International Test Conference*, pp. 290–299, Oct. 2004.

[35] M. Pecht, R. Radojcic, and G. Rao, *Guidebook for Managing Silicon Chip Reliability.* CRC Press, 1998.

[36] J. Alt and U. Mahlstedt, “Simulation of non-classical faults on the gate level-fault modeling,” *Eleventh Annual 1993 IEEE VLSI Test Symposium*, pp. 351–354, Apr 1993.

[37] M. L. Bushnell and V. D. Agrawal, *Essentials of Electronic Testing for Digital, Memory, and Mixed-Signal VLSI Circuits.* Kluwer Academic Publishers, 2001.

[38] B. Kruseman, “Comparison of defect detection capabilities of current-based and voltage-based test methods,” *IEEE European Test Workshop*, pp. 175–180, 2000.

[39] B. Kruseman, S. V. den Oetelaar, and J. Rius, “Comparison of i/sub ddq/ testing and very-low voltage testing,” *International Test Conference*, pp. 964–973, 1992.

[40] J.-Y. Chang and E. J. McCluskey, “Detecting delay flaws by very-low-voltage testing,” *International Test Conference*, pp. 367–376, 1996.

[41] A. K. Majhi and V. K. Agrawal, “Delay fault models and coverage,” *Eleventh International Conference on VLSI Design*, pp. 364–369, 1998.

[42] 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?” *International Test Conference*, pp. 95–104, 2000.

[43] R. Rajsuman, “Iddq testing for cmos vlsi,” *Proceedings of the IEEE*, vol. 88, no. 4, pp. 544–568, Apr 2000.

[44] Y. Higami, Y. Takamatsu, K. K. Saluja, and K. Konshita, “Fault models and test generation for iddq testing,” *Asia and South Pacific Design Automation Conference*, pp. 509–514, 2000.

[45] A. Chehab, A. Kayssi, and A. Ghandour, "Transient current testing of gate-oxide shorts in cmos," *2nd International Design and Test Workshop*, pp. 77–81, Dec. 2007.

[46] T. W. Williams, R. H. Dennard, R. Kapur, M. R. Mercer, and W. Maly, "Iddq test: sensitivity analysis of scaling," *International Test Conference*, pp. 786–792, Oct 1996.

[47] K. Fuchs, F. Fink, and M. Schulz, "Dynamite: an efficient automatic test pattern generation system for path delay faults," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 10, no. 10, pp. 1323–1335, Oct 1991.

[48] M. S. Gaur, "Integration of built in self test during behavioural synthesis," Ph.D. dissertation, Electronics and Computer Science, University of Southampton, 2004.

[49] E. Vandris and G. Sobelman, "Switch-level differential fault simulation of mos vlsi circuits," *Vlsi Design*, vol. 4, no. 3, pp. 217–229, 1996.

[50] Y. Liao and D. M. H. Walker, "Optimal voltage testing for physically-based faults," *VLSI Test Symposium*, pp. 344–353, May 1996.

[51] J. Gracia, J. C. Baraza, D. Gil, and P. J. Gil, "Comparison and application of different vhdl-based fault injection techniques," *IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems*, pp. 233–241, 2001.

[52] P. Oikonomakos, "High level synthesis for on-line testability," Ph.D. dissertation, Electronics and Computer Science, University of Southampton, 2004.

[53] L. Antoni, R. Leveugle, and B. Feher, "Using run-time reconfiguration for fault injection in hardware prototypes," *IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems*, pp. 245–253, 2002.

[54] B. Benware, C. Schuermyer, S. Ranganathan, R. Madge, P. Krishnamurthy, N. Tamarapalli, K.-H. Tsai, and J. Rajski, "Impact of multiple-detect test patterns on product quality," *International Test Conference*, vol. 1, pp. 1031–1040, 30-Oct. 2, 2003.

[55] N. Mukherjee, T. Chakraborty, and R. Karri, "Built in self test: a complete test solution for telecommunication systems," *IEEE Communications Magazine*, vol. 37, no. 6, pp. 72–78, Jun 1999.

[56] V. D. Agrawal, C. R. Kime, and K. K. Saluja, "A tutorial on built-in self-test. i. principles," *IEEE Design and Test of Computers*, vol. 10, no. 1, pp. 73–82, Mar 1993.

[57] U. Ingelsson, P. Rosinger, S. Khursheed, B. M. Al-Hashimi, and P. Harrod, "Resistive bridging faults dft with adaptive power management awareness," *Asian Test Symposium*, pp. 101–106, Oct. 2007.

[58] M. Hashimoto, K. Fujimori, and H. Onodera, "Standard cell libraries with various driving strength cells for 0.13, 0.18 and 0.35 /spl mu/m technologies," *Asia and South Pacific Design Automation Conference*, pp. 589–590, Jan. 2003.

[59] K. S. Kim, S. Mitra, and P. G. Ryan, "Delay defect characteristics and testing strategies," *IEEE Design and Test of Computers*, vol. 20, no. 5, pp. 8–16, Sept.-Oct. 2003.

[60] J. Gatej, L. Song, C. Pyron, and R. Raina, "Evaluating ate features in terms of test escape rates and other cost of test culprits," *International Test Conference*, pp. 1040–1049, 2002.

[61] J. C.-M. Li and E. J. McCluskey, “Diagnosis of sequence-dependent chips,” *20th IEEE VLSI Test Symposium*, pp. 187–192, 2002.

[62] X. Lu, Z. Li, W. Qiu, D. M. Walker, and W. Shi, “Parade: parametric delay evaluation under process variation [ic modeling],” *5th International Symposium on Quality Electronic Design*, pp. 276–280, 2004.

[63] C. J. Lin and S. M. Reddy, “On delay fault testing in logic circuits,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 6, no. 5, pp. 694–703, 1987.

[64] W. Qiu, “Fault simulation and test generation for small delay faults,” Ph.D. dissertation, Computer Science, Texas A and M University, 2006.

[65] B. Bennetts and B. Kruseman, *Delay Fault Testing: From Basic to ASICS: Tutorial at ETS 2006*, 2006.

[66] M. Geilert, J. Alt, and M. Zimmermann, “On the efficiency of the transition fault model for delay faults,” *IEEE International Conference on Computer-Aided Design*, pp. 272–275, 1990.

[67] A. Krstic and T. Cheng, *Delay Fault Testing for VLSI Circuits.* Kluwer Publications, 1998.

[68] M. M. V. Kumar and S. Tragoudas, “Quality transition fault tests suitable for small delay defects,” *IEEE International Conference on Computer Design: VLSI in Computers and Processors*, pp. 468–470, Oct. 2005.

[69] M. V. Kumar and S. Tragoudas, “High-quality transition fault atpg for small delay defects,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 26, no. 5, pp. 983–989, May 2007.

[70] J. A. Waicukauski, E. Lindbloom, B. K. Rosen, and V. S. Iyengar, “Transition fault simulation,” *IEEE Design and Test of Computers*, vol. 4, no. 2, pp. 32–38, April 1987.

[71] M. T. Schmitz, "Energy minimisation techniques for distributed embedded systems," Ph.D. dissertation, Electronics and Computer Science, University of Southampton, 2003.

[72] A. K. Pramanick and S. M. Reddy, "On the fault coverage of gate delay fault detecting tests," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 16, no. 1, pp. 78–94, Jan 1997.

[73] B. Seshadri, I. Pomeranz, S. M. Reddy, and S. Kundu, "On path selection for delay fault testing considering operating conditions [logic ic testing]," *Eighth IEEE European Test Workshop*, pp. 141–146, 2003.

[74] T. Williams, B. Underwood, and M. Mercer, "The interdependence between delay-optimization of synthesized networks and testing," *28th ACM/IEEE Design Automation Conference*, pp. 87–92, 1991.

[75] K.-T. Cheng and H.-C. Chen, "Classification and identification of non-robust untestable path delay fault," *IEEE Transactions On Computer-Aided Design of Integrated Circuits and System*, vol. 15, no. 8, pp. 845–853, 1996.

[76] Z. Li, X. Lu, W. Qiu, W. Shi, and D. M. H. Walker, "A circuit level fault model for resistive opens and bridges," *VLSI Test Symposium*, pp. 379–384, April-1 May 2003.

[77] R. R. Montanes, E. Bruis, and J. Figueras, "Bridging defects resistance measurements in a cmos process," *International Test Conference*, pp. 892–899, Sep 1992.

[78] V. Krishnaswamy, A. Ma, and P. Vishakantaiah, "A study of bridging defect probabilities on a pentium (tm) 4 cpu," *International Test Conference*, pp. 688–695, 2001.

[79] A. Jee and J. F. Ferguson, “Carafe: an inductive fault analysis tool for cmos vlsi circuits,” *VLSI Test Symposium, 1993. Digest of Papers.*, pp. 92–98, 1993.

[80] T. M. Storey and W. Maly, “Cmos bridging fault detection,” *International Test Conference*, pp. 842–851, Sep 1990.

[81] I. Polian, P. Engelke, B. Becker, S. Kundu, J.-M. Galliere, and M. Renovell, “Resistive bridge fault model evolution from conventional to ultra deep submicron,” *23rd IEEE VLSI Test Symposium*, pp. 343–348, May 2005.

[82] M. Renovell, P. Huc, and Y. Bertrand, “Cmos bridging fault modeling,” *IEEE VLSI Test Symposium*, pp. 392–397, Apr 1994.

[83] ——, “The concept of resistance interval: a new parametric model for realistic resistive bridging fault,” *VLSI Test Symposium*, pp. 184–189, 1995.

[84] J. Sousa, F. Goncalves, and J. Teixeira, “Ic defects-based testability analysis,” *International Test Conference*, pp. 500–509, Oct 1991.

[85] J. Brown and R. Blanton, “Exploiting regularity for inductive fault analysis,” *Proceedings 24th IEEE VLSI Test Symposium, 2006*, pp. 364–369, May 2006.

[86] Y. Liao and D. M. H. Walker, “Fault coverage analysis for physically-based cmos bridging faults at different power supply voltages,” *International Test Conference*, pp. 767–775, 1996.

[87] S. Chakravarty and A. Jain, “Fault models for speed failures caused by bridges and opens,” *IEEE VLSI Test Symposium*, pp. 373–378, 2002.

[88] M. Renovell, P. Huc, and Y. Bertrand, “A unified model for inter-gate and intra-gate cmos bridging fault: the configuration ratio,” *Third Asian Test Symposium*, pp. 170–175, Nov 1994.

[89] A. Keshk, Y. Miura, and K. Kinoshita, "Procedure to overcome the byzantine general's problem for bridging faults in cmos circuits," *Eighth Asian Test Symposium*, pp. 121–126, 1999.

[90] M. Favalli, M. Dalpasso, P. Olivo, and B. Ricco, "Analysis of dynamic effects of resistive bridging faults in cmos and bicmos digital ics," *International Test Conference*, pp. 865–874, Oct 1993.

[91] H. T. Vierhaus, W. Meyer, and U. Glaser, "Cmos bridges and resistive transistor faults: Iddq versus delay effects," *International Test Conference*, pp. 83–91, 17-21 Oct 1993.

[92] V. R. Sar-Dessai and D. M. H. Walker, "Resistive bridge fault modeling, simulation and test generation," *International Test Conference*, pp. 596–605, 1999.

[93] C. F. Hawkins, J. M. Soden, A. w. Righter, and J. F. Ferguson., "Defect classes-an overdue paradigm for cmos ic testing," *International Test Conference*, pp. 413–425, 1994.

[94] M. Renovell and Y. Bertrand, "Test strategy sensitivity to defect parameters," *International Test Conference, 1997*, pp. 607–616, 1-6 Nov 1997.

[95] O. V. Maiuri and W. R. Moore, "Implications of voltage and dimension scaling on cmos testing: the multidimensional testing paradigm," *16th IEEE VLSI Test Symposium*, pp. 22–27, 26-30 Apr 1998.

[96] S. Midkiff and S. Bollinger, "Classification of bridging faults in cmos circuits: experimental results and implications for test," *Eleventh Annual 1993 IEEE VLSI Test Symposium*, pp. 112–115, Apr 1993.

[97] H. Konuk and F. J. Ferguson, "Oscillation and sequential behavior caused by interconnect opens in digital cmos circuits," *International Test Conference*, pp. 597–606, Nov 1997.

[98] N. Devtaprasanna, A. Gunda, P. Krishnamurthy, S. Reddy, and I. Pomeranz, "Test generation for open defects in cmos circuits," *21st IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems*, pp. 41–49, Oct. 2006.

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

[100] J. Li, C.-W. Tseng, and E. McCluskey, "Testing for resistive opens and stuck opens," *International Test Conference*, pp. 1049–1058, 2001.

[101] V. Avendano, V. Champac, and J. Figueras, "Signal integrity loss in bus lines due to open shielding defects," *The Eighth IEEE European Test Workshop*, pp. 79–84, May 2003.

[102] R. R. Montanes, J. P. de Gyvez, and P. Volf, "Resistance characterization for weak open defects," *IEEE Design and Test of Computers*, vol. 19, no. 5, pp. 18–26, 2002.

[103] C. L. Henderson, J. M. Soden, and C. F. Hawkins, "The behavior and testing implications of cmos ic logic gate open circuits," *International Test Conference*, pp. 302–310, Oct 1991.

[104] A. Czutro, N. Houarche, P. Engelke, I. Polian, M. Comte, M. Renovell, and B. Becker, "A simulator of small-delay faults caused by resistive-open defects," *European Test Symposium*, pp. 113–118, May 2008.

[105] T. D. Burd, T. A. Pering, A. J. Stratakos, and R. W. Brodersen, "A dynamic voltage scaled microprocessor system," *IEEE Journal of Solid-State Circuits*, vol. 35, no. 11, pp. 1571–1580, 2000.

[106] J. Rajski, “Dft for high-quality low cost manufacturing test,” *Asian Test Symposium(ATS)*, pp. 3–8, 2001.

[107] M. Rodriguez-Irago, J. Andina, F. Vargas, M. Santos, I. Teixeira, and J. Teixeira, “Dynamic fault test and diagnosis in digital systems using multiple clock schemes and multi-vdd test,” *11th IEEE International On-Line Testing Symposium*, pp. 281–286, July 2005.

[108] M. Rodriguez-Irago, J. Rodriguez Andina, F. Vargas, J. Semiacy, I. Teixeira, and J. Teixeira, “Dynamic fault detection in digital systems using dynamic voltage scaling and multi-temperature schemes,” *12th IEEE International On-Line Testing Symposium*, pp. 6 pp.–, July 2006.

[109] H. Hao and E. J. McCluskey, “Very low voltage testing for weak cmos logic ics,” *International Test Conference*, pp. 275–283, 1993.

[110] J.-Y. Chang and E. J. McCluskey, “Quantitative analysis of very low voltage testing,” *VLSI Test Symposium*, pp. 332–337, 1996.

[111] S. Bota, M. Rosales, J. Rossello, and J. Segura, “Low v/sub dd/ vs. delay: is it really a good correlation metric for nanometer ics?” *24th IEEE VLSI Test Symposium*, pp. 6 pp.–363, May 2006.

[112] H. Yan, A. Singh, and G. Xu, “Delay defect characterization using low voltage test,” *14th Asian Test Symposium*, pp. 8–13, Dec. 2005.

[113] K. D. Wagner and E. J. McCluskey, “Effect of supply voltage on circuit propagation delay and test applications,” *IEEE International Conference on Computer Aided Design*, pp. 42–44, 1985.

[114] S. Nassif, “Delay variability: sources, impacts and trends,” *IEEE International Solid State Circuits*, pp. 368–369, 2000.

[115] E. J. McCluskey and C.-W. Tseng, “Stuck-fault tests vs. actual defects,” *International Test Conference*, pp. 336–342, 2000.

[116] C.-W. Tseng, E. Mccluskey, X. Shao, and D. Wu, “Cold delay defect screening,” *18th IEEE VLSI Test Symposium*, pp. 183–188, 2000.

[117] M. Renovell, P. Huc, and Y. Bertrand, “Bridging fault coverage improvement by power supply control,” *VLSI Test Symposium*, pp. 338–343, Apr-1 May 1996.

[118] S. Kundu, “Private communication,” *University of Massachusetts. Amherst*, July 2007.

[119] C. Q. Tran, H. Kawaguchi, and T. Sakurai, “Low-power high-speed level shifter design for block-level dynamic voltage scaling environment,” *International Conference on Integrated Circuit Design and Technology*, 2005.

[120] B. Zhang, L. Liang, and X. Wang, “A new level shifter with low power in multi-voltage system,” *International Conference on Solid-State and Integrated Circuit Technology*, pp. 1857–1859, 2006.

[121] R. Garg, G. Mallarapu, and S. P. Khatri, “A single-supply true voltage level shifter,” *Design, Automation and Test in Europe*, pp. 979–984, 10-14 March 2008.

[122] R. Aitken, “Private communication,” *Arm Ltd, USA*, July 2006.

[123] Q. A. Khan, S. K. Wadhwa, and K. Misri, “A single supply level shifter for multi-voltage systems,” *VLSI Design*, 2006.

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

[125] *Intel PXA270 Processor for Embedded Computing- Technical Documents*, Intel, Inc, June 2006, http://www.intel.com/design/embeddedpca/products/pxa270/techdocs.htm/.

[126] B. Kruseman and M. Heiligers, "On test conditions for the detection of open defects," *Design, Automation and Test in Europe*, pp. 896–901, 2006.

[127] M. Tripp, S. Picano, and B. Schnarch, "Drive only at speed functional testing; one of the techniques intel is using to control test costs," *International Test Conference*, pp. 8 pp.–, Nov. 2005.

[128] J. Wang, D. M. H. Walker, A. Majhi, B. Kruseman, G. Gronthoud, L. E. Villagra, P. van de Wiel, and S. Eichenberger, "Power supply noise in delay testing," *International Test Conference*, pp. 1–10, Oct. 2006.

[129] A. Krstic, Y.-M. Jiang, and K.-T. Cheng, "Delay testing considering power supply noise effects," *International Test Conference*, pp. 181–190, 1999.

[130] M. Horowitz, T. Indermaur, and R. Gonzalez, "Low-power digital design," *Symposium On Low Power Electronics*, pp. 8–11, Oct 1994.

[131] J. Zeng, M. Abadir, G. Vandling, L. Wang, A. Kolhatkar, and J. Abraham, "On correlating structural tests with functional tests for speed binning of high performance design," *International Test Conference*, pp. 31–37, Oct. 2004.

[132] B. Kruseman, A. K. Majhi, G. Gronthoud, and S. Eichenberger, "On hazard-free patterns for fine-delay fault testing," *International Test Conference*, pp. 213–222, 26-28 Oct. 2004.

[133] C.-W. Tseng, J. Li, and E. McCluskey, "Experimental results for slow-speed testing," *VLSI Test Symposium*, pp. 37–42, 2002.

[134] W. Needham, C. Prunty, and E. H. Yeoh, “High volume microprocessor test escapes, an analysis of defects our tests are missing,” *International Test Conference*, pp. 25–34, Oct 1998.

[135] N. H. Weste and K. Eshraghian, *Principles of CMOS VLSI Design.* Addison-Wesley Publishing Company, 1992.

[136] J. M. Rabaey, *Digital Integrated Circuits, A Design Perspective.* Prentice Hall International Edition, 1996.

[137] A. P. Chandrakasan, S. Sheng, and R. W. Brodersen, “Low-power cmos digital design,” *IEEE Journal of Solid-State Circuits*, vol. 27, no. 4, pp. 473–484, Apr 1992.

[138] H. Y. Abujbara and S. Al-Arian, “Testing and simulation of degrading faults,” *37th Midwest Symposium on Circuits and Systems*, vol. 1, pp. 242–246 vol.1, Aug 1994.

[139] J. ni Ohban, V. G. Moshnyaga, and K. Inoue, “Multiplier energy reduction through bypassing of partial products,” *Asia-Pacific Conference on Circuits and Systems*, vol. 2, pp. 13–17 vol.2, 2002.

[140] D. Flynn, “Private communication,” *ARM Ltd, Cambridge*, June 2008.

[141] H. Takahashi, N. Iuchi, and Y. Takamatsu, “Test generation for combinational circuits with multiple faults,” *Pacific Rim International Symposium on Fault Tolerant Systems*, pp. 212–217, 1991.

[142] Z. Wang, M.-S. M., T. K.-H., and J. Rajski, “Analysis and methodology for multiple-fault diagnosis,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 25, no. 3, pp. 558–575, 2006.

[143] *System Vision*, Mentor Graphics, December 2008, http://www.mentor.com/products/sm/system_integration_simulation_analysis/systemvision/.

[144] *HSPICE*, Synopsis, December 2008, http://www.synopsys.com/Tools/Verification/AMSVerification/CircuitSimulation/HSPICE/Pages/default.aspx.

[145] *Standrad Datasert MGH09*, National Institute Standard and Testing, March 2008, http://www.itl.nist.gov/div898/strd/nls/data/mgh09.shtml.

[146] *Coefficient of determination*, TAMU, October 2007, http://www.stat.tamu.edu/stat30x/notes/node47.html/.

[147] P. Chowdhury and C. Chakrabarti, "Static task-scheduling algorithms for battery-powered dvs systems," *IEEE Transactions On Very Large Scale Integration (VLSI) Systems*, vol. 13, no. 2, pp. 1571–1580, 2005.