Computation Reduction for Statistical Analysis of the Effect of Nano-CMOS Variability on Integrated Circuits

A THESIS SUBMITTED TO THE UNIVERSITY OF MANCHESTER
FOR THE DEGREE OF DOCTOR OF PHILOSOPHY
IN THE FACULTY OF ENGINEERING AND PHYSICAL SCIENCES

2012

By
Zheng Xie
School of Computer Science

Contents

List of Figures

Abstract

The intrinsic atomistic variability of nano-scale integrated circuit (IC) technology must be taken into account when analysing circuit designs to predict likely yield. These 'atomistic' variabilities are random in nature and are so great that new circuit analysis techniques are needed which adopt a statistical treatment of the variability of device performances. Monte Carlo (MC) based statistical techniques aim to do this by analysing many randomized copies of the circuit. The randomization can take into account correlation between parameters due to both intra-die and inter-die effects. A major problem is the computational cost of carrying out sufficient analyses to produce statistically reliable results.

The use of principal components analysis (PCA) and 'Statistical Behavioural Circuit Blocks (SBCB)' is investigated as a means of reducing the dimensionality of the analysis, and this is combined with an implementation of 'Statistical Blockade (SB)' to achieve significant reduction in the computational costs. The purpose of SBCBs is to model the most important aspects of the device's or circuit building block's behaviour, to an acceptable accuracy, with a relatively small number of parameters. The SB algorithm applies Extreme Value Theory (EVT) to circuit analysis by eliminating randomised parameter vectors that are considered unlikely to produce 'rare event' circuits. These circuits are needed for circuit yield failure predictions and occur on the 'tails' of Gaussian-like probability distributions for circuit performances.

Versions of the circuit analysis program 'SPICE' with a Python harness called RandomSPICE are used to produce SBCBs by generating and statistically analysing randomized transistor-level versions of the sub-blocks for which behavioural models are required. The statistical analysis of circuits employing these sub-blocks is achieved by a new MATLAB harness called RandomLA. The computational time savings that may be achieved are illustrated by the statistical analysis of representative circuits. A computation time reduction of 98.7% is achieved for a commonly used asynchronous circuit element.

Quasi-Monte Carlo (QMC) analysis with 'low discrepancy sequences (LDS)' is introduced for further computation reduction. QMC analysis using SBCB behavioural models with SB is evaluated by applying it to more complex examples and comparing the results with those of transistor level simulations. The analysis of SRAM arrays is taken as a case study for VLSI circuits containing up to 1536 transistors, modeled with parameters appropriate to 35nm technology. Significantly faster statistical analysis is shown to be possible when the aim is to obtain predictions of the yield for fabrication. Saving of up to 99.85% in computation time was obtained with larger circuits.

Declaration

No portion of the work referred to in the thesis has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.

Copyright

Acknowledgements

I would like to express my thanks to my supervisor, Dr. Doug Edwards, for valuable guidance and support during my Ph.D. research and thesis writing. I especially appreciated the free research space Doug provided, which enabled me to develop my research interests in this interdisciplinary field.

Thanks also to the people of the Advanced Processor Technologies group for those inspiring and interesting seminars. The author acknowledges the financial support from EPSRC and the collaboration among the people of the Nano-CMOS pilot project – Meeting the Design Challenges of Nano-CMOS Electronics.

The encouragement and inspiration from my parents and family in China have been a major driving force and source of comfort.

Finally, and certainly not least, the encouragement, help and support of my husband Philip and my family in England are lovingly acknowledged as the source of any success this thesis brings. Without Philip, I could never have undertaken this work.

Chapter 1

Introduction

1.1 Introduction to the Research Area

Variability has always been a problem in electronic circuit design, especially in integrated circuit design. Anticipating the impact of variability on performance is a critical aspect of design procedures. Before nano-scale technology, the variability came mainly from imperfect control of the fabrication processes which caused device performance to vary from wafer to wafer, and from die to die within each wafer. This 'inter-die' variability could be assumed to have a global nature for each die and was catered for in the design process by making sure that the circuit would work when all the parameters of each of its devices were at the 'worst case' extremes of their anticipated values. The result of a circuit design procedure would therefore be analysed with the set of parameters for each copy of a given device, for example a MOS transistor, at the 'worst case corners' of possible values. The results from the analysis could then be used as feedback in an optimization process leading to the finalised circuit.

As technology scaling has reached the nano-scale region, variability in device performance within each die, that is 'intra-die' variability, is becoming a much more important consideration in the design of integrated circuits [1]. Now that CMOS technology has reduced gate sizes to 90nm and below, and some dimensions are approaching atomic scales, intrinsic atomic scale variations such as line edge roughness and dopant granularity have become the main sources of variation [2].

Those 'atomistic' variabilities are random in nature and result in random 'within-die' fluctuation which cannot be disregarded. The potential variations are so great that traditional variability analysis, based on 'worst case' corner-models [3] and guard-bands for parameter variations, is likely to be very pessimistic in its estimations of the effects of the variability. Consequently, new circuit analysis techniques are needed which adopt a statistical treatment of the intra-die variability of device performances.

As the significance of process variations has grown with decreasing device sizes, it has become clear that traditional design methodologies, both for analysis and optimization, are no longer acceptable [3]. This has led to much interest in statistical modelling techniques that can be used to enable statistical analysis and optimization. Statistical analysis can take into account parameter variation on all portions of a design in a single comprehensive computational procedure, allowing the impact on yield to be efficiently estimated.

Because of the high dimensionality of the parameter space, it is very difficult to derive analytical models of large scale ICs for analysis. Sets of ordinary or partial differential equations describing specific circuit performance parameters, like timing or yield, in terms of the huge number of parameters would be very difficult to derive analytically. Therefore the use of conventional statistical methods, based on the analysis of such equations, has restricted applicability for the variability analysis of nano-scale ICs. With Monte Carlo simulations, the integrated circuits are simulated directly, and there is no need to derive differential equations that describe the dependency of the properties of interest on circuit parameters. The only requirement is that the circuit properties that are to be modeled are capable of being described by probability density functions (pdf's) that are dependent on the pdfs of circuit parameters and input variables. Monte Carlo simulation proceeds by generating a random sample of the value of each circuit parameter and input variable, based on the known pdf's. It then simulates the circuit thus obtained, using a package such as SPICE, to compute a random sample of the circuit property of interest. The process is repeated many times to obtain a sequence of samples of the property of interest from which statistical models can be derived. For example, the shape of the

property's pdf may be inferred and its mean and standard deviation can readily be derived, assuming there are sufficient observations. In some cases the circuit parameters may be assumed to be statistically independent; meaning that there is no correlation between the samples of one circuit parameter and any other. This assumption may be reasonable for some aspects of atomistic variations. However, there is good reason to believe that the variation that occurs in each component will be to some extent correlated to that of other devices on the same IC especially when they are close to each other. The effect of this correlation is 'intra-die' variation which must be given consideration.

Monte Carlo analyses are particularly suitable to nano-scale IC statistical simulation to achieve statistical estimates of properties of interest. Conclusions may be drawn that are representative of the true behaviour of the circuit. To quantify these conclusions, statistical averages may be produced based on many randomised examples. The statistical reliability of the conclusions generally improves as the number of examples increases, though the rate of improvement can often be increased by carefully choosing the examples. For such applications, MC techniques are simple, flexible, robust and scalable to exceptionally large numbers of parameters. In principle, they allow arbitrary accuracy given sufficient computation. The statistical distribution of circuit performances in response to carefully randomised vectors of device parameters may be used for estimating anticipated circuit yield, failure probability and other performance measures.

A major problem is the computational cost of carrying out sufficient simulations to produce statistically reliable results for all but the most trivial circuits. For circuit simulation at transistor level, each transistor model may have many parameters and there may be a large number of transistors in the circuit or sub-circuit being simulated. A very large number of MC analyses may be required because of the large number of parameters.

1.2 Research Motivation and Context for This Project

The research motivation and context for this PhD project is the EPSRC pilot project:

'Meeting the design challenges of nano-CMOS electronics' [7]. The aim of the pilot study is to propose new design styles that cope better with device variability. We aim to demonstrate that it is possible to produce circuit designs that are optimized or improved in their suitability for sub-45 nanometre technology by using statistical models of variability, and that the computation costs required for the statistical analysis can be made feasible.

There are five university partners working on the development of various elements of statistical simulation using approaches which will be suitable for grid implementation. Their efforts are interlinked [7]. The Device Modelling Group of Glasgow University focus on process and device simulation. One outcome of their work so far is the RandomSPICE package which is used in this PhD project. The Microsystems Technology Group of Glasgow University develop integrated circuit/device simulators, circuit level compact models and parameter extraction strategies. Their results include the sets of randomized mosfet parameters that are used by the RandomSPICE circuit randomization process. The Electronic Systems Design Group of Southampton University produce behavioural models for standard cell libraries. Their work will be studied to find accurate ways of producing low complexity models of sub-circuits with parameters that may be randomised from knowledge of their statistical properties, as extracted by RandomSPICE analyses. The Mixed-Mode Design Group of Edinburgh have been developing techniques for circuit-level noise simulation, and the Intelligent System Group of York University have developed evolutionary circuit and system design techniques.

The Advanced Processor Technology of Manchester University group has traditionally focused on the development and use of statistical analysis tools for use in the design and optimization of asynchronous (self-timed) circuits and chip multiprocessors. This has led us to investigate new design techniques that cope better with device variability and reliability. To be able to do this, we firstly need to investigate how to estimate the likely effect of the parameter variations predicted by device level models on the performance of clocked and self-timed implementation styles. Then we can consider how the ability to obtain these estimates can be used to improve future synthesis tools for nano-scale ICs.

1.3 Research Approach

Statistical variability analysis has been available in the commercial package 'HSPICE' for some time. HSPICE offers an approach to MC simulation that is professionally designed and well adapted to the demands of commercial manufacturers and circuit design companies. However it does not cater for all the computation reduction techniques we wish to investigate (e.g. Statistical Blockade) and is not ideally suited to a research project because it implements proprietary approaches and does not have the flexibility needed to investigate new research ideas. NGSPICE is a mixed-signal (analogue and digital) circuit simulator combining three open source software packages: SPICE3, Cider and Xspice. It is under continuing development as part of the 'gEDA' project [37] for developing a full GNU public licensed suite of electronic circuit design (EDA) tools. It did not have any MC simulation facilities when this project began, though rudimentary ones have very recently been introduced in version 23 which was released in June 2011 [37]. Researchers are still contributing to the 'gEDA' project, and it is intended that the work in this thesis will be relevant to the project. The same analysis engine is the basis of many different versions of SPICE including HSPICE and NGSPICE.

A software package called 'RandomSPICE' [14] was employed initially for the randomization process, though an early objective was to adapt this and eventually to develop a new randomization package called "RandomLA" (Randomisation for LSI). A major consideration was the need to perform the simulations and analyses with reasonable computation and to allow the use of parallel computation as provided by MATLAB and CONDOR [39] [40] for circuits of realistic complexity. Hence the need for a variety of complexity reduction techniques and the use of NGSPICE are explored and evaluated in the thesis.

1.4 Research Hypothesis

The research hypothesis is that it is possible to analyse circuit designs for sub-45 nanometre technology by the use of SPICE simulation with statistical models of

variability with computation cost significantly reduced from that required if traditional MC methods are employed.

1.5 Introduction to the Methodology

The tools developed by this project may be described as forming a test-harness which allows SPICE simulation to be used for particular forms of statistical analysis, based on the SPICE simulation of many randomised copies of a circuit. The randomisation must be capable of reflecting both intra-die and inter-die variation of devices and other circuit components such as wires. Intra-die transistor parameter variation can be based on published measurements of devices as provided by manufacturers or researchers. Alternatively, the results of 3D device modelling as carried out by our collaborators in Glasgow University [20] may be used. Applying principal components analysis (PCA) to such sets of representative parameters reduces their dimensionality and provides a convenient way of introducing intra-die correlation. The subsequent computation reduction methods to be investigated in this thesis are the use of behavioural modelling, a technique known as 'Statistical Blockade' based on published ideas of 'extreme value theory' [15], and quasi-random parameter variation with the use of 'low discrepancy sequences'.

1.5.1 Behavioural Modelling

The use of 'behavioural' or 'functional' models of sub-circuits derived by previous statistical analysis of the sub-circuit separately has great potential. For example a behavioral NAND gate model could consist of four 'look-up table' switches each with a statistically variable 'Tau model' of delay [16]. SPICE switches are dependent sources (such as voltage-dependent current sources) whose input-output relationships may be defined by simple look-up tables. These cannot implement delay. However, a simple Tau model which introduces a single RC time-constant can introduce delay, but with a characteristic exponential rising or falling RC wave-shape that may not be appropriate. The true switching behaviour of each gate output, may be therefore modeled by a combination of the switch and the Tau model, with the look-up table

elements optimized (by a simple MATLAB procedure within the harness) to match the RC waveform to the required switching waveform . This simple approach is well suited to the computational methods adopted by SPICE and the demands of simulating asynchronous circuits whose behaviour relies on many 'C-elements' switching at slightly different instants of time. C-elements are widely used asynchronous logic components with a highly non-linear bistable operation. The approach is complementary to the behavioral modelling proposed by Southampton University [7].

1.5.2 Statistical Blockade

Extreme value theory (EVT) offers statistical methods for analysing the behaviour of systems in situations that rarely occur. Such analysis is clearly problematic with traditional MC techniques which require a large sample, therefore extensive computation, for rare events to be observed. The Statistical Blockade (SB) algorithm applies Extreme Value Theory (EVT) to circuit analysis by eliminating randomised parameter vectors that are considered unlikely to produce rare event circuits. In our application, the rare events are the circuit yield failure predictions which are extreme in the sense that they are on the 'tails' of Gaussian-like probability distributions for circuit performances. Since they are designed to be rare, reliable estimates of these failures by conventional MC techniques require very large numbers of randomised input vectors. SB performs 'biased' or 'partial' sampling of the performance distributions which is the basis of EVT. Many input vectors are generated, but only the ones likely to produce 'rare events' are simulated. The process requires a classifier which, in this thesis, will be implemented as a 'least squares' trained linear estimator combined with a threshold comparator. After a period of initial training, the classifier can be trained recursively as the simulation proceeds. The computational complexity involved in introducing the bias, and compensating for it, is much less expensive than performing lots of uninteresting circuit simulations.

1.5.3 Low-Discrepancy Sampling with Quasi MC Methods

The term 'Quasi Monte Carlo' (QMC) describes Monte Carlo methods where the input vectors are not totally random, but are to a degree deterministic in that they conform to 'low-discrepancy sequences' [15][21][36]. A low discrepancy sequence is a sequence of N-dimensional vectors which covers a finite space more uniformly than is achieved by N-dimensional vectors of independent uniformly distributed random elements. It is known that the use of low discrepancy vector sequences can achieve significant speed gains over standard Monte Carlo integration techniques by reducing the number of input vectors needed for a given accuracy [17]. Similar gains are anticipated when QMC is used for statistical circuit simulation.

1.5.4 Originality

All the techniques introduced above are generally well known and have previously been applied in some form to circuit simulation. The originality here is in the way we have applied them and the evaluation and insight that has resulted. Although the principles may be known, the implementation of them has required some innovation such as:

(i) The application of PCA to the introduction of correlation.

(ii) The combination of optimized Tau models and 'look-up table' switches in behavioural modelling.

(iii) The implementation of recursive SB using 'pseudo-inverse' least squares optimization.

(iv) The application of 'SOBOL' LD vector sequences to SPICE simulation (though others have been used) and to SB classifier training and the implementation of recursive SB.

1.5.5 Software Development

This work has resulted in the development of four phases of a MATLAB harness for SPICE called RandomLA (LSI circuit analysis) and some supporting MATLAB programs not yet integrated into RandomLA. The harness works equally well for

HSPICE and NGSPICE and is adaptable to a parallel implementation. The four phases of RandomLA are:

RandomLA-Nonblockade (either MC or QMC)

RandomLA-Training (either MC or QMC)

RandomLA-Evaluation (either MC or QMC)

RandomLA-RecursiveSB (either MC or QMC)

All of these versions implement both MC and QMC (with 'Sobol' sequences) will allow correlation to be introduced, based on PCA, to model intra-die variation. All the results presented were generated by these scripts apart from some of those in Chapter 4 which required the use of RandomSPICE.

1.5.6 RandomSPICE

RandomSPICE is a randomisation package developed by collaborators in the EPSRC nano-CMOS project [7]. Currently it is supplied to collaborators with two transistor models, an nmos and a pmos transistor representing a 35nm technology by Toshiba. The parameters of each model have been randomised to produce 200 copies, where the variations from copy to copy are based on the results of three-dimensional atomistic simulations [18]. The geometrical and quantum physics based simulations were carried out [20] to reflect the statistical nature of 'atomistic' variations due to, for example, the effects of random discrete dopant levels, line edge roughness and oxide thickness variations as would be expected to occur from device to device on a single die. Therefore these are intra-die variations. They are intending to reflect truly what will be observed in real circuits. Intra-die correlation between the parameters of adjacent devices should be taken into account.

1.6 Research Aims and Objectives

The aims of this thesis are to reduce the computational complexity of traditional Monte Carlo (MC) methods for modelling the effects of variability in deep sub-micron CMOS circuits, and to enable a deeper understanding of these effects.

The first objective was the design and implementation of a statistical simulation

method, capable of predicting the effect of parameter variations on the performance and yield of a nano-CMOS circuit, by using traditional MC methods with facilities for including the effect of inter-die and intra-die correlation in the variability. To allow the research to be disseminated in reproducible form, all software was required to be compatible with 'NGSPICE' and the associated GNU public licensed suite of electronic circuit design (EDA) tools. The software was required to be suitable for distributed or parallel computation.

The second objective was to investigate dimension reduction techniques for MC simulation, focusing on the use of Principal Components Analysis (PCA), and the use of behavioural modelling for replacing device level analogue sub-circuits by computational simpler circuit models.

The third objective was to investigate two further computation reduction methods which are a technique known as 'Statistical Blockade' based on published ideas of 'extreme value theory' [15], and the use of Quasi MC techniques based on the use of 'low discrepancy sequences'[98] [99] [123]. It was required to be discovered to what extent computation reduction can be achieved by these two methods both individually and in combination.

All three objectives were approached within the context of the aims stated above, and were designed to achieve the greater understanding and reduced computational complexity required, with illustrations of what is achievable.

1.7 Structure of This Thesis

This thesis contains nine chapters. The first is an introduction to the general research area and the research aims. It defines the problem and some terms. The context, research hypothesis, aims and objectives are stated with a brief introduction to the methodology. Finally, the structure of the following chapters is surveyed.

The second chapter is a background and literature survey covering the sources and classifications of variability. A discussion of the effect of variability on clocked and self-timed circuits is included.

Chapter 3 describes the current state-of-the-art in statistical simulation

techniques for current and next generation integrated circuits. The widely used design tools provided by Synopsis (HSPICE), Cadence (Encounter) and NGSPICE are discussed. Their relevance to IC design and simulation is explained with some examples.

Chapter 4 deals with the use of Monte-Carlo (MC) techniques for the statistical analysis-by-synthesis of nano-scale technology. The existing MC features of HSPICE and the randomization package called RandomSPICE, which is one of the outcomes of the EPSRC Pilot project [7], are outlined and considered for adaptation to the requirements of this PhD project. The reasons for adopting NGSPICE with a new harness called RandomLA, are outlined. Some of the basic features of RandomLA are discussed and the use of traditional MC simulation using RandomSPICE and RandomLA is illustrated with a sample circuit.

Chapter 5 discusses two method of reducing the dimensionality of the input parameter space to achieve computational efficiency in MC simulations. The first of these, principal components analysis (PCA), also provides a convenient way of introducing intra-die correlation between parameters. The second method introduces the use of statistical behavioural circuit blocks (SBCB) which substitute functional but computationally simpler circuit models for device level analogue sub-circuits

Chapter 6 introduces the concept of Extreme Value Theory (EVT) and explores an algorithm known as 'Statistical Blockade' (SB) [28] which applies EVT to statistical circuit analysis by eliminating or 'blocking out' randomised parameter vectors that are classified as being unlikely to produce circuits that fall in the low-probability tails of the distributions of measurements of interest. The potential for using this technique to achieve major computational savings is explored and illustrated by examples.

Chapter 7 investigates the use of Quasi Monte Carlo (QMC) techniques and 'low-discrepancy' sampling to achieve further efficiency improvements, over what was achieved in earlier chapters, with Monte Carlo circuit simulation. The effect of using a 'Sobol' low discrepancy sequence generator to replace the uniformly distributed pseudo-random number generator previously used to produce the required Gaussian variation is discussed and illustrated by example.

Chapter 8 evaluates the results obtained with VLSI SRAM circuits. This chapter considers the significance and reliability of the results obtained, how to decide how many randomized circuits are needed and how best to populate the transistor model sets, taking advantage of QMC and PCA.

Chapter 9 presents conclusions and suggestions for further work in this area. The use of parallel processing for efficiently undertaking the intensive computation required will be discussed, taking into account the intrinsically parallel nature of massive Monte Carlo simulations.

Chapter 2

Nano-CMOS Technology and the Causes and Effects of Variability in Integrated Circuits

2.1 Introduction

The International Technology Roadmap for Semiconductors (ITRS) [8] and Moore's Law [9] demonstrate how progressive scaling of CMOS integrated circuit technology has driven the phenomenal success of the semiconductor industry in delivering larger, faster and cheaper integrated circuits. Scaling is measured in terms of the size of each 'metal–oxide–semiconductor field-effect transistor' (MOSFET) on the integrated circuit; specifically the length of the silicon channel between the source and drain terminals of the MOSFET which would, for example, be 90 nm in '90 nm CMOS technology'. Considering the current status of nano-CMOS technology, integrated circuits with 45nm MOSFETs have been in mass production for some time and circuits with sub-10nm MOSFETs are expected to be available in 2016. This is well ahead of the 2006 version of the ITRS road-map [8] which predicted that 22nm devices would be scheduled for production only in 2018. Further, 4 nm transistors have already been demonstrated experimentally, highlighting silicon's potential for scaling beyond the end of the current ITRS prediction [7].

Size reduction of MOSFETs brings several advantages, such as the ability to pack more and more transistors into a given area of silicon, which results in more functionality per unit area. In fabricating chips of a certain complexity, smaller

integrated circuits are therefore required and more chips per wafer can then be fabricated. Since the cost of wafers remains more or less constant, this reduces the price per chip. It may also be expected that smaller transistors can be made to switch faster.

The main device dimensions of MOSFETs are the transistor length, width, and the oxide thickness. One approach to size reduction is a scaling that requires all device dimensions to reduce proportionally. In older technologies, if each of these dimensions was scaled by a factor of 0.7, the transistor channel resistance would not change, while the gate capacitance would reduce by a factor of 0.7. Hence, the RC time-constant which determines the delay of the transistor would reduce by a factor of 0.7. In more recent technologies, this proportionality relationship does not apply and the effect of scaling is rather more complicated. Some of the complications in scaling state-of -the art MOSFETs arise because of the delay due to interconnections.

Along with the advantages gained by reductions of size, some difficulties also arise. MOSFETs whose sizes are below a few tens of nanometers create operational problems, since they tend to have higher sub-threshold conduction, increased gate-oxide leakage, increased junction leakage, lower output resistance, lower trans-conductance, interconnect capacitance, heat production and process variations.

When integrated circuits are fabricated, the dimensions and characteristics of the MOSFETs will not be exactly as assumed in the design process and there will be variations from device to device arising from many different sources. As MOSFET sizes become smaller, these variations have a more and more significant effect on the overall behaviour and viability of circuits. They may cause a particular design of MOSFET to be unusable, hence new device architectures may have to be devised. Also, the variations in device parameters that will be observed after fabrication must be anticipated by the design process. In large-scale circuits, they are too complex to be considered in a deterministic manner and therefore the variability must be modeled by appropriate statistical processes.

Variability in the characteristics of fabricated devices, and the need to introduce new device architectures, are vital considerations for the current and the next generations of nano-CMOS based integrated circuits. Fundamental changes in the

way these integrated circuits and systems are designed are now necessary. Adapting to new device architectures and the variability of fabricated device characteristics will increase the complexity of integrated circuit design processes. For example, statistical models of the intrinsic parameter variations which cause devices on integrated circuits to behave differently from the manufacturer's specification, and from each other, must be used. Failure to accommodate these manufacturing tolerances will challenge the achievable power efficiency, yield and reliability of digital circuits.

In the EPSRC pilot project: 'Meeting the design challenges of nano-CMOS electronics' [7], the APT (Manchester) group's role is to study new design styles that cope better with device variability and reliability. To be able to devise new design styles, we firstly need to investigate how the parameter variations predicted by device level models will affect the performance, power requirements and area of clocked and asynchronous implementation styles, what this will imply for digital microelectronics design, and how these results will affect future synthesis tools.

2.2 CMOS Technology for Integrated Circuits

2.2.1 MOS Transistors

MOSFET transistors are the main building blocks used to design large scale integrated circuits, both analogue and digital. The traditional metal-oxide-semiconductor (MOS) structure which can fabricate a field effect transistor (FET) consists of a layer of silicon-dioxide sandwiched between a layer of metal on top and the semiconductor substrate below, as illustrated in figure 2.1. The silicon-dioxide acts as an insulator, and only a very thin layer is required, often with the thickness of a few hundred molecules. 'Polysilicon gate' FET's, with highly conductive polycrystalline silicon layers replacing the metal layers, are nowadays used in place of traditional MOSFETs, though they are generally still referred to as MOSFETs. Metal and polysilicon FETs are more correctly referred to as 'insulated gate field effect transistors' (IGFETs).

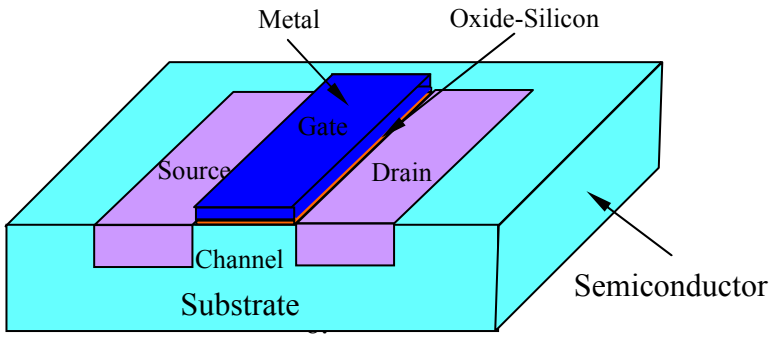


Figure 2.1: MOSFET structure

The transistor has a "source" and "drain consisting of semiconductor material which has been modified by being "doped" with a different type of material than exists in the region under the gate. The metal or conducting polycrystalline material forms the "gate". An NPN or PNP type structure exists between the source and drain regions and electrical current can flow from the source to the drain depending on the degree and polarity of a charge applied to the gate. Figure 2.2 shows two types of MOSFET :

(a) "N-channel" where the source and drain regions have been doped with N type material and the substrate has been doped with P-type material.

(b) "P channel" where the source and drain regions have been doped with P type material and the substrate has been doped with N-type material.

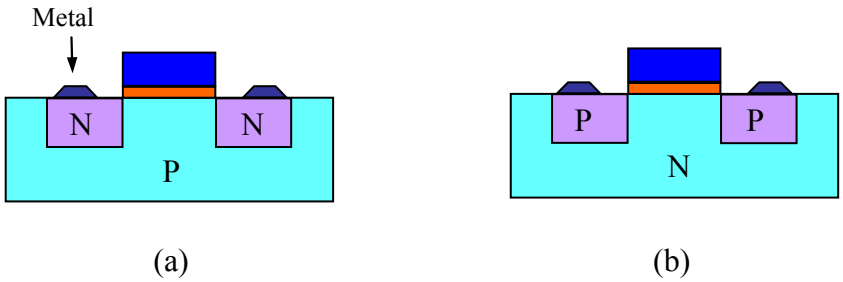


Figure 2.2: Two types of MOSFET.

When either type of MOSFET is operating within a circuit, the source must be connected to a supply of charge carriers which travel towards the drain. For an N channel MOSFET, the source voltage must be negative with respect to the drain voltage to allow charge to flow. For a P-channel device, the source must have the more positive voltage to allow current to flow. The area under the gate is the "channel" through which the current flows.

Either type of MOSFET can be made to act as a switch. Figure 2.3(a) shows an N-channel MOSFET with N-type source and P-type substrate connected to ground and drain connected to a positive voltage V_{DD}. There are two reverse-biased PN junctions between the two N wells and the substrate, therefore no current can flow and the MOSFET is turned off.

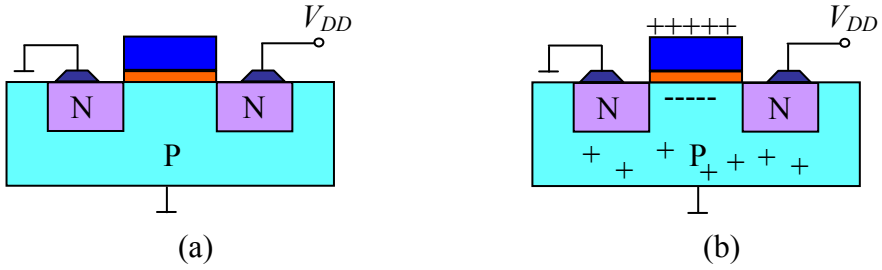
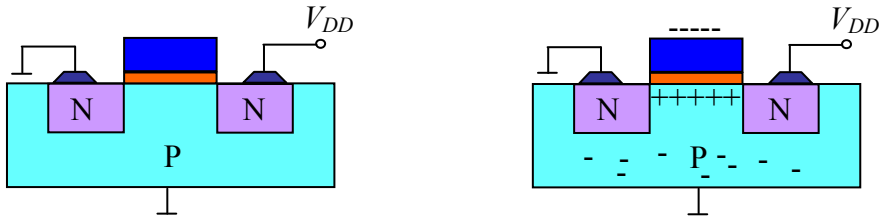


Figure 2.3: Enhancement mode MOSFET (a) turned off, (b) turned on

If a positive charge is applied to the gate as shown in Figure 2.3(b), electrons will be attracted from the substrate into the channel region between source and drain. If the positive charge is enough, sufficient electrons will be attracted into the channel to ensure that there are more electrons than 'holes'. Then the channel will become N-type rather than P-type, current will be allowed to flow from source to drain and the MOSFET will have been turned on. The minimum gate voltage needed to ensure that the gate has sufficient charge to attract enough electrons to allow current to flow is the "threshold voltage" V_{th}. This is an N-channel "enhancement mode" MOSFET because the charge is applied to the gate to enhance the channel conduction. A P-channel enhancement mode MOSFET has P-type source and drain and an n-type substrate. The conduction in the channel is now induced by applying a negative voltage between gate and substrate to create a negative charge and thus attract P-type

charge carriers (holes) into the channel. CMOS technology is based on the use of both N-type and P-type enhancement mode MOSFETs.

"Depletion mode" MOSFETs acting as normally closed switches, are feasible, but not used in CMOS designs. Figure 2.4 shows an N-channel depletion mode MOSFET.



(a) (b)

Figure 2.4: Depletion mode MOSFET: (a) turned off (b) turned on

A thin layer of semiconductor immediately beneath the gate oxide is doped with the same type material as the source and drain. Current can flow across the channel when no charge is applied to the gate, but when a negative charge is applied, the electrons beneath the gate oxide will be repelled leaving no free charge carriers. Therefore conduction will cease and the transistor turns off. P-channel depletion layer MOSFETs can similarly be fabricated. Depletion mode MOSFETs are commonly used as resistors rather than switches. A permanently "on" transistor, has a much higher resistance than doped semiconductor material, and the value of resistance can be determined simply by its dimensions or the number of ions which are implanted in the gate region.

Both enhancement and depletion mode MOSFETs are used in IC design. Conventional NMOS technology uses both enhancement and depletion mode devices; the former as switches and the latter as resistors. CMOS technology uses enhancement mode MOSFETs.

2.2.2 MOS Transistor Models

The design and analysis of ICs requires complex devices such as MOSFETs to be represented by circuits consisting of simpler elements. These are models which are needed for the design of the devices themselves as well as the circuits they populate. The design of devices requires the use of 'process models' which reflect how manufacturing processes such as ion implantation, oxide growth, impurity diffusion, etching and annealing affect their characteristics. Process models translate the device "geometry" into circuit parameters. The effects of readily identified geometrical features and also details such as the doping profiles must be accurately represented.

Transistor models used for circuit design are called "compact models" because ideally they should use as few circuit elements as possible to keep the analyses as simple and computationally efficient as possible.

Enhancement mode MOSFETs can be modeled as simple switches which are on or off, in effect acting as variable resistors controlled by capacitor charges. More sophisticated models can be used, but this simple approach is useful for logic verification and approximate timing simulations.

Many important aspects of the performance of modern integrated circuits are difficult to predict without accurate models of the devices used and their interconnections. Ideally, the models must take into account the circuit layout: length, width, interdigitation, proximity to other devices; transient and DC current-voltage characteristic; parasitic device capacitance, resistance, and inductance latencies and temperature effects. For digital design, large-signal non-linear models are required which may be classified as 'physical', 'empirical' or 'tabular' models.

Physical models are based on device physics and the approximate modelling of physical phenomena within a transistor. Parameters are physical properties such as oxide thicknesses, substrate doping concentrations, carrier mobility, etc. The complexity of modern devices often makes physical models too computationally complex for circuit design purposes.

Empirical models are based on curve fitting to produce functions that recreate known responses to particular stimuli and interpolate this behaviour appropriately for

any given stimuli. An empirical model need have no physical basis and can be considered in purely mathematical terms.

Tabular models use look-up tables containing large numbers of values for common device parameters such as drain current and device parasitics. The use of tabular models can greatly reduce the computational complexity of analysis and simulation software. They can work well in operating conditions whose parameters may be interpolated from entries within the table. However, they tend to be unreliable for operating conditions whose parameters fall outside the table and require extrapolation.

Commercial programs for simulating the behaviour of MOS integrated circuits generally offer a wide range of different transistor models, often with many parameters. The SPICE circuit simulation program [82] is probably the most widely used. The transistor models in SPICE are a hybrid of physical and empirical models. Such models require a specification of how their parameters are to be obtained for real devices, since there is the danger that totally inappropriate parameters can be made to fit measured data for given devices resulting in quite unsuitable models for interpolating or extrapolating the data.

As devices become smaller, new models are needed to accurately represent their behaviour. Simulation packages such as SPICE are continually introducing new device models. A working group called the Compact Model Council [43] has been set up to try to standardize such models across different simulators. This group must consider how the next generation of devices will work by identifying technology trends and motivations. The aim must be to have models available before the devices themselves become available. The BSIM models, developed at U. C. Berkeley already provide such standardized models which include BSIM3, BSIM4, and BSIMSOI.

The BSIM (Berkeley Short-channel IGFET Model) [42] is a family of MOSFET models suitable for integrated circuit design, analysis and simulation. The models represent current flow and capacitance as functions of the control voltages on gates, sources, drains and substrate and other parameters which include channel dimensions and operating temperature. The models are claimed to have features that improve the

convergence rates and accuracy of circuit simulations. They are compact semi-empirical models [47] comprising sets of equations originally derived from physical analysis though subsequently modified empirically to better match available measured data. A 'two-stage' modelling approach is used which first pre-processes the various temperature and device geometry specifications to produce circuits whose elements are suitable for a circuit simulator such as SPICE, and then computes the required component values for such circuits. For each semiconductor manufacturing process, a single geometry-independent parameter set allows the circuit simulator to adapt the model to the dimensions of particular devices. The geometry-independent BSIM parameter sets are therefore functions of the semiconductor processing only and are referred to as "process" models. The actual parameters are extracted using an automated test and data analysis system which provides the means of acquiring large amounts of parameter data as required for statistical modelling of integrated circuit variability. Software which performs cycles of testing and with parameter extraction calculations was developed at U. C. Berkeley [48].

2.2.3 CMOS Logic

N-channel MOSFETs are smaller than P-channel MOSFETs and producing only one type of MOSFET on a silicon substrate is cheaper and technically simpler. NMOS logic uses N-channel MOSFETs exclusively but has the disadvantage of consuming power even when no switching is taking place. In principle, "complementary" MOS (CMOS) logic gates only consume power when switching and have the further advantage over NMOS that both low-to-high and high-to-low output transitions are faster since the load resistors in NMOS logic are replaced by pull-up transistors which have low resistance when switched on. In addition, the output signal swings the full voltage between the low and high rails. This strong, more nearly symmetric response also makes CMOS more resistant to noise. CMOS logic has, since the mid 1980s, displaced NMOS to become the preferred technology for digital ICs.

A CMOS gate has a pull-down circuit of N-MOSFETs for connecting the output

to "0" (GND) and a pull-up circuit of P-MOSFETs for connecting the output to "1" (V_{DD}), as shown in Figure 2.5. One circuit is intended to be ON while the other is OFF. Two or more MOSFETs in parallel are ON if any one of them is ON. Two or more MOSFETs in series are ON only if all of them are ON. CMOS logic gates can be constructed by using appropriate combinations of parallel and series MOSFETs in each circuit. A CMOS inverter, for example, has a 'pull-up' circuit consisting of one P-MOSFET, and a pull-down circuit with one N-MOSFET. The output of a CMOS logic gate can be in four states as summarised in table 2.5. The "1" output level occurs when the pull-up circuit is on with the pull-down off, and vice versa for the "0" output level. When both pull-up and pull-down circuits are OFF, the output level becomes indeterminate or 'floating' with only a very high-impedance connection to the input. This is referred to as the 'floating Z' output state and is used in multiplexers, memory elements, and bus drivers. When both pull-up and pull-down circuits are simultaneously turned ON, an indeterminate level again results but with power being dissipated. This 'crowbarred X' condition is avoided as much as possible in a CMOS gate.

	pull-up OFF	pull-up ON
pull-down OFF	Z	1
pull-down ON	0	Crowbarred (X)

Table 2.1: Output states of CMOS logic gate

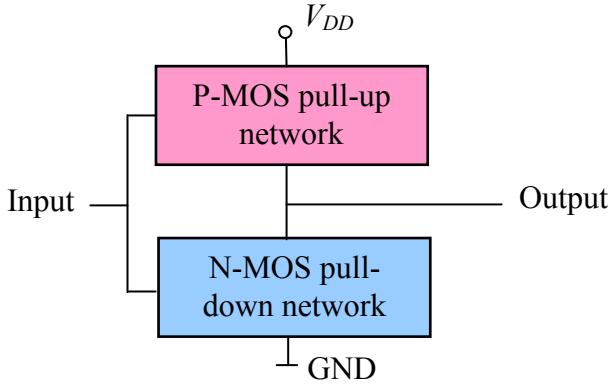


Figure 2.5: General logic gate using CMOS pull-up and pull-down networks

2.2.4 CMOS Circuit Design

Particular logic functions can often be implemented with many different combinations of AND, OR and other gates. When designing ICs, there are many aspects other than the correct logical operation to consider. For example, the fan-in and fan-out of each gate, and the transistor sizes to be used, must be decided. Such decisions affect the speed, power consumption, area and many other potentially important characteristics of the IC being designed.

Logic circuit design tools can make these decisions automatically. Such tools can search through available libraries of logic cells for the best implementation. The resulting circuits are often quite acceptable. However, when there are critical requirements, maybe for low power consumption or high speed, customised circuit design may be needed for the whole IC or for critical portions of it. Customised design effort can also be cost-effective as a means of reducing the surface area required for ICs that are expected to be manufactured in large volumes.

Figure 2.6(a) shows a logic circuit for the Muller C-element. Implementing a non-customised 'gate level' design would require 26 transistors and four stages of gate delay, since the AND and OR gates would be implemented by library cells each consisting of a NAND or NOR gate followed by an inverter. The 'transistor level' design in Figure 2.6(b) requires only 10 transistors and two stages of gate-delay. This example illustrates the advantages to be gained by customised design, at transistor level, based on a direct knowledge of the properties of CMOS technology.

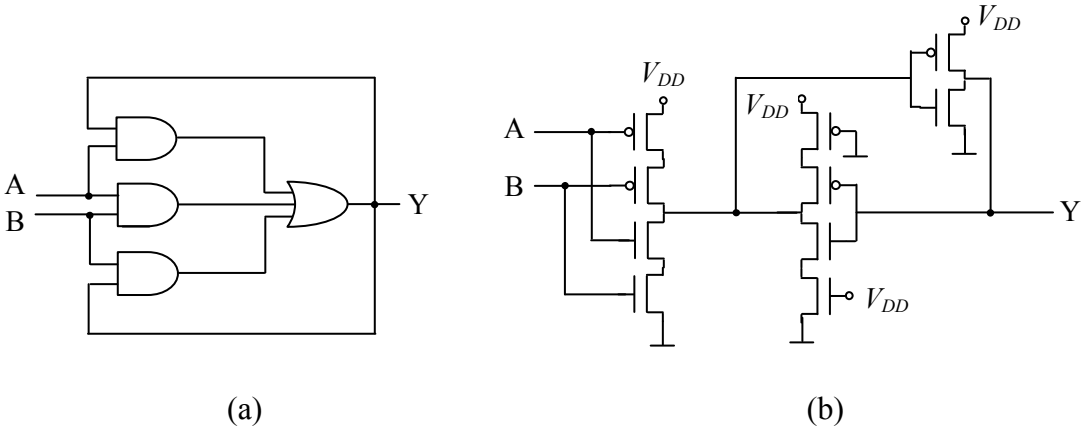


Figure 2.6: Muller C-element: (a) gate-level, (b) transistor-level

Either of the schematics in figure 2.6 may be converted to a 'netlist' description using 'HDL (hardware description language)' terminology which caters for 'gate-level' and 'transistor-level' schematics. A gate-level netlist is a technology-independent structure description, because standard logic gates may be referred to without the need to specify the implementation technology. The electrical characteristics of the gates, speed, fan in, fan out etc. are not yet part of the description. At transistor level, a circuit description will include this type of information and specify the size of the transistors and the levels of stray capacitance. However, the Verilog HDL language was designed primarily as a gate-level language and is not well suited to transistor level descriptions.

The circuit simulator SPICE uses another common structural language whereby internal models represent the electrical characteristics of the MOS devices. SPICE calculates appropriate values of, for example parasitic capacitance inherent in the MOS transistor, using parameters such as device dimensions that are specified in the netlist. Capacitance, resistance and other phenomena can be introduced independently from the transistor models by including appropriate statements in the SPICE netlist. Thus additional routing capacitance and resistance can be included to accurately model the physical characteristics of the circuitry for each gate and the interconnections between gates. The SPICE netlist has all the information necessary to fully characterize a transistor level circuit description, in terms of its speed, power, and connectivity.

2.2.5 CMOS Technologies

Complementary MOS (CMOS) gates employ both P and N channel MOSFETs to allow a signal which turns *on* one transistor to turn off another. This eliminates the need for pull-up resistors and the power they would dissipate. Figure 2.7 shows a CMOS inverter and its switch equivalent.

Figure 2.7: A CMOS inverter and its switch equivalent

The substrate for the N-channel device is connected to ground, while that for the P-channel device is connected to the positive voltage supply. Figure 2.8 shows the arrangement of channels for an IC implementation of the inverter.

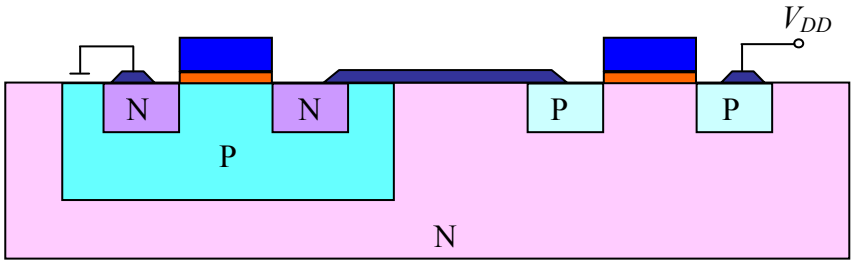


Figure 2.8: CMOS inverter in cross-section

In principle CMOS gates consume no power when not changing state since there is no resistive path to ground. Outputs can be made symmetrical in the way they switch from 0 to 1 and vice-versa, by making the pull-down and pull-up resistances of the N-channel and P-channel transistors equal. This will equalize the delays for each direction. CMOS technology is more complex than NMOS since it requires two different types of transistor to be fabricated on a single substrate.

2.2.6 MOSFET Scaling and the Adaptations on Design

The ITRS Roadmap [8] forecasts a major new technology generation every three years. Each new generation is expected to double the number of transistors per unit

area and increase their operation speed by a factor S equal to the square root of two. Constant field scaling by the factor S multiplies channel length, width, oxide thickness, supply voltage and all threshold voltages by $1/S$ with substrate doping level increased by the factor S. Scaling dimensions, voltages and dopant levels in this way keeps the electric field strength unaffected and the gate capacitance per unit width of channel also remains approximately constant since channel length and oxide thickness scale in proportion. If only the gate length is scaled, leaving other dimensions, voltages, and doping levels unchanged, this is 'lateral scaling' or 'gate shrink' which can be easily applied to existing masks. In theory this results in a decrease in gate delay by a factor proportional to S squared. In practice, the decrease factor will be approximately proportional to S, rather than S squared, because 'velocity saturation' will cause there to be a constant relationship between channel current and resistance.

The improved device density and IC performance improvements over the past 40 years have been remarkable in providing ever-increasing functionality and speed. They have also led to reductions in manufacturing costs since the silicon die area for given functionality is reduced. Therefore, more dies can be fabricated on a silicon wafer of fixed size and cost of manufacture. Smaller die sizes also lead to higher yields since a given density of imperfections will affect fewer dies [49].

As the trends in scaling continue, and MOSFET gate lengths reduce to below 35nm with gate oxide thickness reduced to the order of 1nm, physical limitations in allowable power density and the increasing significance of off-state leakage current make state-of-the art IC design an increasingly challenging task. Innovations in device structures and materials are now required. Limitations in the fabricated materials are imposing new and increasingly difficult problems. For example, since carrier mobilities are affected by the increased vertical electric fields that occur [50], mobility enhancement techniques such as strained-Si [51], [52] and high-mobility channel materials [53] are increasingly being called for. Also, gate tunnelling leakage becomes significant for oxide thickness below 1 nm [54], hence the need to replace oxide/oxy-nitride dielectric layers with a high-permittivity or 'high-K' gate dielectrics [55]. New device structures include designs based on 'silicon-on-

insulator' (SOI) technology. This was developed to offer an enhanced trade-off between power and performance by reducing source/drain junction capacitance, and achieving superior control of 'short channel effects' (SCE). Therefore, SOI is beneficial as a low-power technology for ICs with stringent leakage specifications. The double-gate FinFET structure [56], [30] has improved electrostatic properties, and if the fin gate is made wider and shorter, a tri-gate SOI MOSFET structure is formed, which has advantages for manufacturability. These advanced device structures have been developed to enable continued transistor scaling beyond current limits.

In addition to the design problems surveyed above, further problems arise due to the increased variability in device performance that is inevitable with the parameter scaling that occurs with each new technology generation. This variability critically challenges the viability of future technology development, manufacturing, and design [57]. Therefore, besides the advances in transistor architectures and process control briefly mentioned above, improved IC analysis, simulation and design techniques are needed to anticipate the effects of variability and eliminate or reduce the failures it may cause in current technology. Further, designers will need to be able to predict the effect of future feature size scaling on chip performance to plan their designs in such a way that future products may be expected to scale gracefully.

2.2.7 Synchronous and Asynchronous ('Self-Timed') Circuits

2.2.7.1 Introduction

Technologies at the nano-scale level, including nano-CMOS, are facing great challenges due to the effect of parameter variations. Among these are the efficient timing control of the required sequences of transitions, and adapting to the impossibility of building global clock networks on highly complex chips [4]. Because of these difficulties, asynchronous (clock-less or 'self timed') logic is commonly regarded as an ideal and perhaps unavoidable choice for digital circuits in the technology of nano-CMOS [4]. The timing issues are discussed in this section, firstly for traditional synchronous circuits and then for asynchronous circuits.

2.2.7.2 Synchronous Circuits

In synchronous circuits, changes in the logical levels of storage elements are intended to be simultaneous with the level change of a single 'clock' waveform. The changes cannot occur instantaneously, but they must be complete before the next synchronizing level change occurs. The speed of the changes determines the minimum time that must elapse between level changes in the clock waveform. This sets the maximum allowed clock speed for the synchronous circuit. Much design effort is needed to design the efficient distribution of the required clock signal from a common entry point to all parts of an integrated circuit. The characteristics of clock signals and the electrical connections used in their distribution have special requirements since they have high fan-out and must operate at the highest speeds of any signal within the entire circuit. The clock waveforms must be particularly clean and sharp to provide accurate timing. These attributes are especially sensitive to the effects of technology scaling since the resistance of long interconnections becomes greater and more variable as line dimensions are decreased. These effects, combined with increasing variability of capacitance, result in uncertainty in the exact arrival times and definition of clock waveform events which can severely limit the maximum performance of the entire circuit and create hazards and race conditions which cause incorrect logical behaviour. The transitions of synchronous circuits are controlled by the careful insertion of pipeline registers to ensure that critical timing requirements are satisfied and that no race conditions exist.

The clock distribution connections often dissipate a significant proportion of the power consumed by an IC, and significant power can be wasted by clock connections to parts of the circuit not being used. A technique called 'clock gating' can turn off connections when they are not needed to achieve significant power savings.

Synchronous circuits have sub-circuits of three types: memory storage elements, combinational logic elements, and clocking distribution networks with associated circuitry. Interrelationships among these three types of sub-circuit are critical to achieving acceptable performance and reliability. While the existence of a single universal clock waveform may be assumed at the design state of synchronous ICs, in

practice a number of related but different clock signals will be required and generated by a structure or network of clock buffers. Engineering design effort is needed to design clock-gating circuitry and to minimize 'skew' between the different clock signals. Satisfying timing specifications in circuit technologies dominated by highly variable wire delays is not an easy task. Current commercial CAD tools employ iterative buffer-insertion-and-resynthesis procedures, but these may not converge and often must be based on questionable assumptions about the characteristics of the connections.

2.2.7.3 Asynchronous circuits

Instead of synchronising all transitions to a common clock waveform, asynchronous circuits use 'handshaking' between elements to achieve the necessary synchronization, communication, and sequencing of operations. Register transitions are only initiated locally and only when needed. The stored contents of registers are considered to be 'tokens' whose values may be changed by combinational circuits connecting the outputs of registers to the inputs of other registers. Combinatorial circuit connections are transparent to the handshaking between registers. An asynchronous circuit is a static data-flow structure [59]. The basic principle is that a given register may store a new value from a data token supplied to it as input, obtained from the output from another register referred to as 'the predecessor register'. But it can do so only if a further register referred to as the 'successor register' has accepted and stored the data token that the given register was previously holding. The states of the predecessor and successor registers are indicated to the current register by 'request' and 'acknowledge' signals respectively. The Muller C-element, a version of which is shown in Figure 2.6, is the basic unit for the implementation of handshaking. By this 'hand-shaking' mechanism, data is transferred from one register to another via combinational circuits. The mechanism reflects the register transfer level (RTL) description of logic circuits that separates the structure and function of a circuit from its implementation. The handshaking between the registers controls the flow of tokens with combinational circuit blocks

transparent to this handshaking. The term "function block" is used to denote such a combinational circuit.

Whereas synchronous circuits are controlled by a single universal clock waveform, asynchronous circuits are controlled by a large number of locally derived control pulses that can occur at any time. The local handshaking ensures that control pulses are generated where and when they are needed. The fact that a single strongly periodic universal clock waveform is eliminated and replaced by many less periodic unsynchronised control signals results in less strongly periodic electromagnetic emission and a smoother supply current without the large supply current spikes that characterize synchronous circuits.

Asynchronous circuits offer low power consumption [58] [60] and high operating speeds [61] since the operating speed is determined by actual local latencies rather than global worst-case latency. Also, less periodic emission of electro-magnetic radiation is to be expected [62]. There are no clock distribution and clock skew problems [6] and timing in asynchronous circuits is insensitive to variability in circuit and wire delay. Therefore, asynchronous circuits may be expected to be robust to variations in supply voltage, temperature, and fabrication process parameters [63].

2.3 Nano-CMOS Variability and Effects on Integrated Circuits

2.3.1 Introduction

When integrated circuits are fabricated, the dimensions and characteristics of the MOSFETs will not be exactly as assumed in the design process and there will be variations from device to device arising from many different causes. The ability of a device or circuit to vary from copy to copy in a particular way due to a particular cause is referred to as 'variability'. Before nano-scale technology, the circuit-to-circuit variability came mainly from imperfect control of the fabrication processes which caused the parameters of each device to vary from wafer to wafer, and from die to die within each wafer. As technology has reached the nano-scale region, device-to-device or intra-die variability is becoming a much more important

consideration [3]. Dimensions are approaching atomic scales and intrinsic atomic scale variations such as line edge roughness and dopant granularity are becoming the most significant sources of variation [2]. These 'atomistic' variabilities must be considered random in nature, though there may be correlation between their effects on adjacent devices. They must be understood in statistical terms and taken into account at the design stage.

It is necessary to understand the sources of the variations and to be able to analyze the impact they have on performance. The information thus gained can then be taken into account during the design process. Based on their causes, the variations are characterized as: 'process variations', 'environmental variations', 'modeling variations' and 'variations due to other sources'. Any of these variations may be: inter-die, intra-die variations [3] or a combination of both. Intra-die variability causes nominally identical devices within a given circuit to have different characteristics, and inter-die variability causes circuit to circuit variability due to nominally identical devices on different dies having different characteristics. Analysing the impact of intra-die and inter-die variability on the performance of individual circuits and the resulting variability of this performance, and ultimately the 'yield', is the main way of judging how well a particular design has been optimized. In nano-CMOS technology, 'atomistic' variabilities, which are 'intrinsic' in the sense that they cannot be eliminated by improved circuit design or manufacturing tools [83], are the most significant source of both intra-die and inter-die variation. Such variabilities include spatially correlated and uncorrelated variations as will be discussed in the following sections.

2.3.2 Sources of Variation

The four causes of intra-die and inter-die variation that were mentioned in the previous section are now defined.

Process variations are the fluctuations in the physical characteristics of devices, such as their geometrical features and distribution of dopant levels, often referred to as 'process parameters', that are caused by inaccuracies and limitations of the

fabrication process [32]. These variations translate to variations in the electrical parameters of the devices. They result from a wide range of imperfections that can occur during the fabrication process.

Environmental variations caused by changes in its immediate surroundings or activity can affect the operation of a fabricated IC. Variations in temperature, power supply voltages and switching activity, for example, can affect the circuit's operation. Increased temperature can result in performance degradation for both devices and interconnects. Leakage currents increase strongly with increases in temperature and the power dissipation caused by increasing leakage currents can further increase the temperature of an IC. This positive feedback mechanism can cause thermal run-away, where the currents and temperatures in a circuit continue to rise until failure occurs. Current leakage and temperature analysis must be performed to make sure that such thermal run-away will not occur during normal operation [3]. A reduced power supply voltage lowers the effective fan-out of devices thus risking malfunction of the circuit. Switching activity changes many properties of an IC from its idle state, for example its temperature, power supply integrity and degree of crosstalk due to substrate noise coupling.

'Modeling variations' arise from inaccuracies in models which do not perfectly reflect the characteristics and switching behaviour of the devices in question. Improving the accuracy of models generally results in greater complexity and higher computational requirements. Therefore some compromise is generally called for when performing analysis and simulation of complex circuits.

Other sources of variations include physical effects that cause temporary or permanent changes in process parameters. These effects include 'hot electrons', 'negative bias temperature instability' (NBTI) and 'electro-migration'. Hot electron and NBTI effects introduce permanent device degradation that increases with time causing threshold voltages to change. Electro-migration causes the resistances of on-chip connections to increase by reducing their physical widths. Increases in propagation delay will then occur owing to increased RC values.

2.3.3 Classification of Variation

It is now widely realized that inter-die and intra-die variations must be considered separately [3]. Intra-die variation causes nominally identical parameters to vary from device to device within each fabricated copy of a circuit. With the statistical modelling of atomistic variation, a different set of parameters are, in principle, required for each device on a die. Therefore, very many different random variables are needed. The sets of variables will clearly have dependencies within them, but it may be considered reasonable to assume that intra-die variations are uncorrelated from device to device. This assumption simplifies the analysis and simulation of circuits and may be justified in view of the nature and cause of atomistic variation. However, it is clear that there will also be correlation from one set of parameters to another especially when the devices in question are close together on the chip. The degree of this correlation, the necessity and feasibility of taking it into account and viable methods of estimating, measuring and modelling this intra-die correlation are currently open research questions. The modelling of 'within die' variations is now receiving much interest in the research literature [84] and the hypothesis that such modelling is necessary for accurate analysis and simulation is certainly being considered. The correlation that exists in intra-die variations may be due to both systematic properties of the technology and the characteristics of the fabrication techniques that are used; for example any recurring imperfections. Despite the increased complexity of analysis and simulation problems incurred by attempting to model intra-die variations with correlation, models have already been proposed for introducing correlation due to proximity-effects [3].

Inter-die variations in the parameters of a given device occur from die-to-die, wafer-to-wafer and lot-to-lot. Clearly both intra-die and inter-die variations will occur in practice, but it is sometimes reasonable to disregard device-to-device variability on each chip to concentrate on just inter-die variations which are considered more substantial. These variations are generally considered independent and modelled by a small number of random variables which represent a deviation from the nominal values of circuit parameters. Typical of inter-die variations that

may be modeled are on-chip consistent differences in gate-lengths that occur owing to fluctuations in exposure time during fabrication. Also, on-chip consistent metal thickness variations between different metal layers may similarly be modelled. Where the nominal value of a device parameter is denoted by P_0, its inter-die variation may be modeled by defining the corresponding parameter for circuit i as:

$$P(i) = P_0 + \Delta P(i) \tag{2.1}$$

where $\Delta P(i)$ is the ith sample of a zero mean random variable of appropriate statistical properties. The random variables $\Delta P(i)$ are often assumed to have an approximately Gaussian probability density function (pdf) with a given variance. To introduce both intra-die and inter-die variability, define parameter $P(i,x,y)$ for a device at co-ordinates (x,y) on chip i as:

$$P(i,x,y) = P_0 + \Delta P(i) + \Delta Q(i,x,y) \tag{2.2}$$

where $\Delta Q(i,x,y)$ introduces random variation that is dependent on (x,y). For each *i, x, y* express:

$$\Delta Q(i,x,y) = \Delta Q_s(i,x,y) + \Delta R(i,x,y) \tag{2.3}$$

with a spatially correlated component $\Delta Q_s(i,x,y)$ and a statistically independent component $\Delta R(i,x,y)$, the latter being just a sample of an independent random variable with no correlation with other devices. A viable approach to the definition of $\Delta Q_s(i,x,y)$, recommended by A. Srivastava, D. Sylvester and D. Blaauw [3] is to divide the surface area of each circuit i into regions for which $\Delta Q_s(i,x,y)$ may be assumed identical (the same random sample used) for all (x,y). An alternative is to define a number of 'anchor points' on the chip's surface and then to define each $\Delta Q_s(i,x,y)$ according to the distance between (x,y) and all or some anchor points. The closer (x,y) is to an anchor point (x_a,y_a) say, the stronger the correlation will be made to exist between $\Delta Q_s(i,x,y)$, and the random value of $\Delta Q_s(i,x_a,y_a)$ at the anchor point.

Intra-die variations can be combinations of wafer-to-wafer variations, layout dependent variations, and statistically independent variations. Layout dependent variations can arise from lithographic and etching fabrication techniques including chemical mechanical polishing (CMP) and optical proximity correction (OPC). Atomistic random dopant variation is considered to produce statistically independent parameter variations. According to the reference [135], line-edge roughness is

statistically independent even for devices close to each other. Some phenomena such as dose concentration vary slowly across locations on the die, and thus exhibit spatial intra-die correlation. Intra-die variation is often assumed to have both a contribution with device-to-device correlation and a statistically independent contribution. More detail of such variability will be given in Section 2.3.4. Intra-die random variations can result from many other sources and strongly influence threshold voltages (Vth) and leakage power. Increased Vth variability and lower Vth values can introduce higher current leakage and increase the likelihood of functional failure.

2.3.4 Intrinsic MOSFET Variability

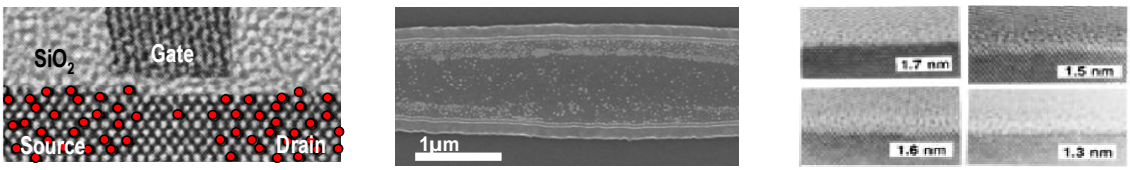
The variability in MOSFET characteristics that occurs due to imperfections of the manufacturing process can be assumed to lie within known constraints and is relatively easy to model in analysis, simulation and design processes. However, with sub-45nm technologies, atomistic effects are becoming increasingly significant. The impact of such effects was small enough to be neglected when the effects of manufacturing imperfections were vastly greater. However, in future devices, they will become a major consideration. Novel design techniques will emerge for reducing the loss of precision that occurs in the manufacturing process. However, the fundamental limitations cannot be overcome, and their importance will increase as device sizes continue to reduce [2], [66].

Atomistic variability is ‘intrinsic’ to sub-45 nm technology and will always be present as a source of intra-die variations regardless of how good the circuit design or manufacturing tools can be made. The physical phenomena that cause this variability can only be modeled in statistical terms. Three forms of atomistic variability are ‘Random Discrete Dopant Fluctuations’, ‘Gate Line Edge Roughness’, and ‘Gate Oxide Thickness Variation’. Figure 2.9 illustrates the physical causes of these three forms of variability.

2.3.4.1 Random Discrete Dopant Fluctuations

Random discrete dopant (RDD) fluctuations result from modern fabrication

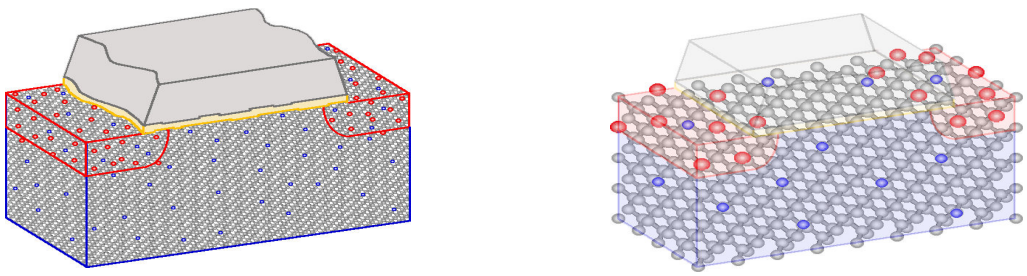
processes which implant relatively small numbers of dopant atoms into silicon at very high energies. Collisions and scattering occur until thermal annealing allows



(a) (b) (c)

Figure 2.9: Intrinsic variability
(a) Random Dopant Fluctuations [64]; (b) Gate Line Edge Roughness [109];
(c) Gate Oxide Thickness Variation [44].

implanted atoms to replace silicon atoms within the crystal lattice, at the same time diffusing their positions still further. It is impossible to precisely control the distribution and positioning of the dopant atoms and every device will have a different distribution of dopants. Therefore, threshold voltages determined by the concentration of dopant atoms will vary from device to device. This variation would have been less with older technologies because of the averaging effect of having a very large number of dopant atoms. Because of the small numbers of atoms involved, increasing or decreasing the number of atoms by an integer number cannot be considered a continuous process and is termed discrete. Figure 2.10 illustrates the atomic structures of simulated 22nm and 4nm devices with typical random distributions of dopant atoms. The 22nm device is indicative of devices that will soon be available, and the 4nm device is close to the limit down-scaling that is ever likely to be possible in silicon.



(a) (b)

Figure 2.10: The atomistic structures of stylized transistors illustrating random discrete dopant placement [67]: (a) 22nm Device; (b) 4nm Device

Clearly, the number of dopant atoms in the channel regions of a device decreases as the critical dimensions are scaled down [20]. As the number of dopant atoms becomes smaller, a modest variation in this number causes an increasingly large variation in device performance since it is ratios of the number of carriers in different channels that determine this performance. The locations of these dopant atoms within the silicon lattice structure also affect the threshold voltages of devices, and there will be increased variability in the distribution patterns from device to device when there are fewer dopant atoms [75]. The effects of uneven local dopant distribution are not averaged out over large regions as with larger devices and therefore threshold voltages will become more variable because the uneven-ness will cause certain regions to become active before others [68]. The variation caused by random dopant variability will be modeled by the uncorrelated intra-die statistical variable $\Delta R(i,x,y)$ in equation 2.3 when the parameter P in equation 2.1 is a threshold voltage.

2.3.4.2 Line Edge Roughness

Line Edge Roughness (LER) occurs at the junctions of channels with other materials causing random deviations from ideal straight-line boundaries. Variable material characteristics and tools used in the lithography processes are causes of LER [69], [70]. LER occurs in photo-resist (PR) processing depending on the PR type, thickness, substrate reflectivity, image contrast and processing conditions. In nano-scale poly-silicon gate etching, the degree of LER is strongly dependent on the poly-silicon grain size and the doping. In fabrication, silicon wafers are spin-coated with PR material, exposed to UV light through a photo-mask and then heated to 'cure' the photo-resist. Wafers are then immersed in a liquid developer to dissolve either the exposed or the unexposed areas of photo-resist material. LER occurs at the boundaries of masked areas because larger aggregates in the PR tend to dissolve more slowly than smaller grains of material. The degree of LER that occurs is found to be closely related to the grain size and molecular weight of the photo-resist material [68].

In older technologies, the dimensions of MOSFET channels were orders of magnitude larger than the roughness, therefore the effects of LER were not

significant. In sub-45 nanometre technologies, line-edge variations remain the same while other dimensions reduce and therefore the variations become much larger in proportion. LER introduces a peak-to-peak variation of about 5 nm in most types of lithography used today [71], [72]. If this level of roughness cannot be reduced, LER will seriously affect ICs with sub-32 nm channel-length devices and will become the main cause of variability in 18nm channel-lengths devices, instead of random dopant variability [73].

LER causes variations in off-state leakage current flow, effective channel lengths [74] and threshold voltages [75]. When considering a parameter P defined by equation 2.1, variations caused by LER may be modeled by the non-correlated intra-die statistical variable $\Delta R(i,x,y)$ in equation 2.3.

2.3.4.3 Gate Oxide Thickness Variation

Gate Oxide Thickness Variations (OTV) are vertical deviations in the depth of the silicon dioxide layer below the gate of a MOSFET. Intrinsic atomic scale roughness in the silicon-to-silicon dioxide and gate-to-silicon oxide junctions will introduce device parameter variability, especially when the thickness of the silicon oxide layer is equivalent to only a few atoms [76]. The random OTV on a sub-45nm technology chip will make each MOSFET different in respect to the surface roughness limited mobility, gate tunnelling current [77] [78] [79], and real [80] or apparent threshold voltage [81]. For circuits with device dimensions below 30 nm, the threshold voltage variability due to OTV will become comparable to the variability due to random discrete dopants [76]. OTV may be modeled by the correlated intra-die statistical variable $\Delta Q_s(i,x,y)$ in equation 2.3.

2.3.5 Effects of Variability on Performance

According to A. Srivastava *et al* [3], probably the most critical form of variability is gate-length variability, although this is disputed by some experts who argue that dopant variability is more critical [136]. It occurs 'inter-die' due to variation in exposure time and 'intra-die' because of other lithography effects [3]. Intra-die

variations in gate-length will have both spatially correlated and uncorrelated contributions.

Device threshold voltages will be dependent on a number of process parameters including gate-lengths and channel doping concentration. Gate-length variation will cause comparable amounts of independent inter-die variability and spatially correlated intra-die variability, whereas channel random doping variation tends to cause uncorrelated random intra-die variability.

The Device Modelling Group at Glasgow University has observed the effect of intrinsic variability on important device parameters for 35nm simulated devices. Their results are summarised by Table 2.2, which illustrates the standard deviations of variations of the threshold voltages of 35 × 35 nm MOSFETs that are introduced by single and combined sources of intrinsic parameter variation [65]. The most significant contributor to the variability of *Vth* at this channel length appears to be caused by RDD. The contribution of LER to *Vth* variability is slightly less than that of RDD, and introduces some spatial correlation. The value of *Vth* used in circuit simulation should ideally reflect spatially correlated and uncorrelated variation as will be discussed in more detail in Chapter 4.

Fluctuation	*Vth*	σ*Vth*
Random Discrete Dopant	133 mV	33.2 mV
Line Edge Roughness	126 mV	19.0 mV
Oxide Thickness Variation	122 mV	1.8 mV
RDD & LER	126 mV	38.7 mV
LER & OTV	123 mV	33.9 mV
LER&OTV	113 mV	22.8 mV

Table 2.2: Threshold voltage variability caused by single and combined intrinsic parameter fluctuations in a 35 × 35 nm atomistic-simulated MOSFET [65]

Variability of device performance will clearly affect the performance of integrated circuits and will reduce parametric yield thus increasing the cost of

manufacture and ultimately reducing the benefits of scaling. Parametric yield is the percentage of manufactured working samples that meet the required specification, whereas manufacturing yield is the percentage of working samples from a manufacturing process that do not necessarily meet the required specification in all its aspects. It was observed [3] that for a particular circuit manufactured in an 180nm technology, device variability caused the chip leakage current to vary by a factor of about 20, and that the usable clock frequency varied by about 30% from chip to chip. These large variations caused a large fraction of the circuits to fail to meet power and timing constraints, which substantially decreased the parametric yields. Variability also affects power dissipation, since the designer must make that the nominal values of threshold voltages are high enough to ensure that off-state leakage currents do not reduce noise margins beyond safe limits. Power supply scaling must be done in such a way as to guarantee acceptable performance.

2.4 Conclusions

The ever-reducing dimensions of nano-CMOS technology mean that statistically based variability analysis will have an increasingly important role in enabling successful circuits to be designed and optimized. The means of analysing the effect of intra-die variability is needed, and can be provided through the use of SPICE simulations of randomised versions of a circuit. Asynchronous or self timed circuits are known to offer advantages over the more widely used synchronous circuits in very small scale technologies.

Chapter 3

Analysis of the Effect of Variability on Integrated Circuits

3.1 Introduction

The differences that exist between the ideal parameters of an integrated circuit and the parameters that are observed when the circuit is fabricated many times are referred to as variability. The words ‘variation’ and ‘variability’ often imply time dependent change, but in this context, the primary differences referred to do not change with time and occur from component to component on a circuit and from realization to realization over a batch of manufactured circuits. Anticipating the impact of such variability on the performance of integrated circuits is always a critical consideration during design procedures.

3.2 Effects of Variability

Variations in the properties of the material and inaccuracies in the manufacturing processes must be expected to produce circuit components whose characteristics will be different from what was specified. The differences between the parameters of each component and their specified target values will vary from component to component within a manufactured circuit and from realization to realization within a batch of nominally identical copies of the circuit. If the parameters of a set of nominally

identical components on a batch of nominally identical copies of an integrated circuit could be measured and analysed as a set of values, differences in these values may, to a degree, be explained and modeled according to known deterministic phenomena; for example die-to-die variations in physical properties such as dopant levels. However, some aspects of the variation from component to component will likely not be treatable as deterministic either because a deterministic model would be too complicated or because there are effects that can only be considered statistically. Statistical models of the variability will consider each parameter to be a sample of a random variable whose statistical properties (mean, variance, probability density function, correlation properties, etc.) are known or assumed. Parameters are assumed random but related statistically to their nominal target values in ways that may be considered totally independent, or may be affected by other considerations, such as proximity.

3.2.1 Modelling Variability

A circuit parameter can be modelled as the sum of a nominal value and a random variable. The nominal value is set by the circuit designer and is considered to be a deterministic quantity. The random variable represents the statistical variation about the nominal value. The nominal value is often referred to as the designable parameter and the random variable as a 'noise' parameter. The channel dimensions (lengths and widths) of MOS devices, resistor and capacitor values, are designable parameters. Noise parameters for CMOS circuits are uncontrollable variations in the gate-oxide thickness, threshold voltage and channel dimensions of MOS devices.

3.2.2 Simulating Variability

Due to the differences in the characteristics of manufactured components from the designed versions, the overall performance of a batch of integrated circuits will vary from copy to copy. Some critical aspect (such as gain or delay) may vary to such an extent that some copies must be rejected as the critical parameter falls outside a specification. The circuit design should therefore build in a tolerance for anticipated

parameter variation. To be able to attempt this task, the designer must have some idea what degree of variation is to be expected, and its likely effect on the circuit.

Predicting the effect of anticipated forms and degrees of variation is the role of simulation. The expected 'yield', i.e. the percentage of successfully functioning manufactured circuits can thus be estimated for a given design before any circuits are actually fabricated. If the expected yield is unsatisfactory, the design can be modified. Also, the designer can discover the degree to which the complexity and cost of an over-specified design can be relaxed without compromising the yield at all or too severely. The more tolerance there is, the more costly will be the design. Hence designs must be optimized with respect to the conflicting demands of minimal complexity and maximum yield.

3.2.3 'Worst case' Analysis of Variability

The complexity of combining yield estimation with an iterative design process can be computationally prohibitive even for quite modest integrated circuits. Traditionally, circuit designs have been verified by modelling the 'worst-case' conditions of the variable parameters [85], [86]. A circuit designed to work under these worst-case conditions, will be expected to achieve a high yield when manufactured. Worst-case analysis must determine the values of the parameters in these worst-case conditions and then carry out an analysis, normally by simulation, to estimate the worst-case circuit behaviour. Worst-case design is efficient in terms of computation and designer effort, and is the most widely-used approach currently for circuit design based on statistical analysis. However, it is well known to be too pessimistic in its prediction of likely failures, and therefore leads to extremely conservative and costly designs. Unnecessary design effort may be caused by simulation results that are too pessimistic. Optimizations that simply ensure that a deterministic estimation of the performances of 'worst case' versions of a circuit meet a given specification, without considering the statistical nature of the expected variation, are likely not to be efficient. In sub-micron (e.g. sub-45 nm) technology, for which there are much higher degrees of variability, such approaches are not even

likely to be feasible. Traditional variability analysis, based on 'worst case' corner-models [3] and guard-bands for parameter variations, is thus likely to be unsuitable for estimating the effects of variability as required for the design of circuits in this latest technology.

3.3 Worst Case Analysis in Practice

Worst-case circuit simulation can verify that circuit performances are acceptable under worst-case conditions. The reliability of worst-case design optimization methods, as described above, depends on the accuracy of the worst-case analysis procedure. It must be ensured that the worst-case analysis is based on realistic estimates of the worst-case sets of parameter values and produces the worst-case performance values. Most common is the "corners" or "one-at-a-time" technique where the value of each parameter is chosen independently, typically ±2 or ±3 standard deviations from the nominal value, assuming a Gaussian distribution about this nominal value. This technique ignores any correlation that may be expected among the parameter variations. The setting of all parameters to their worst case values produces simulation results with measured parameters that lie in the tails of their joint probability densities. These measurements will be extremely pessimistic [87] as estimates of what will occur in the majority of circuit copies.

The worst case analysis may be carried out in many different ways and at different levels. One approach is to operate at process level [86] by introducing variability into process parameters, such as device dimensions. The behaviour of each resulting component is then modelled to allow circuit simulation to be carried out to estimate the required worst-case performance. An alternative approach is to operate at device level (or transistor level) by modelling the ideal components (e.g. the transistors) and then introducing variability into the parameters that characterize the device models. The application of the latter method to VLSI design can be difficult due to the high dimensionality of the device parameter space and the consequent high cost of the simulation. Muller [304] proposed a means of limiting the computational complexity by randomizing only those parameters which are

capable of causing the most serious variations in the circuit performance with small deviations from their nominal (ideal) values. The circuit performance measurements are assumed to be linearly dependent on the deviation and thus a 'gradient' vector of sensitivities to variation around the nominal parameter vector may be computed. A difficulty with this linearised model of sensitivities of circuit measurements to parameter variations is that the coefficients may not be accurate for parameters not close to the nominal values, which is where the worst-case conditions are likely to occur. Also, any interaction between the degree of variation and the nominal values of the designable parameters is not considered. The possibility of variation in different parameters being correlated to some extent is disregarded also.

3.3.1 Illustration of Concept of Worst Case Analysis

As an illustration of the concept of worst case analysis consider first the design of a circuit which has only one type of transistor with one parameter whose value from die to die is assumed to vary randomly about its nominal value with an approximately Gaussian distribution as illustrated below. Assume that the mean, μ, is the nominal value, and the degree of variation is defined by the standard deviation, σ. If it is assumed that there is no intra-die variation, i.e. that all transistor parameters on a single die are identical, and the circuit is designed such that it is guaranteed to work when the parameter in question is within $\pm$ 3 standard deviations of the mean, the Gaussian probability of a die having its parameters outside this range falls to 0.2%. Then 99.8% of the copies of this circuit will be catered for in the design process. Reducing the bounds such that the design process caters for up to $\pm$2 standard deviations means that the 'parametric' yield may reduce, in theory, to 95.6 %. This may be an acceptable cost of reducing the complexity of the design. With $\pm$1 standard deviations the theoretical yield will be 68.2 %.

If we now consider the existence of many transistors on each die with intra-die as well as inter-die variability, things become more complicated. Worst case analysis could still be valid if the circuit is assumed to fail when a single one of its transistors has one of its parameters outside a set of worst case bounds. However,

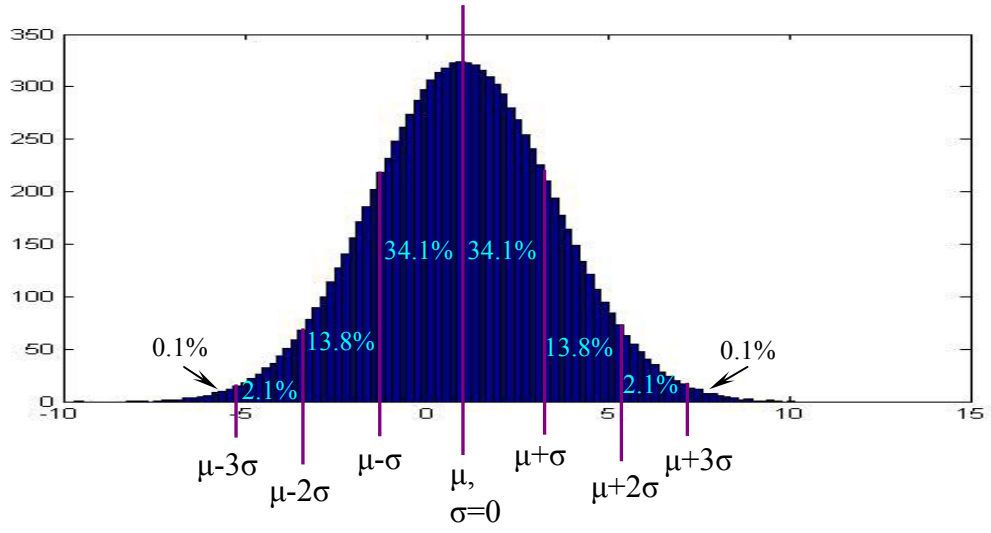


Figure 3.1: Gaussian probability density function about nominal value (mean μ) of a parameter when standard deviation is σ (Reproduced from P.Khurana & M. Jacobs, Cadence Design Inc.)

this is not a realistic assumption as there are many circumstances where variations in some parameters are less critical than in others, and where large changes in one parameter are compensated by complementary changes or the absence of large changes in others. If we are designing for deep sub-micron technology, the device-to-device variations to be anticipated may be very large and difficult to accommodate. Also, the fact that intra-die and inter-die variations may be correlated, for example among devices close together on a die, raises additional complexity.

3.3.2 Corners

To take a simple example, assume that two independent parameters A & B are both nominally 3 with standard deviation (σ) equal to 0.5 and 1 respectively.

In theory, combinations of A and B, each within ±2σ of its mean, can occur anywhere within the rectangle shown in figure 3.2. If a function F = A+B, its mean is 6 and its standard deviation is $\sigma = \sqrt{(0.5^2+1^2)} = 1.12$ and its minimum and maximum values are 9 and 3, or 6 ± 2.68σ. Therefore, if A and B each lies within ±2

standard deviations of its mean, F lies between ±2.68 standard deviations of its mean.

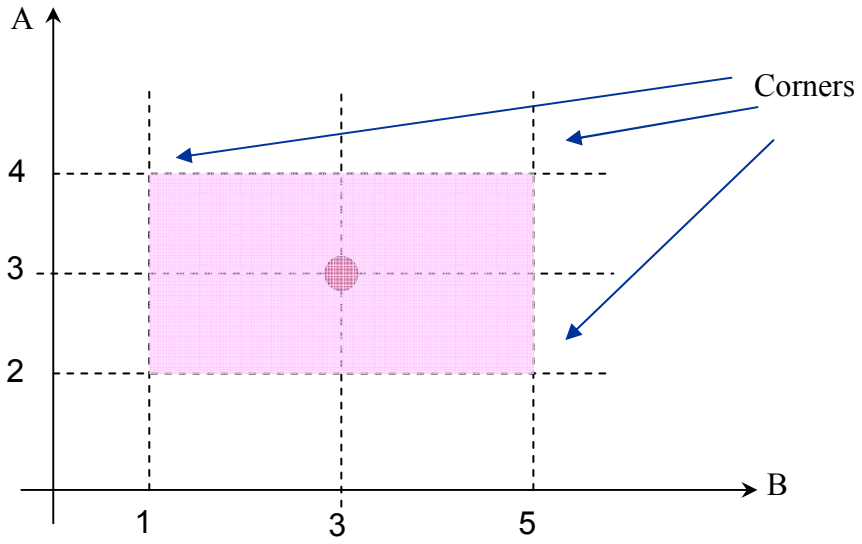


Figure 3.2: Illustration of 'worst case corners' for two variables

If F were a circuit parameter dependent on two device parameters A and B, designing the circuit to work for A and B each between ±2 standard deviations of its mean forces the circuit to be acceptable for F between ±2.68 standard deviations of its mean with a success rate ('yield') of about 99.6%. If the circuit is only required to work for F within ±2 standard deviations of its mean, corresponding to a success rate of about 95.6%, clearly the range of A and B for which the circuit must work can be relaxed. This simple example illustrates the mechanism by which forcing circuits to work at corners guarantees higher yield than may have been intended but this is at greater expense.

Assume the effect of increasing/decreasing parameter A speeds up/slows down transistor A, and similarly for parameter B and transistor B. If the two transistors are in serially connected gates, the slowing down of one gate could be compensated by the speeding up the other. Alternatively if they are in parallel, a speeding up of one gate may not matter so much if the parallel gate speeds up also. From a power consideration, high power consumption in some parts of the circuit may be compensated by lower power consumption in other parts. These are three examples

where the yield will be much greater than would be predicted by worst case predictions. With statistical methods, performance is modelled as a distribution rather than a deterministic quantity with 'worst case limits'. A circuit is analyzed, designed, and tested considering the statistical characteristics of the circuit based on the accurate modelling of the variability.

The size and complexity of circuits being implemented in deep sub-micron technology, and the problem of accommodating large parametric variations in, for example, threshold voltage (V_t), and gate oxide thickness due to non-systematic variations in the manufacturing process, form the background to this research project. Designers can no longer afford to ignore the intra-die variation or simplify the problem by producing conservative designs which are required to accommodate the worst-case corners. Effects like random dopant fluctuation (RDF) and line edge roughness (LER) [23], which can vary greatly from device to device on a single die, are becoming dominant as the transistor size is shrinking. Simplistic conservative designs will be extremely expensive or impossible. More sophisticated tools, such as Monte Carlo simulation [24]), as provided by HSPICE, may be useful as described below.

Process corners offer one method of designing integrated circuits taking into account expected variability in the technology process. In VLSI design and fabrication, a process corner occurs with all parameters at some maximum or minimum value within the range over which the circuit is designed to work correctly.

Variability occurs for many reasons, including changes in the humidity or temperature of the clean room used, and the locations of the dies on a wafer. With CMOS logic, two-letter acronyms are often used to denote process corners with the first letter referring to the N-MOS FET, and the second referring to the PMOS FET corner. Three corner types exist: typical, fast and slow. Fast corners occur when carrier mobilities are higher than normal and slow corners occur when carrier mobilities are lower than normal. Therefore, an 'SF' corner is caused by a slow N-MOS FETs and fast P-MOS FETs.

Five combinations are of interest: TT (typical-typical), fast-fast (FF), slow-slow (SS), fast-slow (FS), and slow-fast (SF). Combinations TT, FF and SS are called

'even' corners, since both types of devices are affected in the same way. Such combinations normally produce correctly operating gates which switch at the nominal speed (TT), a little faster (FF) or a little slower (SS). Combinations FS and SF are called 'skewed' corners for which one type of FET will switch faster than the other. This can cause gates to function incorrectly and latches to register incorrect logic values.

As well as variability in the parameters of the FETs, there are other on-chip parameters that have variability. On-chip variability (OCV) effects include 'process, voltage and temperature' (PVT) variability which affects interconnections between devices, as well as the devices themselves.

3.4 Introduction to Monte Carlo Simulation

As will be seen in detail in Chapter 4, Monte Carlo algorithms compute definite integrals of functions of vectors (containing many variables) by evaluating the function for large sets of randomised vectors covering the space or range of integration. In this thesis, the function will be some circuit parameter, for example a delay, as may be estimated by SPICE simulation. The vectors will contain variables such as the parameters of transistors and other components such as wires, which in practice will be expected to vary randomly. The definite integral will be the volume of the 'tail' of the probability density function (PDF) where some aspect of the performance, for example the delay, falls outside some defined limit. The parameter values of transistors and other components will have particular statistical distributions and correlations determined by the physics of the fabrication process and many other effects, and these must be represented by the choice of vectors supplied to the Monte Carlo process. Therefore repeated SPICE simulations must be performed for the randomised vectors to generate the required distribution of circuit measurements.

For the circuits considered in this thesis, the dimensions of the input vectors, i.e. the number of variables, will be extremely high. Each transistor model may have as many as 300 parameters, and there may be a very large number of transistors.

Although Monte Carlo methods are known to be efficient for very high dimensional applications, the computational complexity of this application is likely to prohibitive for all but the very simplest circuits. Therefore it is vital to find ways of reducing computational complexity. There are many possibilities, one of which, proposed by Amith Singhee [15], makes use 'Quasi Monte Carlo' methods [25] as described below. The SiLVR model, and 'statistical blockade', also proposed by Amith Singhee [15][21][29] achieve computational savings by different methods. The SilvR method falls under the category of response surface models which gain speed by sacrificing accuracy. Monte Carlo techniques are particularly well suited to IC design in nano-scale technology.

Quasi Monte Carlo methods are modified forms of Monte Carlo methods where the input vectors are not totally random, but are to a degree deterministic in that they conform to 'low-discrepancy sequences' [15][36]. More detail will be given in Chapter 6.

3.5 Statistical Static Timing Analysis (SSTA)

High-performance integrated circuits are traditionally characterized by the clock frequency range over which they can operate. Determining this at the design stage requires an ability to estimate the delay at different parts of the circuit. Such delay estimations must be incorporated into optimization processes at various stages of design, such as the logic synthesis stage, the layout (placement and routing) stage, and the in-place optimizations that are performed prior to finalisation. Such estimates can be performed by circuit simulation, but this may be too computationally demanding in some cases. Static Timing Analysis (STA) is a method of estimating the expected timing behaviour of a digital circuit without requiring complex simulation. STA plays a vital role in efficiently obtaining reasonably accurate estimations of circuit timing behaviour. The efficiency arises from the use of simplified delay models, though its ability to consider some of the logical interactions between signals is limited. Nevertheless, it has been a widely used approach for many decades.

The word 'static' means that the circuit analysis is carried out in an input-independent manner, and aims to estimate the worst-case delay over all possible input combinations. The computational requirement is linearly dependent on the number of components. STA is widely used despite its limitations. For example, it cannot easily take into account within-die correlation due to spatial characteristics. It needs to consider many corners and for high degrees of random variability and its conservative estimations are too pessimistic for the design of circuits which are economic in their use of circuit resources.

In recent years, the increased variability of device parameters and the behaviour of interconnections between devices have introduced design problems that cannot be successfully solved using traditional (deterministic) STA methods. Statistical static timing analysis (SSTA) represents the timing behaviour of devices, logic gates and interconnections by probability distributions. It is thus possible to obtain distributions of circuit behaviour estimations rather than a single estimation.

SSTA can model correlations among circuit parameters to compute more accurate statistical distributions of circuit measurements, for example of overall delay. There are two main types of SSTA algorithms: path-based and block-based. Path-based algorithms [88] sum device and wire delays for specific paths over which signals may propagate within the circuit. The paths must be identified prior to running the analysis and this is difficult and has the danger of missing paths which are critical, or of identifying far more paths than it is feasible to analyse. Block-based algorithms [89] [90] generate the required and actual arrival times of signals for each component or sub-circuit, working both forwards and backwards from each clock signal source. There is now no need for path selection, but statistical estimates of maximum or minimum delays are needed that take into account correlation that will exist between the delays [91]. These are very difficult to derive. The main reason we prefer Mont Carlo methods rather than SSTA in this thesis is that SSTA focuses on just one specific problem.

3.6 Integrated Circuits (IC) Design Flow

Analysis of the effect of variability may be executed at several stages of the IC design flow. With deep submicron technology, there is an increasing number of previously insignificant effects that are now becoming first order effects. For sub-wavelength nano-meter processes, effects such as resistance, inductance, crosstalk, leakage, and electro-migration become significant. If they are not taken into account in the initial design stages, an additional number of design iterations may be required in order to fix problems found very late in the design cycle. Ensuring an expected performance with manufacturability, cost scaling and economical use of power and area, within a reasonable design cycle time is today's challenge. The aspects mentioned, and several others, need to be dealt with at all levels of the design flow including technology processing, data extraction and library modelling, logic synthesis, circuit design, placing and routing, clock distribution, verification, and finally testing and assembly.

3.6.1 IC Design Flow

Design flows are the sequences of actions that are performed, using specific automated or semi-automated design tools [12], to accomplish the design of integrated circuits. Design techniques are required for both analogue and digital ICs. Analog ICs realize amplifiers, filters, modulators, oscillators, regulators and phase locked loops, for example. Analogue IC design techniques for power applications and signal processing applications tend to be rather different. In both application fields, the techniques are more concerned with the physics of the semiconductor devices and parameters such as gain, power dissipation, and resistances than is the case with digital IC design. The fidelity of analogue signal processing is usually critical and consequently analogue ICs are generally less dense in circuitry and require transistors which require larger areas of silicon than digital ICs.

Digital ICs realise microprocessors, memories (RAM, ROM, and flash), FPGAs and digital ASICs, for example. Designing digital ICs must achieve logical correctness, maximize circuit density, and ensure that connections, especially those

carrying clock and timing signal, are routed efficiently. Digital IC design can be considered to have three phases.

1. System-level phase which creates a functional (or 'behavioural) specification for the required circuit. The functionality is expressed in the form of an input-output model that suppresses the details about gate and physical level implementations. This may be done in various ways, for example, by developing System-Verilog Transaction Level or C/C++ models of what is required, or by using the facilities of languages such as MATLAB, SIMULINK or SystemC. The intended input-output relationship can be verified using functional simulation.
2. Register Transfer Language (RTL) description phase which converts the functional specification into an RTL description which describes the exact behavior of the required digital circuit and its interconnections on the chip. At this level, the design is a data-flow model consisting of components and their interconnections. The RTL phase is responsible for making the chip do what is required.
3. Physical design phase which begins by converting the RTL description to a viable chip description in the form of a gate-level netlist. This is a technology independent description of the circuit in terms of standard cells such as gates, latches, multiplexors, counters and interconnections between them. The synthesis tool must ensure that the netlist meets timing, area and power specifications. A suitable library of logic gate realizations must then be adopted and decisions must be taken regarding which gates to use, where they are to be placed on the chip and how they are to be interconnected. This 'library binding' or 'technology mapping' process transforms the design into a vendor-specific network based on parameterized cells. Vendors usually provide the physical layout, timing models and behavioural models for each cell to allow for checking. The final task is to input the netlist to an automatic 'Place and Route' tool to generate the physical layout, which is verified and then fabricated as an IC chip. The physical design phase is not intended to affect the

functionality, but the way it is executed strongly determines how fast the chip will operate and how much it will cost.

RTL design is the most difficult of the above three phases, and it must include functional verification. Simple statements in the functional description often require thousands of lines of computer code to be implemented and verified. It is difficult to verify that the RTL description correctly caters for all possible cases that may arise. Techniques that are used include extensive logic simulation, formal methods, hardware emulation and lint-like code checking. The consequences of an error at this stage remaining undiscovered can be catastrophic as was the case with the Pentium FDIV bug in 1990 which remained unnoticed until the processor had been in production for months.

The main steps involved in the physical design phase are summarized below in more detail. Iteration between the steps is generally required, with steps repeated until all objectives are met simultaneously and design closure is achieved.

1. Floor-planning which means that RTL description is assigned to regions of the chip, input/output pins are assigned and areas are reserved for memory elements, cores and other parts of the circuit identified as requiring large surface areas.
2. Logic synthesis to map the RTL description onto a gate-level netlist appropriate to the selected technology.
3. Placement of the gates specified in the netlist onto non-overlapping locations on the die.
4. Iterative refinement of the logic gate placements by repeated logical and placement transformations to satisfy performance specifications and power constraints.
5. Clock insertion by introducing clock signal wiring into the design.
6. Routing to add the ‘wires’ that inter-connect the gates in the netlist.
7. Post-wiring optimization to remove violations of performance (timing), signal integrity and yield requirement.

8. Design modifications for manufacturability to make the circuit as easy and efficient as possible to deal with. These modifications are sometime achieved by adding extra vias or metal diffusion layers.
9. Final checking to make sure that the mapping to logic was done correctly, and that the manufacturing rules have been adhered to.
10. Tape-out and mask generation to turn the data into photo-masks.

In recent years the demands of Moore's law development has required standalone synthesis, placement, and routing algorithms to be replaced by integrated construction and analysis flows[19]. The increasing importance of allowing for interconnect delay, leakage power, variability, and reliability are changing design procedures in many fundamental ways. The lack of good predictors for delay has called for significant changes in design flows. It is said, with respect to IC design, we are now entering the 'age of integration' after passing through ages of 'invention' and 'implementation'. During the invention age, routing, placement, static timing analysis and logic synthesis were invented. During the age of implementation, these independent procedures were improved to cater for rapidly decreasing device sizes. With present day technology, the inability to devise meaningful cost functions has made it impossible to execute design flows in discrete step. Hence the age of integration begins, where the design steps are based on incremental cost analyzers.

3.6.2 Asynchronous Circuit Design Flow with Balsa

3.6.2.1 The Balsa Development System

'Balsa' [5] is both a framework for synthesizing asynchronous hardware systems and a language for describing such systems [5]. It has been developed over a number of years by the APT group of the School of Computer Science at the University of Manchester. It is built around a 'handshake' circuit methodology and can generate gate-level 'net-lists' (alpha-numerical descriptions of arrays of logic gates and their interconnections) from high-level descriptions of asynchronous circuits expressed in the BALSA language. Both dual-rail (QDI) and single-rail (bundled data) circuits [6] can be generated.

A Balsa 'backend' system allows for implementations in different technologies and different asynchronous styles. The technologies correspond to different cell libraries which may be either custom-built or vendor-supplied standard cell libraries. At this level, several nano-scale technologies, such as 45nm, 30nm and 18nm, could be added to Balsa to implement circuits in nano-technologies. Although variability analysis cannot be carried out at the Balsa circuit description level, the gate-level net-lists of handshake circuits created by Balsa may be analysed at lower-level design stages to perform variability analyses. Figure 3.3 shows the complete design flow for asynchronous circuits based on Balsa with the possibilities for simulation at various levels.

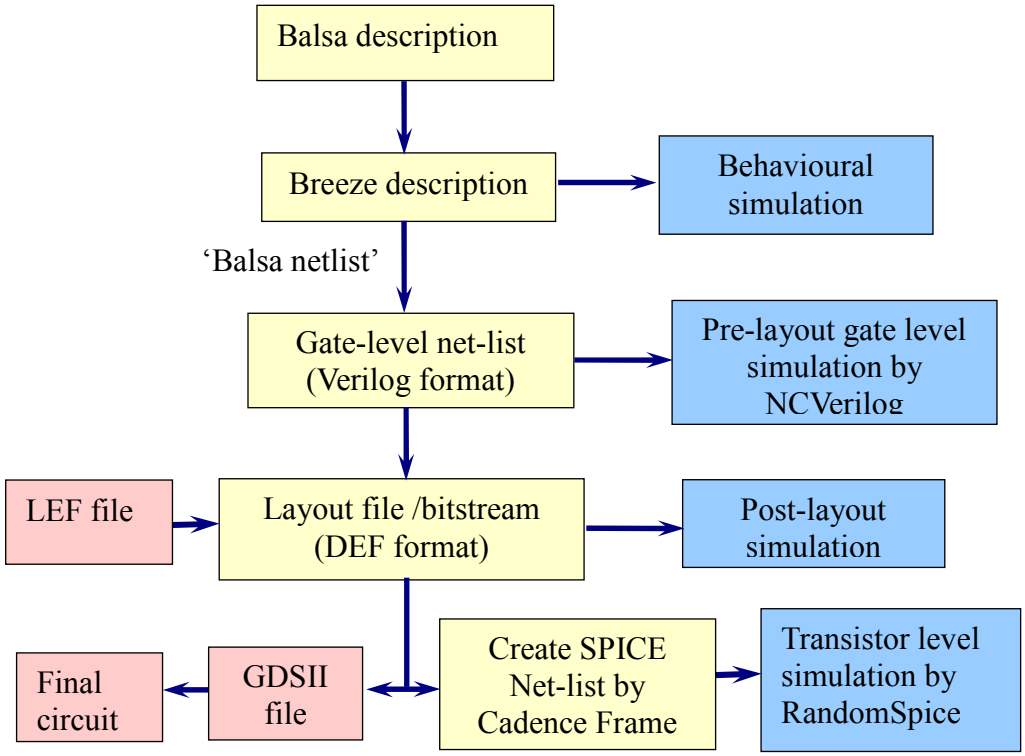


Figure 3.3: Asynchronous Circuit Design Flow based on 'Balsa'

3.6.2.2 Analysing Reasons for Failure

Statistical analyses for failure prediction for the purposes of optimisation may be undertaken at the layout (cell) level and at the transistor level. After layout, if the timing, power or yield do not meet the design specification, we go back to the Balsa behavioral description to change the circuit structure or implementation style and then re-compile the circuit. To decide how to modify the circuit, it will be necessary to know why the circuit failed.

For circuits failing to meet the timing specification, 'timing files' and diagrams produced by the simulation package may be examined to determine which part of the circuit has failed due to excessive delay. Similarly, simulation results may be examined to determine parts of the circuit where dynamic or static leakage current and local power consumption is excessive. Statistical techniques are very useful here as alternatives to the more widely used analytical and 'worst-case' analysis techniques for predicting power dissipation since they take into account fabrication variations. Localised power consumption may be reduced at particular parts of a circuit possibly at the expense of switching speed where switching speed is not critical.

When the Encounter simulation meets the design specification, we still need to do more precise analysis at the transistor level, using SPICE. If the SPICE simulation does not match the design specification at the transistor level, we may need to go back to earlier design stages, maybe to the layout stage or right back to the Balsa specification, to further modify the design.

3.6.2.3 Modifying the Balsa Circuit Description to Increase the yield

Yield predictions based on timing, power consumption and performance are references for circuit optimization. Circuit optimization may be performed at the Balsa behavioral description level. With Balsa [5], an asynchronous circuit may be implemented with a choice of different styles for some functions. The choice may be made for different purposes, such as reducing area, increasing speed, reducing power consumption and increasing robustness.

As shown in Figure 3.3, the Balsa description is converted via an intermediate 'Breeze' format to a gate-level net-list in Verilog format and then to 'design exchange format' (DEF) files which may be exported to 'GDSII' files as required for a final integrated circuit fabrication. GDSII is a binary file format representing the required layered geometric shapes, text labels and other information needed for IC fabrication. 'Layout exchange format' (LEF) information, as supplied by the circuit manufacturer is required for the conversion to DEF. The DEF format may also be converted to SPICE net-lists to allow transistor-level simulation, with statistical parameter variation, using HSPICE. Figure 3.3 illustrates four levels of simulation that may be carried out to verify the intermediate results of the design-flow. These are behavioral, functional (gate level), layout and transistor level simulations.

Behavioral simulation implements the statements of the Balsa description as if they were a computer program, without any regard to how they are intended to be implemented on a circuit. Pre-layout gate-level simulation tests just the functionality of the logic gates without consideration of timing effects and delays. Post-layout gate-level simulation takes into account the switching delay of gates and the delays introduced by the wired connections among gates. Transistor level simulation performs a mathematical analysis on the circuit produced from the Balsa description with gates replaced by their actual CMOS circuitry and the MOSFETs within this circuitry modeled by standardized circuits composed of resistors, capacitors and controlled voltage and/or current sources. The values of the MOSFET model components are the parameters of the model.

The results of any of these simulations may be fed back to earlier design stages to cause adjustments to be made to eliminate problems identified by the simulation processes. Two of these four simulations, i.e. the layout simulation as performed by simulation tools provided by the Encounter package, and the 'SPICE' transistor level simulation, can take into account statistical variations in device parameters. Some detail about these packages will now be given.

3.7 Simulation of ICs by EDA Tools

The simulation of statistical variations in integrated circuit characteristics is feasible at the layout (DEF) level using tools provided by the Cadence 'Encounter' platform and at the switching transistor device level using SPICE in its commercial version, HSPICE. Some details of these simulation facilities are now given.

3.7.1 Encounter

The Cadence 'Encounter' platform [10] is an integrated software package for complex and low-power integrated circuit design. It provides a complete design-flow from 'register transition language' (RTL) specification to GDSII (or equivalent) stream format for fabrication and incorporates tools for test design, 'virtual prototyping' by simulation, partitioning, and timing. It claims to deliver the required CMOS circuitry with accurate verification and 'signal-integrity-aware routing'. Yield prediction and low-power design capabilities have recently been included.

Statistical variability analysis is made feasible by the provision, by Cadence, of a set of library models referred to as the 'Encounter Library Characterizer'. This is claimed to incorporate the required statistical timing and leakage parameters as supplied by major 'intellectual property' (IP) vendors. These parameters characterise global, local, and random process variations as required for statistical static timing analysis (SSTA) and statistical leakage analysis. Timing, noise and power aspects of devices are included in the representation. Although CMOS technologies below 45nm have not yet been included in this commercial design tool, for our research, which focuses on technologies below 45nm, we can still take advantage of the 'Encounter Library Characterizer by plugging in statistical model parameters that will be obtained from research partners. This should allow the Encounter simulation tools to be applied to sub-45 nm technologies to investigate the impact of variability on timing, power dissipation and yield. To do this, it will be necessary to build up Library Exchange Format (LEF) files containing parameters of cells of size 45nm and below. After layout, placing and routing, 'design exchange format' (DEF) files

would be created, which could be applied to the 'Design Rule Check' (DRC) within the Cadence Framework, or exported to GDSII files for integrated circuit fabrication.

The analyses of timing and power efficiency performed by Encounter are carried out after the layout procedure and are based on models of cells and their inter-connection. They are not the most precise analyses possible. More precise circuit simulations and analyses may be achieved at transistor level using the simulation package 'HSPICE'. The use of HSPICE for performing yield, timing and power analysis, at transistor level, is discussed in the following section.

3.7.2 Introduction to HSPICE

HSPICE is a component of the 'Synopsys' comprehensive mixed-signal verification package [11]. It is widely used for analogue and digital circuit simulation and combines validated integrated circuit device models with advanced simulation and analysis algorithms. It is useful for predicting the timing characteristics, power consumption, and functionality of circuits before they are actually fabricated. HSPICE is able to perform statistical 'Monte Carlo' type and worst-case 'corner' type analyses. Circuit optimization is also provided for the creation of circuits that satisfy the design constraints across various processes, voltages and temperatures. The circuit optimization features of HSPICE support multi-parameter optimizations based on AC, DC, and transient analysis.

HSPICE's worst case 'corner' analyses can be used to predict guaranteed yield, power efficiency, and performance. Parameter variation limits must be known for such analysis. To simulate the worst cases, HSPICE sets all variables to their 2-sigma or 3-sigma standard deviation 'worst case' values. Because several independent variables rarely attain their worst-case values simultaneously, this technique tends to be overly pessimistic, and lead to over-designing the circuit. However, this analysis is often useful as a fast check.

The 'yield' predictions resulting from HSPICE's Monte Carlo type statistical analyses aim to predict the viability of circuit designs in the light of statistically modeled parameter variations considered likely to occur in fabricated circuits. As

defined by HSPICE, the 'yield' (or 'parametric yield') is the percentage of integrated circuits that are found to meet the electrical test specification when appropriate statistically modeled parameter variations are imposed on a large number of copies of a given circuit. The 'fabrication yield' as ultimately measured from samples of real fabricated circuits will be affected by other factors, such as wafer defects, not modeled by HSPICE. Monte Carlo analysis requires knowledge of the statistical distributions and standard deviations of parameter values likely to occur in fabrication. It uses pseudo-random number generators with Gaussian, uniform or random limit distributions to simulate the specified statistical variability. A 'random-limit distribution' function is defined by Synopsys [31] as an absolute variation of +/- A from the nominal (or mean) value of some parameter where A is a fixed limit value and the choice of +A or –A is random. This may be considered as a binomial distribution about the nominal parameter value. Where there are many parameters, the use of random limit distributions for each gives a 'worst case' or 'corner' simulation, though this is not explicitly stated. The results of Monte Carlo analysis may be fed back to earlier stages of a design process to try to optimise process yield with the assumption of realistic parameter tolerances.

HSPICE supports behavioral modelling which according to Chapter 26 of the User Guide [31], 'substitutes more abstract, less computationally intensive circuit models for lower level descriptions of analog functions'. There are several ways of defining behavioral models of sub-circuits offered by HSPICE, including the use of a version of Verilog, the programming of logic functions in algebraic form, the use of sampled waveforms and the use or switches and controllable sources.

To cater for the demands of modern semiconductor technologies, a new approach was recently introduced [31] based on the concept of a 'Variation Block' which allows both global and local variations to be specified and characterised for each parameter of any model. For global variations, each parameter of all devices within a circuit that use a given model is changed by the same random value. There is no intra-die variation in this case. For local variation, a specified parameter is varied by a different random value for each device within the circuit. This is intra-die variation as described in Section 2.3.3. The effects of the defined global and local variations are added together for each parameter. The randomised copies of the

circuit are saved in files, and once the randomisation process is complete, the required series of HSPICE simulations is executed and the measurement results for each run are saved. When series of simulations is complete, HSPICE performs a statistical analysis of the results and generates the output as specified.

A number of sampling methods may be defined by the Variation Block including:

(a) Factorial Sampling for estimating worst case and best case behavior by evaluating the circuit response at the extremes of the ranges of values for the parameters

(b) One-Factor-at-a-Time Sampling where 2m+1 circuit copies are generated and analysed when there are m independent variables. For the first copy, there is there is no perturbation, then a specified degree of first negative and then positive perturbation is applied to each variable in turn.

(c) Latin Hypercube Sampling is a form of quasi-Monte Carlo analysis which reduces the number of randomisations that are needed for a given accuracy.

The Variation Block replaces older methods of defining the variability of integrated circuits within HSPICE and has the advantages of consolidating all variation definitions within a single record, distinguishing global and local variability and allowing different aspects of variation to be selected. Local and global variations may be defined as functions of device geometry, and local variation may be specified as a spatial function of device 'on-chip' location.

As mentioned in Section 2.2.3, spatial variations are due to material properties and imperfections of lenses and spin processes. There are, as yet, no industry-wide standards for specifying such process variability, so the Variation Block allows any company to implement its own model for each of its technologies based on the measurement of test circuits.

Like device models, Variation Blocks can be encrypted to make them inaccessible to designers. A Variation Block has a general section and three sub-blocks for specifying global variability, local variability and spatial variability. Each sub-block can add extra information about the characteristics of a particular model. Therefore each device may be referred to three times, once in each sub-block, the resulting effects being added together. Within the variation sub-blocks, any number

of independent random variables can be defined with a Gaussian distribution assumed by default, though other distributions may be specified. Correlation between a given parameter and any other parameters may be introduced by including a reference to these other parameters in the definition of the given parameter. Dependent variables can thus be defined as functions of more than one independent random variable. For spatial variation, the sub-block needs to know an x and y co-ordinate for each of the devices.

The contents of the Variation Block are intended to be created by a foundry. A simple example is available in the HSPICE demo directory [31] to illustrate how global variations on transistor parameters vth0 and u0 are introduced by the 'global sub-block' and local variations on these same parameters as a function of device area may be introduced by the 'local sub-block'. Local variations on the implicit value of resistors (relative) are also illustrated.

3.7.3 Introduction to NGSPICE

NGSPICE is an open-source version of the HSPICE circuit simulator capable of performing basic SPICE simulations. The SPICE netlists for NGSPICE and HSPICE are slightly different in the format of commands. NGSPICE is based on three open source software packages: 'SPICE', 'Cider' and 'Xspice'. 'Cider' is a mixed-level simulator that adds a device level simulator to the functionality of SPICE. Cider thus improves achievable simulation accuracy, at the expense of greater simulation time, with critical devices characterised by tables of technology parameters. Less critical devices may still be characterised by the original SPICE compact models. Xspice is an extension to SPICE that allows digital components to be modeled by coded functional software and simulated by an embedded event-driven algorithm. Both HSPICE and NGSPICE may be used by the harness developed in this thesis for Monte Carlo type circuit simulations.

3.7.4 RandomSPICE

RandomSPICE is a software package developed by the DMG (Device Modelling

Group) in Glasgow University. DMG are one of the 'NanoCMOS' project partners. RandomSPICE was designed to be used as a 'front-end' to some version of SPICE, such as HSPICE or NGSPICE. Its function is to allow statistical variability analysis to be applied to a given circuit realization by 'randomising' the parameters of its devices. The effect of variability may be statistically studied by analysing different versions of the same circuit and observing the differences in output.

3.7.4.1 The Randomisation

RandomSPICE allows the repeated SPICE simulation of a given circuit with randomised selections of transistor parameters from the given libraries. The software application, written in Python, takes a standard SPICE netlist as its input, and produces a batch of netlists, each of the MOSFET devices replaced with randomized models from the variability-aware libraries.

3.7.4.2 Restrictions

In the versions of RandomSPICE available for this thesis, the transistor devices were restricted to channel-lengths of a single fixed size. Future versions were expected include the ability to vary channel-lengths between a set of discrete values. In practice, the channel-widths can be set to an integer multiple of the supplied channel-length by the expedient of using parallel devices, but this is cumbersome and inefficient. With future revisions, sub-integer multiples may be possible. In reality, it is generally possible to create sub-multiple lengths through the use of lithographies with RET. A library based on 32-nm minimum channel lengths could include devices with 48-nm channel lengths and widths.

RandomSPICE and its libraries were under development while this thesis was being produced. The software application has changed in various ways. Originally, if a device with a channel width four times the channel length was specified in the supplied netlist, it was replaced in each of the randomised netlists by a parallel sub-circuit consisting of four square devices each with channel width and length equal. In later versions of RandomSPICE, models were created with channel-widths of 1, 2, 4

and 8 times the channel length. Thus the rectangular transistor referred to above could be randomized as a single device in each output netlist. This reduced the simulation time significantly for each randomised netlist. A further highly significant improvement was made to RandomSPICE in response to discussions involving the work of this thesis. We found that the original version was not realistic in the way it randomized sub-circuits containing several transistors; a logic gate for example. Clearly such a sub-circuit would be employed many times in a typical logic circuit. The parameters of the devices were appropriately randomized within the sub-circuit, but were found to be identical for each copy of the sub-circuit. The problem prevented us from using RandomSPICE directly, though the randomized library remained useful. RandomSPICE was later remedied to randomize sub-circuits appropriately, but this thesis had by then developed its own harness based on MATLAB and HSPICE.

3.7.4.3 RandomSPICE Transistor Model Libraries

Two 'variability libraries' are referred to in this thesis. The first is based on a Toshiba 35X35nm device, with only the effects of RDD (Random Discrete Dopant) simulated. This library has only single width devices. The second library is based on 35X35nm high performance models, which include the effects of LER and surface-roughness in addition to RDD. It has higher accuracy than the first library and includes of multi-width models as mentioned in the previous section. Both libraries contain 200 NMOS and 200 PMOS devices for each width.

3.7.5 Statistical Analysis with RandomSPICE

RandomSPICE, as illustrated in the block-diagram below, generates an ensemble of circuits, each using randomly different parameters for each nominally identical device. To conform to the original version of RandomSPICE, non-square transistors are 'decomposed' into sub-circuits containing only square transistors to reduce the range of transistors that need to be characterized. Originally, the RandomSPICE documentation did not refer to any distinction between intra-die and inter-die

variability and the modelling of correlation between the imposed variations. A representative library containing 201 randomised copies of two 35 nm devices was provided, the parameters having been devised by simulating the geometric and physical properties of the fabrication technology [32]. The randomised libraries have been produced to reflect parameter variation representative of the desired technology. Ideally they should also have realistic intra-die device-to-device correlation. It is clear that the concept of RandomSPICE can be used with such custom randomized libraries to achieve useful statistical analysis, though the design of the customized libraries may have to be modified and extended.

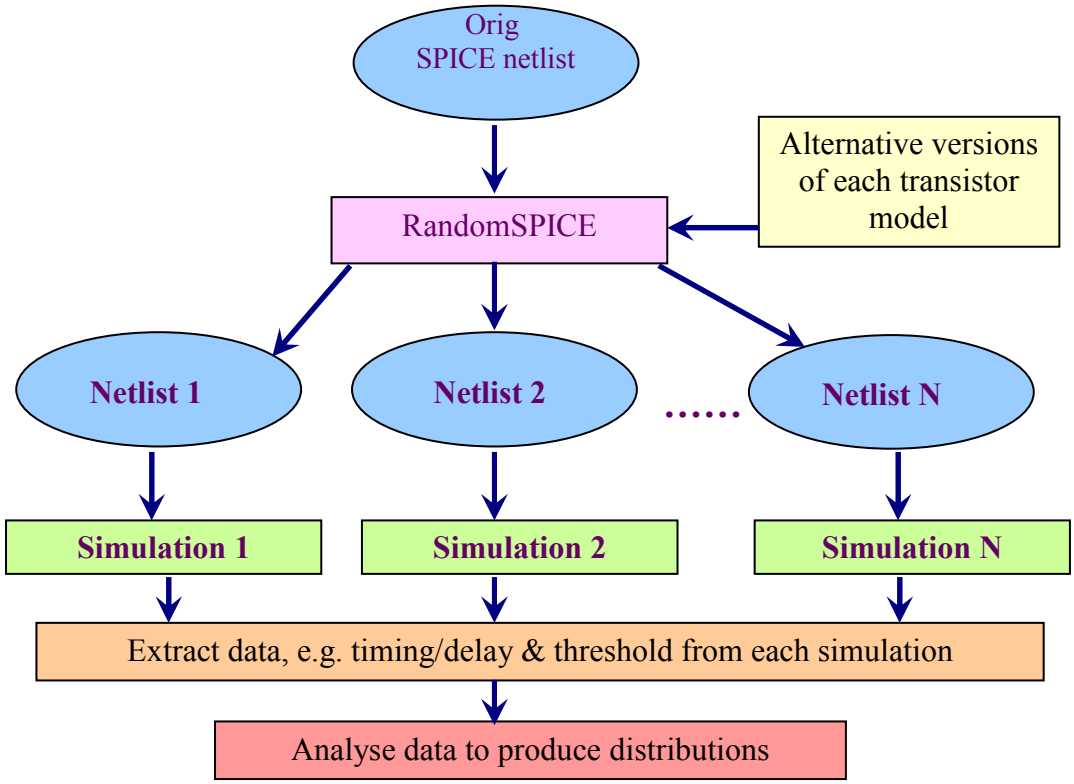


Figure 3.4: Functional block diagram for RandomSPICE

The randomization of nested sub-circuits (or sub-blocks) within a SPICE netlist raises interesting problems and opportunities for reducing computational complexity.

As mentioned above, RandomSPICE originally made all sub-circuits identical in each randomised circuit copy, thus eliminating intra-die variability. RandomSPICE has more recently been given a mechanism for introducing intra-die variability between sub-circuits, but at the time of writing this thesis, it still has to be fully tested. The efficient randomization of nested sub-circuits is a problem to be addressed by the research collaboration and this PhD project. The use of 'behavioural models' of sub-circuits, as proposed by Southampton University has great promise for reducing the computational complexity of statistical variability analysis. A further opportunity for reducing the computation required for RandomSPICE analyses of larger circuits is the use of quasi MonteCarlo and extreme value theory based approaches, as proposed by Amith Singhee [15][26][27], which could be introduced into the randomised libraries of RandomSPICE and the way they are used.

3.8 Conclusions

The need for statistical analysis techniques for current and next generation integrated circuits is clear in view of the over-pessimistic predictions of circuit failure that are given by traditional 'worst case' analysis techniques. The idea of combining Monte Carlo techniques with SPICE circuit simulation is an obvious approach to the required statistical analysis, and it has been widely studied. Alternatives such as statistical static timing analysis have also been proposed and used successfully. The role of statistical analysis in the design process for integrated circuits is outlined and is clearly of great importance. The context of the research in this thesis is asynchronous circuit design and some background is given on this context. The widely used design tools provided by Synopsis (HSPICE), Cadence (Encounter) and NGSPICE have been introduced and it is clear that the current version of HSPICE already has quite extensive facilities for Monte Carlo type statistical analysis, and even a form of quasi Monte Carlo analysis in the form of 'Latin Hypercube Sampling'. The open source version 'NGSPICE' did not have any of these facilities until recently (June 2011) and even now (January 2012), only rudimentary MC

techniques are supported. Their relevance to IC design and simulation is explained with some examples. The software package 'RandomSPICE' developed by Glasgow University was designed as a 'front-end' to any version of SPICE for statistical variability analysis. This package, and extending its functionality, was the inspiration for the work in this thesis.

Chapter 4

Monte Carlo Simulation for the Design of Nano-Scale Integrated Circuits

4.1 Introduction

In nano-scale integrated circuits (ICs), the main sources of failure are likely to be due to intrinsic atomic scale variations of materials and component dimensions. These 'atomistic' variabilities can only be considered random in nature. Their effects are so significant that to design such ICs effectively, new circuit analysis techniques are needed. These new techniques must adopt a statistical rather than a 'worst case' treatment of the variability of device performances. Traditional Monte Carlo (MC) based statistical analysis may be used for predicting the likely performance, yield and failure probability of an IC design, before it is fabricated, by carrying out analogue simulations of many possible realizations of it. This is referred to as 'Monte Carlo Simulation'. MC simulation is flexible, robust to large numbers of device parameters and allows arbitrary accuracy given sufficient computational resources.

4.2 Monte Carlo Methods

As introduced in Section 3.4, Monte Carlo methods use repeated random sampling of the behaviour of mathematical equations, or real or simulated systems, to solve mathematical problems or to determine the properties of systems [92]. In this thesis, the systems are integrated circuits simulated using SPICE. The transient behaviour

of sets of highly complex multidimensional equations describing integrated circuits is being analysed by repeated random sampling. The repeated random sampling produces observations to which statistical inference can be applied to obtain information about the equations or systems.

The name Monte Carlo refers to the famous casino in Monaco. Just as gambling requires a random process such as the spinning of a roulette wheel, the throwing of a six-sided dice or the dealing of well shuffled playing cards, Monte Carlo methods use pseudo-random processes implemented in software. The term 'pseudo-random' means that the software process is not truly random. It is, in theory, deterministic because a person who knows the algorithm that is being used can predict precisely the variables that will be generated. However, with a little care, pseudo- random variables can be used to simulate realistically the effects that true physical randomness would create. The simulation is not required to be numerically identical to the true physical process since the aim is to produce statistical results such as averages, expectations and distributions rather than deterministic numerical measurements. The derivation of such results requires a 'sampling' of the population of all possible modes of behaviour of the system.

One of the most often quoted applications of Monte Carlo methods is the evaluation of multi-dimensional integrals [102]. It may be illustrated by integrating $\sin(x)$ over $0<x<\pi$ by generating a set of N pseudo-random number pairs (x_i, y_i) uniformly covering the area $0<x<\pi$, $0<y<1$ as illustrated below:

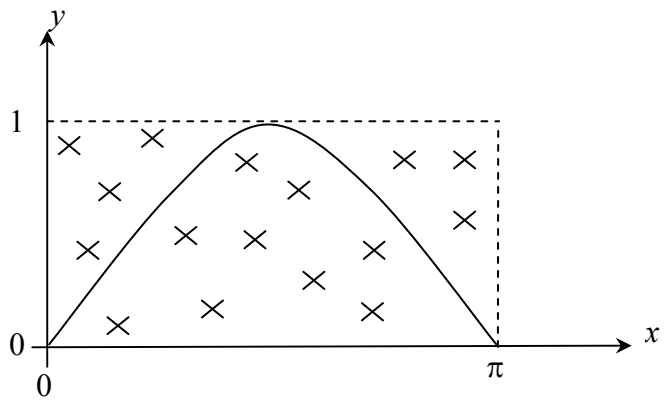


Figure 4.1: MC integration of $\sin(x)$, $0<x<\pi$

If the number of pairs that lie under the sine-wave curve is counted and found to be M, the ratio M/N will be an estimate of the ratio of the area under the curve to the area of the rectangle, which is π. Increasing N can make the estimate more accurate. Therefore, π × M/N will approach

$$\int_0^{\pi} \sin(x)dx \tag{4.1}$$

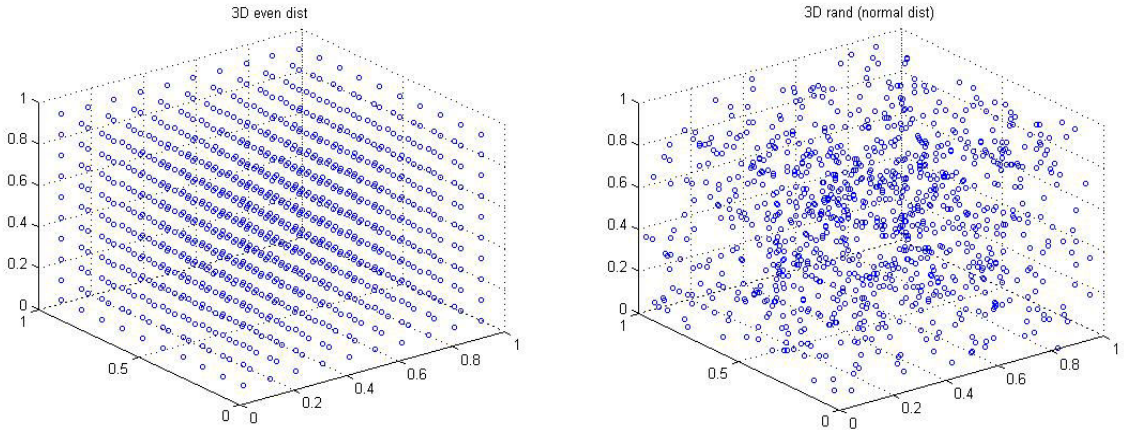
as N tends to infinity. This example illustrates the simplicity of MC techniques, but not their computational advantages in comparison to numerical integration with regularly placed rather than randomly placed points in the rectangle. In fact, there are no advantages for a one-dimensional problem.

Consider a multi-dimensional integration:

$$I = \int_{a_1}^{b_1}\int_{a_2}^{b_2}\int_{a_3}^{b_3}\ldots\int_{a_K}^{b_K} f(x_1, x_2, x_3, \ldots, x_K)dx_K dx_{K-1}\ldots dx_2 dx_1 \quad \text{written as} \quad \int_V f(\underline{x})d\underline{x} \tag{4.2}$$

Here, $\underline{x}$ is the vector (x_1, x_2, x_3, ..., x_K), f is some function of the K variables of $\underline{x}$ and V denotes the region of integration. Again, $f(\underline{x})$ may be evaluated at regularly spaced points or uniformly distributed random points in K-dimensional space as a means of evaluating the integral. However, the advantages of the Monte Carlo method with randomly distributed points now become apparent. The problem with regularly spaced points is that the number of them, N say, must increase exponentially with the dimension K if the error is not to increase exponentially with K. The error with regular spacing and a fixed value of N is known to increase as the Kth root of the order of magnitude [99]. The error for one-dimensional 'trapezoidal rule' integration with N regularly spaced points can be shown to be proportional to $1/N^2$ whereas the error for the K-dimensional equivalent of the trapezoidal rule has an error proportional to $1/N^{2/K}$. This is 'the curse of dimensionality' [96]. With regular sampling and fixed N, as K increases, each dimension must be sampled more and more sparsely and less and less efficiently, since more and more points will have the same value in a given dimension. Monte Carlo sampling avoids the inefficiency of the rectangular grids created by regular sampling by using a purely random set of N points uniformly distributed over the K-dimensional region V. Two illustrations of the positions of 1000 points distributed over a cube in 3-dimensions are shown in

Figure 4.2. Figure 4.2(a) illustrates the regularly distributed point-set that would be used for trapezoidal integration, and figure 4.2(b) shows a random uniform-distributed point set as would used for Monte Carlo integration.



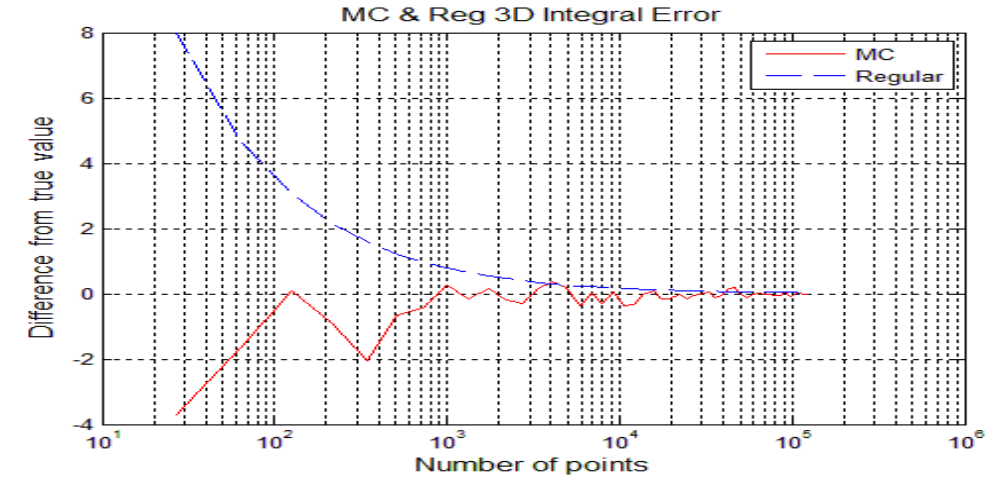
(a) (b)

Figure 4.2: Point-sets for three-dimensional integration:
(a) regularly spaced, and (b) uniform random distribution.

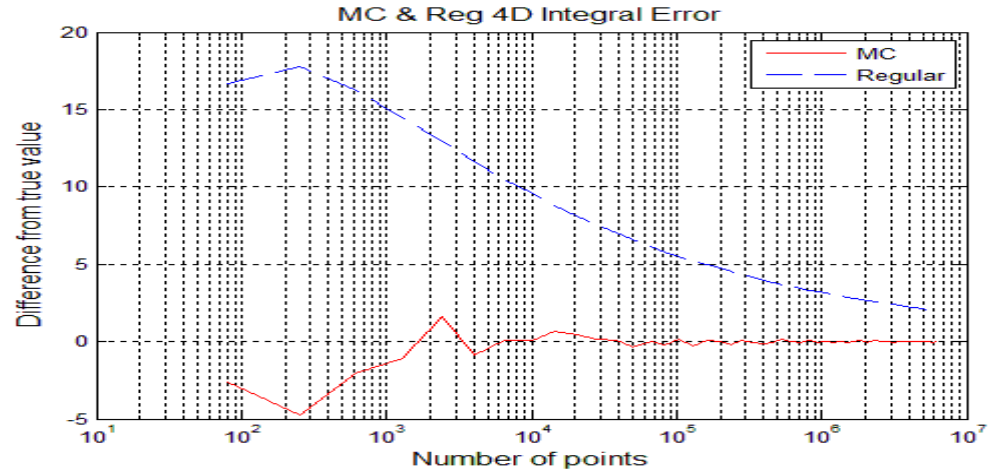
The Monte Carlo error has the property that for high dimensions K, its error is proportional to $1/(\sqrt{N})$ which means that to reduce the error by a factor of 10, the sample size N must be increased by a factor of 100. This effect is independent of the order of magnitude K when the input variables are statistically independent which means that there is no correlation between them. It can be simply illustrated by comparing the performances of regular sampling and Monte Carlo integration for the following integral:

$$I = \int_0^{\pi}\int_0^{\pi}\int_0^{\pi}\ldots\int_0^{\pi} \sin(x_1 + x_2 + x_3 + \ldots + x_K)dx_K dx_{K-1}\ldots dx_2 dx_1 \qquad (4.3)$$

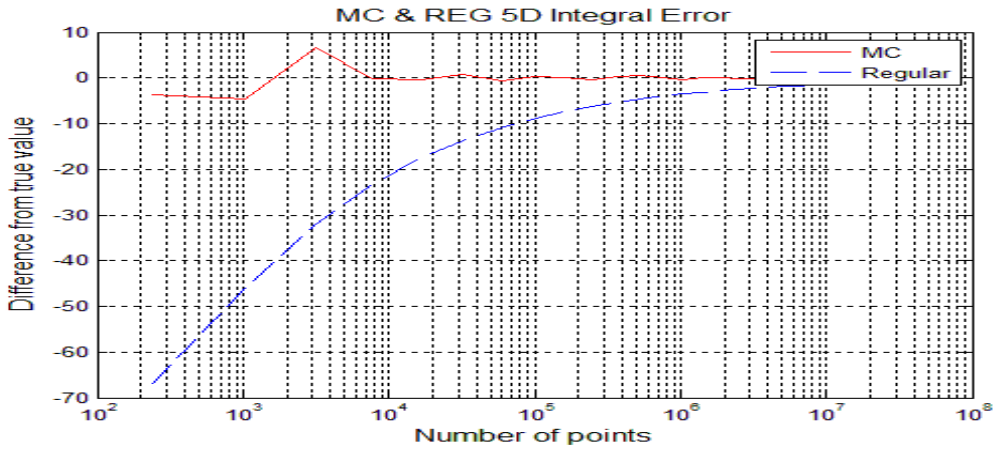
Figures 4.3(a), (b) and (c) show the error for K = 3, 4 and 5 respectively for both regular sampling and uniformly distributed random sampling for a range of values of N. The random sampling has zero correlation from dimension to dimension and from sample to sample. The increasing advantage of independent random sampling over



(a)



(b)



(c)

Figure 4.3: Convergence of MC and regular integration for (a)3D integral, (b) 4D integral, and (c)5D integral.

regular sampling as K increases is clear.

Much research has been devoted to finding ways of decreasing the Monte Carlo error even further to make the technique still more efficient. One approach has been to use variance reduction techniques [95] which are able to reduce the variance of f by transforming it to another function whose integral is the same. Another approach is to use 'quasi-random' or 'low-discrepancy' sampling of the space V. The use of such quasi-random sampling for numerical integration is referred to as "quasi–Monte Carlo" integration. Quasi-random sampling based on 'scrambled nets' [95][21][97][98] [100] has the property that, for 'well behaved' functions, the error becomes proportional to $N^{-3/2}\ \log^{K/2N}$ which is much less than the $1/(\sqrt{N})$ for traditional Monte Carlo integration. Chapters 5 and 6 will discuss quasi–Monte Carlo methods. Other approaches use 'recursive stratified sampling' [105] which breaks down a Monte Carlo calculation into a series of Monte Carlo sub-calculations with feed-back from each stage to decide how best to continue the series. Thus, the calculation is adapted to the application. In the case of integration, the region of integration is progressively divided up into sub-volumes to concentrate the sampling into regions where the variance of the function is largest or of most interest. This makes the sampling more efficient. In the example of the multi-dimensional *sin* function considered above, there would be a higher probability of samples occurring at the edges of the function, where the rate of change is greater, than in the central part. The well-known 'MISER' algorithm of Press and Farrar [101] is based on this approach.

The 'Vegas' Monte Carlo approach of G.P Lepage, inspired by another gambling city, is based on 'importance sampling'. The idea, when used for integration, is to use the probability distribution function (pdf) of the function to determine how sampling points can be concentrated in sub-spaces that produce values of the function that contribute most to the integral. Vegas has been amended to incorporate stratified sampling as well as importance sampling, and it also uses a form of variance reduction in sub-spaces where importance sampling is inappropriate because the sampling turns out to be too sparse [103]. From these well-known references, and many others on Monte Carlo integration, it is clear that much

improvement in efficiency can be gained over the original Monte Carlo approach.

The ideas proposed for integration have inspired similar ideas for efficiency improvement when Monte Carlo techniques are used for simulation, and these prove to be especially valuable for integrated circuit simulation where the dimensionality and complexity is very high.

4.3 Monte Carlo Simulation

As introduced in Section 3.4, Monte Carlo simulation is the application of Monte Carlo methods to study properties of systems having stochastic components. It uses repeated pseudo-random sampling of input variables to determine the behaviour of some physical system as characterized by a computer model. In this thesis, the physical system is an integrated circuit modelled by SPICE, the input variables are component values which are variable due to the uncontrollability of manufacturing effects referred to in earlier chapters, and the behaviour we are interested in may be viability, or otherwise, of the circuit. With repeated sampling used to simulate the fabrication of batches of nominally identical integrated circuits with the specified component variation, the estimation of the probability of a circuit being viable, i.e. that it works, can be considered an estimate of the expected 'yield', i.e. the percentage of working circuits within a batch. The criteria that determine viability are many, including correct logical operation, the power consumption and the propagation delay in the whole circuit or parts of it.

As argued in [95], Monte Carlo simulation can sometimes be formulated in terms of integration. Monte Carlo simulation and Monte Carlo integration may be viewed as two different ways of formulating the same problem. The direct simulation approach provides a more intuitive way of setting up the problem, while transforming it into an integration formulation can be useful when studying theoretical properties of the estimators obtained, especially when variance reduction or quasi–Monte Carlo techniques are used. A comparison of the direct view of simulation versus an integration formulation can be summarised as follows:

The direct simulation approach samples observations of the random vector, $\underline{X}$, of

inputs to and parameters of the simulated system, and for each vector calculates the outputs that are of interest. We thus obtain a random distribution of each output measurement, $f(\underline{X})$ say, which is of interest.

The integration formulation of simulation samples the "source of randomness" $\underline{U}$ as a vector of independent uniformly distributed random numbers. This is transformed into a vector of observations $\underline{X}$ with the appropriate multivariate distribution (e.g. Gaussian) and covariance matrix C by calculating:

$$\underline{X} = A\, g\,(\underline{U}) \tag{4.4}$$

where g is the appropriate 'inverse cumulative distribution function' ('*norminv*' for Gaussian) and A is a matrix such that:

$$A^T A = C \tag{4.5}$$

The covariance matrix C expresses the inter-dependency that exists between different elements of $\underline{X}$ and may be derived from simple assumptions about the effects of proximity as explained in Chapter 2 of [3]. There are many ways of deriving A for a given covariance matrix C, the best known and fastest being Cholesky decomposition, provided as a MATLAB function. In fact, A is not unique and an alternative, preferred in this thesis despite its greater computational complexity, is to calculate:

$$A = \left[\sqrt{\lambda_1}\,\underline{v}_1 \quad \sqrt{\lambda_2}\,\underline{v}_2 \quad \sqrt{\lambda_3}\,\underline{v}_3 \quad \ldots \quad \sqrt{\lambda_N}\,\underline{v}_N\right] = V\Lambda^{1/2} \tag{4.6}$$

where $\lambda_1\ \lambda_2\ \ldots\ \lambda_N$ are the eigenvalues and $\underline{v}_1$, $\underline{v}_2$, …, $\underline{v}_N$ are the corresponding eigenvectors of C. C must be positive semi-definite to be a covariance matrix. V is the matrix of eigenvectors and Λ is the diagonal matrix of eigenvalues as supplied by the MATLAB function:

$$[V\ \ \Lambda] = \mathrm{eig}(C) \tag{4.7}$$

In comparison to the Cholesky decomposition, the eigenvalue/eigenvector method is more intuitive, and a recent paper by J. Keiner and U. Waterhouse [134] gives a more efficient way of performing the same transformation.

Calculating the distribution, between limits, of $f(Ag(\underline{U})) = h(\underline{U})$ say for an output measurement of interest, f say, can now be seen as a process of integration over the range of the multidimensional vector $\underline{U}$. The integration formulation simply re-

expresses the simulation in terms of input vectors, $\underline{U}$, of uncorrelated uniformly distributed random elements which are transformed into input-vectors $\underline{X}$ intended to be representative of the observations expected of true inputs and parameters. This approach has the immediate advantage of being able to model correlation between elements such as may occur intra-die on integrated circuit devices or inter-die from sample to sample within a batch of integrated circuits.

The direct approach is clearly applicable when actual component or parameter measurements are available, or when they have been synthesised for example for the transistor set provided by RandomSPICE [13]. The randomization is then achieved by selecting randomly from the sets of parameters provided. Although a pseudo-random modelling process of physical effects is generated the RandomSPICE parameters, it could have been obtained by direct measurement of real devices. Then a model of the parameter variation is not needed since the true natural physical variation, with its inherent distribution and correlation, will be fed directly into the series of simulations.

However, even when real sets of parameters are available, instead of using them directly, it may be advantageous to produce a model of them as the transformation of a smaller set of independent uniformly distributed random variables. Then the integration form of simulation may be adopted, based on models derived from real data. The models may be derived by employing Principal Components Analysis (PCA) to extract a smaller set of statistically independent parameters that may be transformed back to the complete set with little distortion. The dimensionality may thus be reduced, and each of the independent parameters may be modelled as the transformation of a uniformly distributed random variable. When the independent variables are Gaussian, it is straightforward to model each of them as a transformed uniform random variable. The transformation to Gaussian is achieved by the Gaussian inverse cumulative distribution function (ICDF) available as the function ‘norminv’ provided by MATLAB. Transformations to independent random variables with distributions other than Gaussian may be achieved by replacing ‘norminv’ by a different ICDF, many of which are also available in MATLAB; for example binomial (binoinv), Chi-squared (chi2inv), extreme value - limit distribution (evinv),

exponential (expinv), Gamma (gaminv), Geometric (geoinv), Generalized Pareto (gpinv), Poisson (poissinv), Rayleigh (raylinv) and Student's *t* (tinv) . Multivariate versions of these functions, allowing correlation to be introduced, are also widely available. With the different distributions, essentially the same methodology with respect to statistical static delay estimation as used with Gaussian, Pareto and low discrepancy sequences in this thesis, can remain valid. However this is not trivial, and is beyond the scope of this thesis.

Apart from the likely reduction in dimensionality, this transformation of a direct simulation approach to an integration-like formulation allows a much larger set of randomised devices to be generated than are available in the original set. Hence more simulations may be run with different sets of parameters based on parameters obtained by measuring real devices.

Applying this approach to the transistor parameter sets provided by RandomSPICE allowed the number of device parameters to be drastically reduced without loss of accuracy, and allowed the restriction of having just 201 examples of each device to be overcome. This application turned out to be less impressive than it could have been due to the device parameters being already model-based rather than true measurements. However, the principle of the approach is demonstrated.

Viewed in either formulation, Monte Carlo simulation samples the probability distribution of all the input variables and system parameters to produce many repeated versions of the system. These are in turn analysed to determine how certain key output measurements vary due to the input variability. A histogram of each key output measurement gives an estimate of its likely distribution, the estimate becoming more and more reliable as the number of simulations increases. Since the number of simulations must be restricted for practical reasons, the accuracy of these results is also limited by practicality. Where valid assumptions can be made about the shape of the distribution, for example that it is Gaussian or Chi-squared, a maximum likelihood fit to the histogram can be made on such assumptions.

Such a fit would produce a value of mean and variance thus allowing a pdf as shown below to be drawn. Assuming the measurement to be a delay that is required to be less than D for viability, and a Gaussian distribution, the yield of the circuit can

now be estimated as the area under the pdf 'tail' from D to infinity which may be derived from the 'complementary error function' evaluated at D.

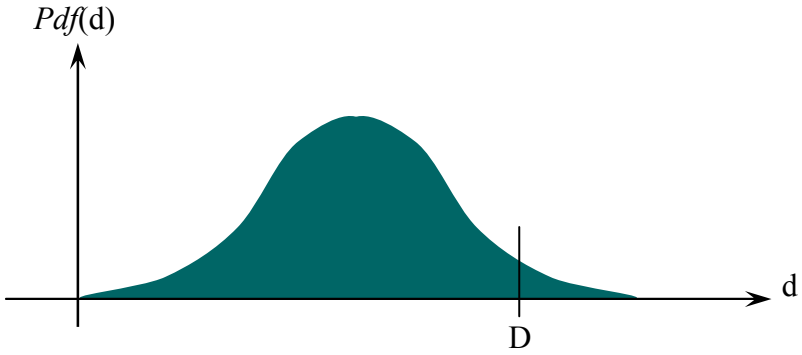


Figure 4.4: Gaussian probability density function of a circuit propagation delay and delay threshold D

The accuracy of this estimation will depend on the number of simulations which must be limited. Unfortunately, the effect of the limitation will be most serious over the tails of the distribution, which is the part we are most interested in. For example, for a Gaussian pdf, the probability of being more than three standard deviations greater than the mean is $Q(3) = 0.5 \times erfc(3/\sqrt{2})$ where *erfc* is the traditional Gaussian 'complementary error function'. $Q(3)$ is about 0.0013 meaning that if 1000 circuits are simulated we can only expect to find one value of the key output in this region. There would need to be many more than one value to have a chance of reasonable accuracy. This illustrates the need for complexity reduction techniques when applying Monte Carlo simulation to Integrated Circuit variability, and such techniques will be considered in the next Chapter.

In general, Monte Carlo methods proceed as follows:

1. The characteristics of the input vectors are determined.
2. Random vectors are generated with appropriate distributions and inter-correlation either directly or by transforming independent uniformly distributed random vectors.
3. A deterministic computation is performed to simulate the behaviour of the system

for each of the randomized input-vectors.

4. For each key output measurement, its pdf is estimated in the best way possible, given the inevitable limitations in the amount of data available.
5. Deductions about the probability of certain events are made from the estimated pdf.

4.4 Monte Carlo Simulation Applied to Integrated Circuits

The procedure of an IC's Monte Carlo simulation is demonstrated in Figure 4.5, where the circuit used is a C-element mentioned in Chapter 2 and the output delay is analysed. The procedure described is as follows:

1. Find statistical distribution for each parameter.
2. Sample statistical process to produce a value for each parameter.
3. Parameterize one circuit and simulate it.
4. Repeat for many copies of circuit and obtain the statistical distribution of a specific measurement.

4.4.1 Using HSPICE Directly

As mentioned in Section 3.7.2, Randomisation is possible using the HSPICE package itself. For statistical MC simulation using HSPICE directly, an appropriate model for each of the components in a particular processing technology is ideally formed by the foundry. These models include the anticipated statistical distribution of different important technological parameters for each component. For a MOSFET these parameters include the threshold voltage, the channel-width, channel-length and oxide thickness. Since the anticipated variations in these parameters originate from many random sources, the Central Limit Theorem predicts that the variations will have close to a Gaussian distribution. Only a mean and standard deviation (σ) is then required to characterise each element of variation. When designing circuits for a given technology, the designer often specifies the mean value for a certain parameter for which the manufacturer gives the standard deviation. For the research of this thesis, such support for 35nm MOSFET technology had not been released by any

foundry, therefore the statistics of the devices from the 35nm MOSFET model set included in RandomSPICE [13] was vital.

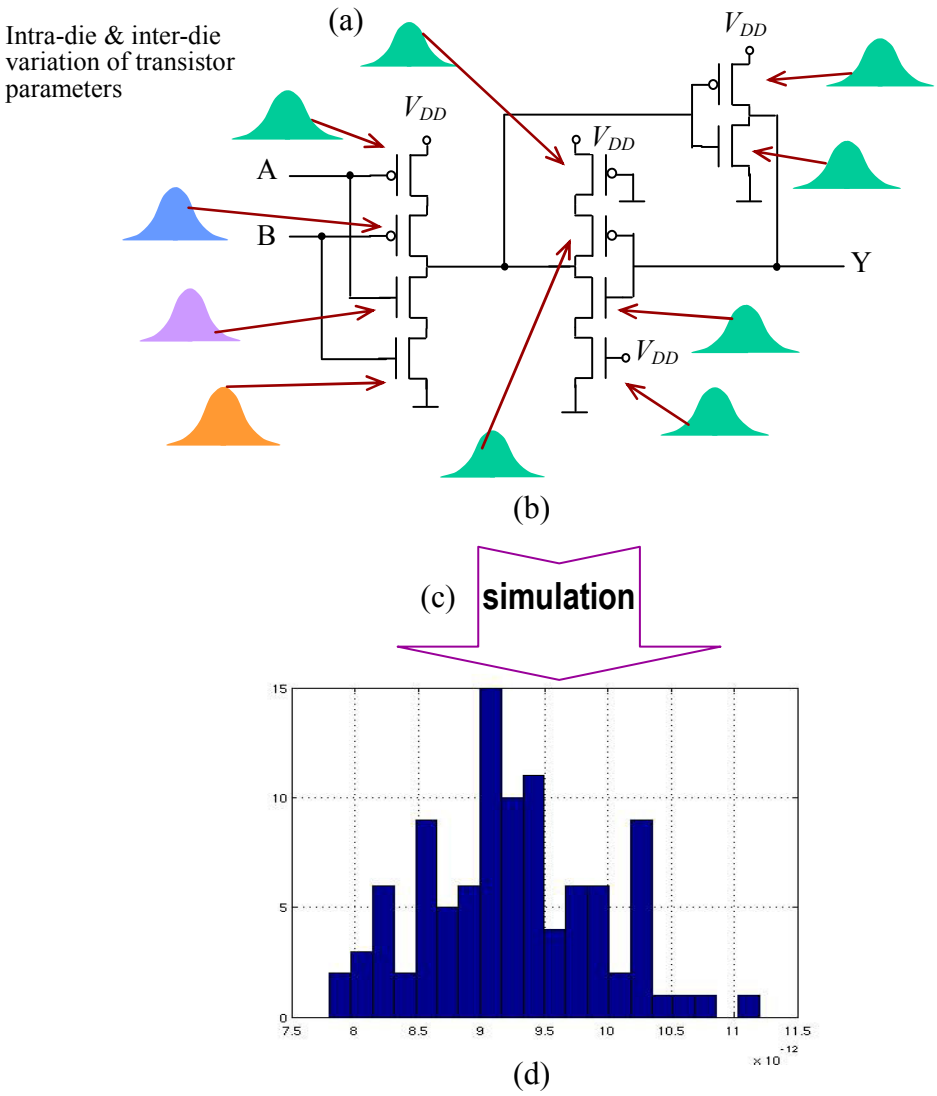


Figure 4.5: Transistor level MC simulation to C-element circuit
(a) statistical distribution for each parameter, (b) sampling parameter value, (c) circuit simulation, (d) obtain output delay statistical distribution.

Monte Carlo analysis has been available in HSPICE for some time and is based on two approaches:

(1) defining random variables (with specified distribution, mean and standard deviation) as global parameters within an HSPICE netlist
(2) defining random variables (with specified distribution, mean and standard deviation) as model parameters when constructing 'model files' for the required devices

The Variation Block concept introduced in Section 3.7.2 allows variation modelling to be introduced based on Principal Components. Independent random variables A1, A2, … may be introduced and used as principal components. Other variations may then be defined as functions of these principal components [31].

4.4.2 Using NGSPICE

HSPICE offers an approach to MC simulation that is professionally designed and well adapted to the demands of commercial manufacturers and circuit design companies. However it is less suited to the demands of the current research project because it implements proprietary known approaches and does not have the flexibility needed to investigate research ideas. Specifically it performs the MC simulations in way that does not allow ideas such as 'Statistical Blockade' (to be introduced in the next chapter) to be conveniently implemented. Also, it is normally available on a single machine, with a fixed license. Its latest version can employ the facilities of a multi-core processor, but as a predefined commercial product rather than a research tool. Hence the investigation of parallel and distributed versions of Monte Carlo circuit analysis envisaged here is not currently possible, and would likely be very expensive to set up.

In this thesis, a Monte Carlo analysis harness was implemented as a MATLAB program that makes repeated calls to the HSPICE circuit simulation package, but does not rely on any of its MC analysis facilities. An immediate benefit of this approach, apart from its flexibility as a research tool, is that the open source version of SPICE, known as NGSPICE, may be used in place of the commercial version HSPICE.

As introduced in Section 3.7.3, NGSPICE is a mixed-signal (analogue and

digital) circuit simulator combining three open source software packages: SPICE3, Cider and Xspice. NGSPICE is under continuing development as part of the 'gEDA' project for developing a full GNU Public Licensed suite of electronic circuit design (EDA) tools. SPICE3 has become the most popular engine for circuit simulation since the invention of SPICE at the University of Berkeley California around 1970. This same engine is the basis of many different versions of SPICE including HSPICE which has been extensively augmented with features for commercial use. Cider introduces highly developed and accurate device simulation to SPICE3 based on the use of 'BSIM' device models [45]. XSPICE augments SPICE3 by providing code modelling support and the simulation of digital components through event-driven 'finite state machines'. Many people have contributed and are still contributing to this project and contributions are always invited. The gEDA project has already produced and is still developing a full suite of EDA tools for electrical circuit design, schematic capture, simulation, prototyping, and production.

The analysis features of NGSPICE, though not as comprehensive as HSPICE, are adequate for the current research and distributing its capability to fellow researchers. It is possible that some of the ideas thus investigated will influence the further development of both NGSPICE and maybe HSPICE also. The version of NGSPICE used in this thesis did not include any MC analysis or randomisation facilities, though the very latest version NGSPICE23 [37] has now introduced some rudimentary features that may prove useful in future.

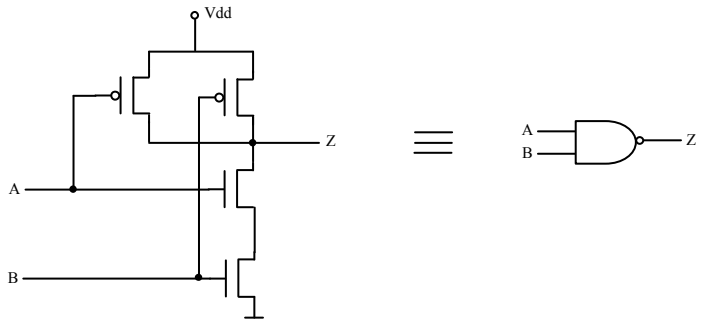
4.4.3 Using RandomSPICE

The origins of the approach presented in this thesis lie in RandomSPICE [13] that produces SPICE netlists for randomised copies of a given circuit or sub-circuit. The randomization is achieved by selecting nmos and pmos transistor models at random from a set of about 201 of each type produced by Glasgow University [13]. RandomSPICE produces a specified number of circuit files, and there normally have to be very many. To make use of RandomSPICE, it was necessary to design a harness for applying SPICE to simulate the files one by one, extract the required

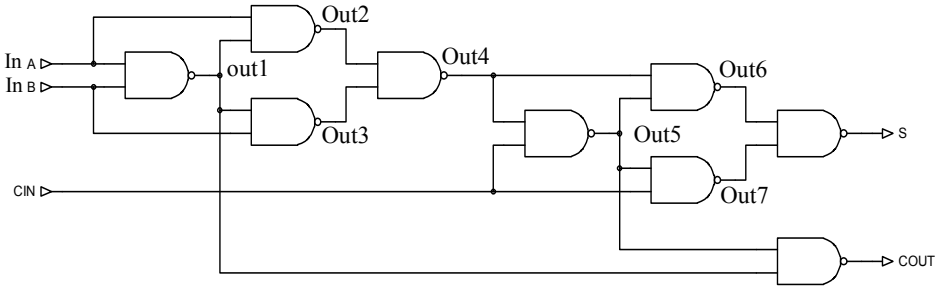
information from the SPICE output files produced, collate the results, present the data in a suitable form, and finally extract statistical information from the data. Some useful preliminary results were obtained by this method such as the graphs shown below for a 'binary full adder' circuit, showing the effect of the predicted variation in the transistor parameters. A deficiency in the early version of RandomSPICE, where devices were not truly randomised for sub-circuits that are repeatedly called in a certain circuit, led to the need for an alternative approach.

A Binary Full Adder (BFA) circuit is used here to demonstrate MC circuit simulation based on the use of RandomSPICE. The BFA circuit is shown as figure 4.6(b), with the hierarchy from transistor to logic gate shown in Figure 4.6(a).

The circuit netlist, before randomization with RandomSPICE, is presented in Table 4.1. In this netlist, an identical NMOS model with name 'atomn' and an identical PMOS model with name 'atomp' are used for the four transistors (two NMOS and two PMOS) in each NAND gate as defined by the sub-circuit NAND2x1:RAND. This sub-circuit is used nine times identically in the BFA circuit.



(a) NAND gate - building block, transistor circuit and symbol



(b) BFA circuit represented with NAND gates

Figure 4.6: Binary Full Adder (BFA) circuit

```
*.GLOBAL vdd
.SUBCKT NAND2X1:RAND Z A B vdd
MMN1 Z A net3 0 atomn L=35e-9 W=70e-9
MMN2 net3 B 0 0 atomn L=35e-9 W=70e-9
MMP1 Z B vdd vdd atomp L=35e-9 W=70e-9
MMP2 Z A vdd vdd atomp L=35e-9 W=70e-9
.ENDS NAND2X1:RAND

.SUBCKT BFA InA InB CIN S COUT vdd
XI1 OUT1 InA InB vdd NAND2x1:RAND
XI2 OUT2 InA OUT1 vdd NAND2x1:RAND
XI3 OUT3 InB OUT1 vdd NAND2X1:RAND
XI4 OUT4 OUT2 OUT3 vdd NAND2X1:RAND
XI5 OUT5 OUT4 CIN vdd NAND2x1:RAND
XI6 OUT6 OUT4 OUT5 vdd NAND2x1:RAND
XI7 OUT7 OUT5 CIN vdd NAND2x1:RAND
XI8 S OUT6 OUT7 vdd NAND2x1:RAND
XI9 COUT OUT1 OUT5 vdd NAND2X1:RAND
.ENDS BFA

XBFA InA InB CIN S COUT vdd BFA
Vdd vdd 0 1.2
VA InA 0 1.2
VB InB 0 0
Vin CIN 0 0  PULSE(0 1.2 0.05n 0.1p 0.1p 0.1n 0.2n)
.TRAN 0.00002n 0.2n
.PRINT TRAN V(CIN) V(S) V(COUT)
*.PLOT V(CIN) V(S) V(COUT)
.END
```

Table 4.1: Netlist for BFA circuit BFA_1.cir

After running RandomSPICE for this BFA netlist, a number of randomised copies is obtained with all the transistors given randomised model parameters. Each randomised copy is titled differently as BFA_1.cir, BFA_2.cir, … The netlist for one of the randomised circuit copies is shown in Table 4.2. Within each copy, each NAND gate is given a different name (NAND2x1_1, NAND2x1_2 to NAND2x1_9) and, within each NAND gate, each transistor model is selected at random from the 201 of each type that are available. The NMOS transistor models are labelled NCH0, NCH1, …, NCH200 and the PMOS transistors are labelled PCH0, PCH1, …, PCH201. For example, NCH80 & NCH85 are chosen at random for the two square NMOS transistors in the sub-circuit NAND2x1_1 within BFA_1.net. Different NMOS transistors will be chosen for NAND2x1_2 within BFA_1.net, and for all other NAND sub-circuits within this copy of BFA. For the second BFA copy, the

randomization process is repeated but with different choices of NMOS transistor model in all cases.

Since the model database only provides square BSIM4 transistors, RandomSPICE is required to split any non-square transistors into several square ones. For example, in the netlist, the PMOS transistor has a width that is twice its length and RandomSPICE splits it into two PMOS transistors with same size of 35nm technology for the width and length, and gives them different model parameters.

```
BFA_1.cir
***BINARY FULLADDER (BFA) HSPICE-RandomSPICE netlist
* Random seed: 1234789535
***Zheng
*.Global vdd
.SUBCKT BFA_1 InA InB CIN S COUT vdd
XI1 OUT1 InA InB vdd NAND2x1_1
XI2 OUT2 InA OUT1 vdd NAND2x1_2
XI3 OUT3 InB OUT1 vdd NAND2X1_3
XI4 OUT4 OUT2 OUT3 vdd NAND2X1_4
XI5 OUT5 OUT4 CIN vdd NAND2x1_5
XI6 OUT6 OUT4 OUT5 vdd NAND2x1_6
XI7 OUT7 OUT5 CIN vdd NAND2x1_7
XI8 S OUT6 OUT7 vdd NAND2x1_8
XI9 COUT OUT1 OUT5 vdd NAND2X1_9
.ENDS BFA_1

XBFA_1 InA InB CIN S COUT vdd BFA
Vdd vdd 0 1.2
VA InA 0 1.2
VB InB 0 0
Vin CIN 0 0  PULSE(0 1.2 0.05n 0.1p 0.1p 0.1n 0.2n)
.TRAN 0.00002n 0.2n
.PRINT TRAN V(CIN) V(S) V(COUT)
*.PLOT V(CIN) V(S) V(COUT)
.PROBE
.OPTION POST
```

Table 4.2: Randomised netlist for BFA_1.cir

For the non-square PMOS transistors within each NAND gate within each BFA copy, a sub-circuit consisting of two square PMOS transistors is required, each being chosen at random from the choice of PMOS models available. A netlist for one of the nine NAND sub-circuits defined for the first randomized circuit copy, BFA_1.cir, is

shown in Table 4.3.

Any number of such netlists could be generated according to the requirement of simulation accuracy. For each BSIM4 transistor model, there are up to 300 parameters which must be provided to SPICE by a '.MODEL NCHXXX NMOS' or '.MODEL PCHXXX PMOS' statement where XXX denotes the NMOS or PMOS randomized model number.

```
.SUBCKT NAND2x1_1 Z A B vdd
MMN1 Z A net3 0 NCH80 L=3.5e-08 W=3.5e-08
MMN2 net3 B 0 0 NCH85 L=3.5e-08 W=3.5e-08
XMMP1 Z B vdd vdd SUBMMP1
XMMP2 Z A vdd vdd SUBMMP2
*DD13 B vdd DP 1.02e-13
*DD15 0 A DN 1.02e-13
.SUBCKT SUBMMP1 SUBMMP1_0 SUBMMP1_1 SUBMMP1_2 SUBMMP1_3
M_SUBMMP1_1 SUBMMP1_0 SUBMMP1_1 SUBMMP1_2 SUBMMP1_3 PCH63 L=3.5e-08
W=3.5e-08
M_SUBMMP1_2 SUBMMP1_0 SUBMMP1_1 SUBMMP1_2 SUBMMP1_3 PCH40 L=3.5e-08
W=3.5e-08
.ENDS SUBMMP1
.SUBCKT SUBMMP2 SUBMMP2_0 SUBMMP2_1 SUBMMP2_2 SUBMMP2_3
M_SUBMMP2_1 SUBMMP2_0 SUBMMP2_1 SUBMMP2_2 SUBMMP2_3 PCH121 L=3.5e-08
W=3.5e-08
M_SUBMMP2_2 SUBMMP2_0 SUBMMP2_1 SUBMMP2_2 SUBMMP2_3 PCH77 L=3.5e-08
W=3.5e-08
.ENDS SUBMMP2
.ENDS NAND2x1_1
```

Table 4.3: Netlist for NAND2x1_1 within BFA_1.cir
(uses models NCH80 & NCH85, PCH63,PCH40,PCH121 & PCH77)

By executing the MATLAB harness developed in Manchester University for RandomSPICE, the required series of Monte Carlo simulations may be executed and the results can be statistically analysed.

In table 4.2, the netlist statement:

```
Vin CIN 0 0  PULSE(0 1.2 0.05n 0.1p 0.1p 0.1n 0.2n)
```

specifies the parameters of an input pulse applied to the 'carry in' port, and the statement:

```
.PRINT TRAN V(CIN) V(S) V(COUT)
```

specifies the type of analysis and the output voltages that are required to be statistically analysed.

The graphs in figure 4.7 show the statistical variation that occurs to the BFA 'carry out' signal over 100 RandomSPICE-randomised circuits. Figure 4.7 (a) plots the series of waveforms produced. Figure 4.7(b) plots the zoomed rising edge waveforms which are used for analysing the delay (propagation) time of the carry-out signal. Figure 4.7 (c) is the distribution histogram of the delay time. Figure 4.7(d) is the result of fitting a Gaussian pdf to the delay distribution, from which can be obtained a mean and standard deviation, and thus an error function specifying the probability of the delay exceeding a given value.

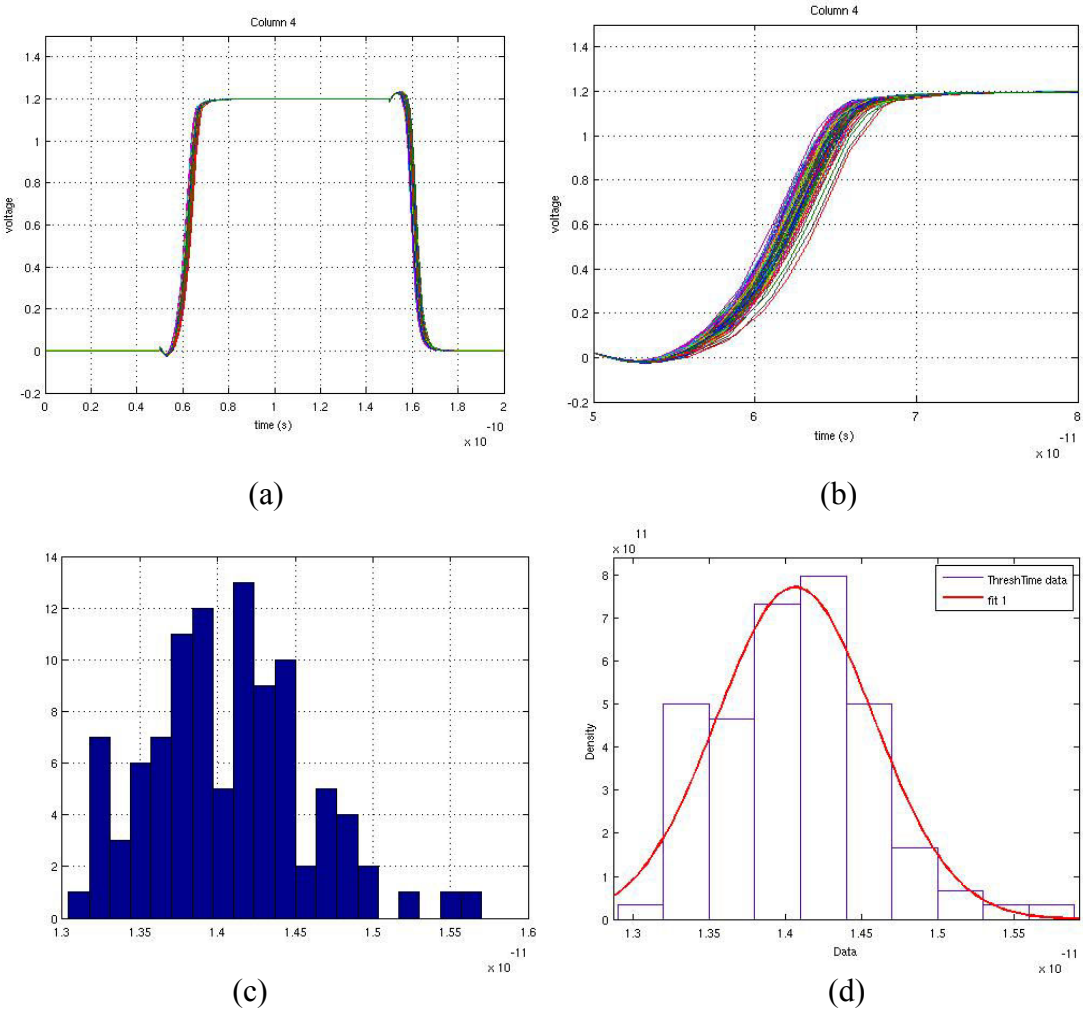


Figure 4.7: MC simulation results for delay time of carry out signal in BFA circuit

The histograms used in this thesis are from two sources: firstly, a standard histogram function provided by MATLAB which allows the number of bins to be specified, and secondly, the statistical analysis package 'dfittool', part of the MATLAB statistical analysis toolbox, which varies the number of bins according to the number of data points and the nature of the distribution. For the standard histograms in the thesis, e.g. in figure 4.7 (c), the number of bins is always fixed at twenty. The 'dfittool' does not allow the number of bins to be selected by the programmer. Hence, with 'dfittool' the number of bins will be variable and therefore different from the twenty bins used elsewhere. There is no loss of accuracy caused by the difference in the number of bins.

4.4.4 Using a New Harness

A new MATLAB harness, named 'RandomLA' (Random LSI circuit Analysis) was designed to test improved MC analysis approaches and to be self-contained and suitable for parallel or distributed implementation to analyse very large integrated circuits. MATLAB now offers very convenient and powerful facilities for parallel processing on multi-core machines and distributed processing on clusters of 'worker' machines each capable of running NGSPICE. The use of the 'Condor' distributed computing package [106] is ideal for this application, since the large number of randomized circuits that are generated by a host running the Harness can be sent to any machine capable of running a version of SPICE and conveying the manageable amount of data produced back to the host once each simulation is complete.

To perform a Monte Carlo simulation using the facilities developed for this thesis, the Harness requires a "seed" circuit to be defined as an 'augmented netlist' The seed is a normal netlist with two very simple innovations. Firstly, any parameter or input variable may be replaced by a random variable whose distribution and statistical parameters (mean, standard deviation) are given in place of the actual circuit parameter. Parameters may be resistor values, capacitor values, transistor parameters, input waveform specifications such as rise and fall times, and there are other possibilities. Secondly, any circuit parameter can be specified as involving some combination of other parameters.

The Harness generates R copies of the seed circuit each with randomized parameters replacing the statistical parameters, but calculated according to their specification. Assuming there are N parameters to be randomized, it is convenient to refer to the parameters for the circuits as:

$$\{x_{11}, x_{12},..., x_{1N}\}$$
$$\{x_{21}, x_{22},..., x_{2N}\}$$
$$....$$
$$\{x_{R1}, x_{R2},..., x_{RN}\}$$

and thus define an R *by* N matrix X_{MAT}. These circuits may be generated one by one in a single core implementation or in batches of an appropriate number for a parallel or distributed version. There is no need for a large reservoir of randomized circuits as produced by RandomSPICE. The Harness initiates the required executions of SPICE, sends them the appropriate randomised netlists and receives the outputs when they become available.

Assume for each of R randomised circuits we require one measurement, of delay say, and that these are referred to as $\{D_1, D_2, ..., D_R\}$. These measurements must be extracted from the numerical SPICE outputs by parsing them using 'regular expressions' available in MATLAB. Given sufficient circuits, it may now be possible to derive reasonable statistical estimates from the measurements. If a particular distribution, such as Gaussian or Chi-squared can be assumed, the assumed distribution can be fitted to the data in a maximum likelihood manner and inferences can then be drawn from the distribution. In the case of delay, assuming it appears reasonable to fit a Gaussian pdf to the measurements, its mean and standard deviation may then be deduced. Likely to be of most interest will be the tail of this distribution, and this is certainly the case if the application is to estimate the percentage of circuits for which the delay is likely to exceed some threshold. The user must decide where to define start of the 'tail', normally in standard deviations above the mean. The Harness can now calculate the probability of a measurement being in the tail (using the Normal 'error function' *erfc*) and likely to make the circuits non-viable. Hence, the likely percentage of failures, or 'yield' may be estimated.

The simple mechanism of a 'seed' circuit is illustrated by the example in Table 4.4. The value of capacitor C1 is specified as an independent Gaussian random variable (default) with mean and standard deviation equal to 4.8×10^{-15} and 24.4×10^{-15} Farads, respectively.

```
.SUBCKT SWNAND2 Y A B VDD
SWP1 VDD X VDD A SW OFF
SWP2 VDD X VDD B SW OFF
SWN1 X ND1 A 0 SW OFF
SWN2 ND1 0 B 0 SW OFF
R1 X Y 1K
C1 Y 0 [[4.8e-15, 2.44e-16]]
```

Table 4.4 A 'seed' netlist with randomisation of capacitor C1

If the final line in the above illustration is replaced by:

- C1 Y 0 [[4.8e-15, 0.244e-15]+ [3.5*C2]+[1.4*C3] …]

then the random value of C1 is defined as the sum of three or more random variables. The first one is independent as before. The second one is a constant times the value of a different component C2 Similarly for the third one, and there may be as many as required. Randomising device parameters is also straightforward and requires the instantiation of devices in the seed to refer to a suitable model with one or more parameters replaced by a mean and standard deviation as for the capacitor C1 value in Table 4.4

When processing the seed, the Harness produces as many randomised versions of the device parameters as there are instantiations, and each is given an index. Therefore, for the BFA circuit in figure 4.4, an instantiation of a model NCH within the seed:

MMN1 Z A net3 0 NCH L=3.5e-08 W=7.0e-08

would be replaced by:

MMN1 Z A net3 0 NCHxx L=3.5e-08 W=7.0e-08

where xx is an index number, and a randomized version of the '.model' NCH would be produced and labeled NCHxx. Spatially dependent correlation may be introduced by associating devices with particular (x,y) locations on the chip. The degree of correlation between the parameters of these devices can now be made dependent on

the distances between the devices.

It is useful to consider how the BFA circuit considered in Section 4.4.2 above can now be analysed using the new harness. The seed circuit is, in this case, identical to the original netlist presented in Table 4.1, except that one or more transistor parameters are given statistical characteristics within [[…]] brackets, rather than direct values. The statistical characteristics for the distribution, mean and standard deviation for each parameter of each model were, for this example, obtained by analysing the variation of each transistor parameter in each of the two Toshiba (PMOS and NMOS) data-sets provided by Glasgow University. Correlation is not included in this example, and is better illustrated for behavioural modelling as introduced in Chapter 5. The randomisation process applied to the seed will generate the required randomised models and appropriate indexing for the instantiations.

Observe that a .MODEL parameter list is generated for each randomized transistor which generates a lot of data when there are many transistors each with many parameters. The results of the simulation specified above are given in figure 4.8. They demonstrate that the harness works, and is a useful vehicle for the investigations which follow in the next chapters.

```
.SUBCKT NAND2X1X1 Z A B vdd
MMN1 Z A net3 0 NCH01 L=3.5e-08 W=3.5e-08
MMN2 net3 B 0 0 NCH02 L=3.5e-08 W=3.5e-08
MMP1 Z B vdd vdd PCH03 L=3.5e-08 W=3.5e-08
MMP2 Z A vdd vdd PCH04 L=3.5e-08 W=3.5e-08
.ENDS NAND2X1X1
        …
.SUBCKT NAND2X1X9 Z A B vdd
MMN1 Z A net3 0 NCH33 L=3.5e-08 W=3.5e-08
MMN2 net3 B 0 0 NCH34 L=3.5e-08 W=3.5e-08
MMP1 Z B vdd vdd PCH35 L=3.5e-08 W=3.5e-08
MMP2 Z A vdd vdd PCH36 L=3.5e-08 W=3.5e-08
.ENDS NAND2X1X9
.SUBCKT BFA InA InB CIN S COUT vdd
XI1 OUT1 InA InB vdd NAND2X1X1
XI2 OUT2 InA OUT1 vdd NAND2X1X2
XI3 OUT3 InB OUT1 vdd NAND2X1X3
XI4 OUT4 OUT2 OUT3 vdd NAND2X1X4
XI5 OUT5 OUT4 CIN vdd NAND2X1X5
XI6 OUT6 OUT4 OUT5 vdd NAND2X1X6
XI7 OUT7 OUT5 CIN vdd NAND2X1X7
XI8 S OUT6 OUT7 vdd NAND2X1X8
XI9 COUT OUT1 OUT5 vdd NAND2X1X9
.ENDS BFA
```

```
XBFA InA InB CIN S COUT vdd BFA
Vdd vdd 0 1.2
VA InA 0 1.2
VB InB 0 0
Vin CIN 0 0  PULSE(0 1.2 0.05n 0.1p 0.1p 0.1n 0.2n)
.TRAN 0.0005n 0.23n
.PRINT TRAN V(CIN) V(COUT)
```

Table 4.5(a): Netlist from seed in Table 4.1 with randomisation of transistor models

```
.MODEL NCH01 NMOS
...
+vth0     = [[2.2870E-01, 1.0E-02]]
...
.MODEL NCH02 NMOS
...
+vth0     = [[2.2870E-01, 1.0E-02]]
…
.MODEL PCH01 PMOS
...
+vth0     = [[-2.2870E-01, 1.0E-02]]
.
.MODEL PCH01 PMOS
...
+vth0     = [[-2.2870E-01, 1.0E-02]]
...
.END
```

Table 4.5(b): Randomised transistor models showing randomized parameter ‘vth0’

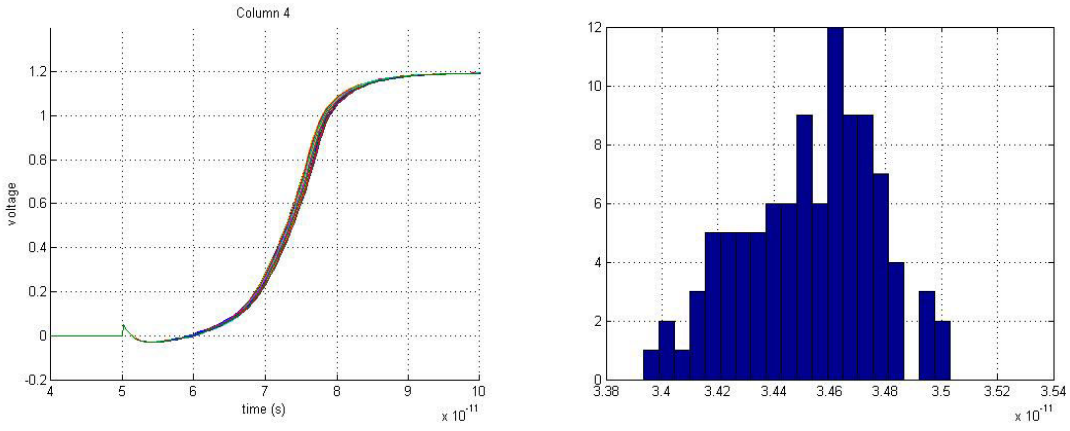


Figure 4.8: MC simulation results for delay time of carry out signal in BFA circuit with randomisation of transistor parameters

4.5 Introducing intra-die correlation

This section describes how correlation may be introduced into the random variables generated by the RandomLA software as the parameters for the randomised circuit copies. The correlation may be due to the intra-die proximity of components, or other causes. The parameters may be values of components such as resistors and capacitors, they may be device parameters, or they may be elements of principal component vectors which are ultimately transformed to device parameters. Define a matrix X_{mat} of 'column' vectors $\underline{X}_j$, each containing M parameters:

$$X_{mat} = \left[\underline{X}_1 \quad \underline{X}_2 \quad \underline{X}_3 \quad \ldots \quad \underline{X}_R\right] \tag{4.8}$$

where

$$\underline{X}_j = \begin{bmatrix} x(1,j) \\ x(2,j) \\ x(3,j) \\ \vdots \\ x(M,j) \end{bmatrix} \tag{4.9}$$

Each column-vector, $\underline{X}_j$, of parameters is for a different randomised circuit copy. Within X_{mat}, there is row-to-row correlation due to inter-dependencies between devices or the parameters of individual devices. Column-to-column correlation is not of interest since it is assumed that the random vector generated for each circuit is independent of all others as required for efficient MC analysis. (In the software 'RandomLA' rows and columns are interchanged, therefore it is column-to-column correlation that is calculated).

To model intra-die correlation between parameters due to proximity, let parameter i be defined for a point $P_i = (a_i, b_i)$ on the chip. So $P_1 = (a_1,b_1)$ is for parameter 1, $P_2 = (a_2,b_2)$ is for parameter 2, and so on. Note that a_i is the measurement along the x-axis and b_i is the measurement along the y-axis for each point P_i. The Euclidean distance between any two points P_i and P_m , in units of nano-meters, is:

$$d(P_i, P_m) = \sqrt{(a_i - a_m)^2 + (b_i - b_m)^2} \qquad \text{for any } i \text{ and } m \qquad (4.10)$$

The 'exponential model' of intra-die correlation defined in the paper by B. Hargreaves, H. Hult and S. Reda [84], models the correlation $k(i,m)$ between parameters i and m as:

$$k(i,m) = exp[-\lambda d(P_i,P_m)]$$

that is:

$$k(i,m) = e^{-\lambda d(Pi,Pm)} \qquad (4.11)$$

This is assumed to be the Pearson correlation coefficient defined as:

$$k(i,m) = \frac{\text{cov}(row_i, row_m)}{\sigma_i \sigma_m} \qquad (4.12)$$

where σ_i = standard deviation of parameter i , σ_m = standard deviation of parameter m, and '*cov*' means covariance. This means that:

$$\sigma_i = \sqrt{\text{cov}(row_i, row_i)} \quad \text{and} \quad \sigma_m = \sqrt{\text{cov}(row_m, row_m)} \qquad (4.13)$$

A value for *lambda* (λ) may be calculated from measurements of actual device parameters, or manufacturer's data. Clearly $k(i,i)$ =1 since the Euclidean distance $d(Pi,Pi)$ between Pi and Pi (the same point) is zero. If, in 35 nm technology, it may be assumed that at a distance of 200 nm the correlation reduces to some small value, α say,

$$exp(-\lambda.200) = \alpha \qquad (4.14)$$

it follows that

$$\lambda = -(1/200)\log_e(\alpha) \qquad (4.15)$$

Assuming $\alpha = 0.01$, we find that:

$$\lambda = 4.61/200 = 0.023 \tag{4.16}$$

Therefore, if we have a location $P_i = (a_i, b_i)$ on the chip for each element i, a value of correlation $k(i,m)$ between rows i and m may be modelled. Hence, we obtain a 'row-to-row' (parameter-to-parameter) correlation matrix:

$$K = \begin{bmatrix} 1 & k(1,2) & \cdots & k(1,R) \\ k(2,1) & 1 & \cdots & k(2,R) \\ \vdots & \vdots & \ddots & \vdots \\ k(R,1) & k(R,2) & \cdots & 1 \end{bmatrix} \tag{4.17}$$

To provide an example, assume that there are four capacitors at P_1, P_2, P_3 and P_4 whose means and std-deviation are specified in the seed file ([[…]]) as m_1, m_2, m_3, m_4 and s_1, s_2, s_3, s_4 respectively. Let:

$$\underline{m} = \begin{bmatrix} m_1 \\ m_2 \\ m_3 \\ m_4 \end{bmatrix} \quad \text{and} \quad \underline{s} = \begin{bmatrix} s_1 \\ s_2 \\ s_3 \\ s_4 \end{bmatrix} \tag{4.18}$$

We need to generate, for each Monte Carlo randomised circuit j, a random variable $x(1, j)$, $x(2, j)$, $x(3, j)$ and $x(4, j)$ as the value for each capacitor to obtain the parameter vector:

$$\underline{X}_j = \begin{bmatrix} x(1,j) \\ x(2,j) \\ x(3,j) \\ x(4,j) \end{bmatrix} \tag{4.19}$$

To generate random variables for each capacitor with the right amount of intra-die correlation, we first need to deduce, from K, an appropriate row-to-row covariance matrix, which is as follows:

$$C = [\, c(i,m) \,] \quad \text{where} \quad c(i,m) = k(i,m) \times s_i \times s_m \tag{4.20}$$

Then find matrix A such that $A^T.A = C$ using the 'Choleski Decomposition' or the 'eigenvector-eigenvalue method' described in Section 4.3 of this thesis. Finally, generate a multivariate (4-element) correctly correlated Gaussian random vector as follows:

$$\underline{X}_j = \underline{m} + A.\underline{r} \tag{4.21}$$

where $\underline{r}$ is a 4 by 1 vector of independent pseudo-random Gaussian variables of zero mean and unit variance. Either of the following MATLAB statements can generate the vector $\underline{r}$:

$$\underline{r} = randn(4,1) \quad \text{or} \quad \underline{r} = norminv(rand(4,1)) \tag{4.22}$$

We can do this repeatedly for each randomised circuit j, producing many column-vectors $\underline{X}_j$.

Introducing intra-die variability into the parameters of transistor devices is possible using the same approach as used for component values. In such applications, it is useful to partition a large correlation matrix into a number of smaller ones, each catering for one type of parameter or principal component. This is possible when there can be assumed no correlation between the different types or principal components; of course this is guaranteed for principal components as will be seen in Chapter 5. Clearly, a different value of λ can be used for each partition.

Note that the unit of λ depends on that of the distance measure $d(P_i,P_j)$ which, in this thesis, is always nanometers (nm). The approach outlined above works for $\lambda = 0$

which is the case where all parameters of a certain type are assumed to be 100% correlated. This can be useful for testing theories about the effects of intra-die correlation.

4.5.1 Results from introducing intra-die correlation into a CMOS NAND gate

Consider the CMOS NAND gate shown in Figure 4.6(a) whose on-chip layout is assumed to be as shown in Figure 4.9.

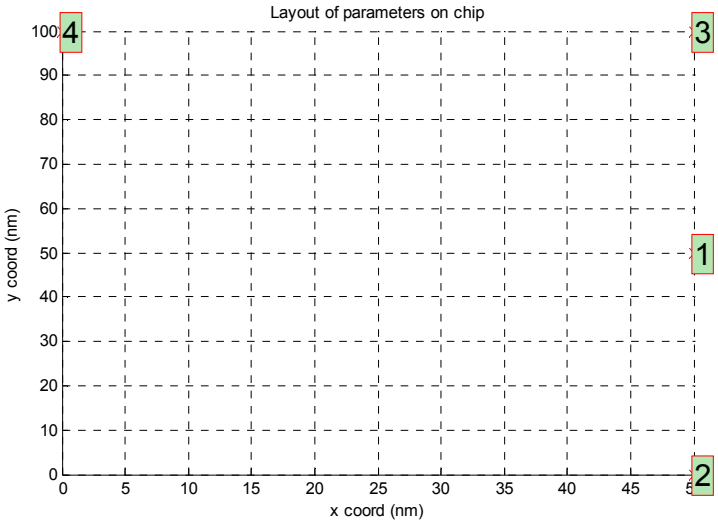


Figure 4.9: CMOS NAND gate on-chip layout assumption

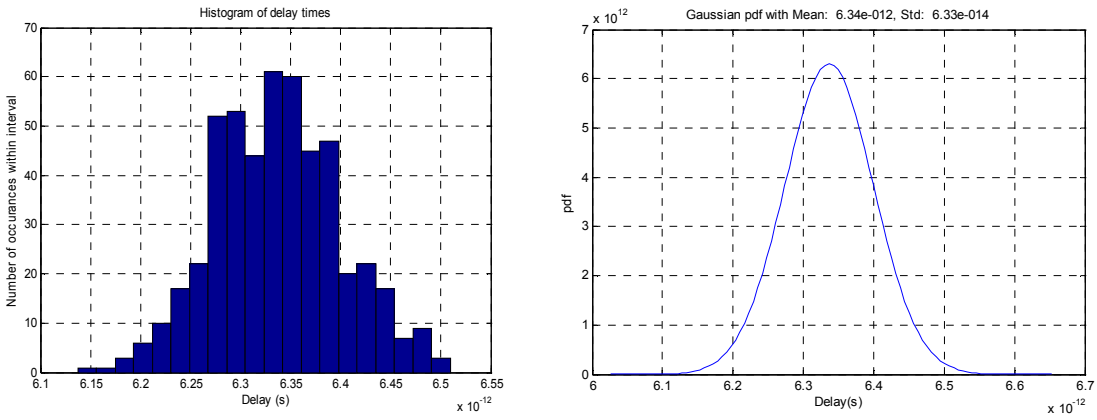
There are four transistors, and we will assume that just one parameter, 'vth0', of the transistor model is subject to statistical variation. Hence, there are just four statistically varying parameters. Assuming the exponential model of intra-die correlation quoted above with $\lambda = 0.007$, the parameter-parameter correlation matrix is found to be as follows:

$$K = \begin{bmatrix} 1 & 0.7 & 0.7 & 0.61 \\ 0.7 & 1 & 0.5 & 0.46 \\ 0.7 & 0.5 & 1 & 0.7 \\ 0.61 & 0.46 & 0.7 & 1 \end{bmatrix} \tag{4.23}$$

Assuming the 'ngNAND.seed' file specifies the standard deviation of all four parameters to be the same, which is 0.01, the correlation matrix can be converted to the following parameter-to-parameter covariance matrix C:

$$C = \begin{bmatrix} 1e-4 & 7e-5 & 7e-5 & 6.1e-5 \\ 7e-5 & 1e-4 & 5e-5 & 4.6e-5 \\ 7e-5 & 5e-5 & 1e-4 & 7e-5 \\ 6.1e-5 & 4.6e-5 & 7e-5 & 1e-4 \end{bmatrix} \tag{4.24}$$

It is convenient to use MATLAB notation, for example 6.1e-5 representing 6.1×10^{-5}, to represent the matrix entries. Computing a matrix A such that $A^T.A = C$ now allows suitably correlated randomised parameter vectors $\underline{X_j}$ to be generated as outlined above. The timing delay results of MC analysis of 500 randomised circuits when λ = 0.007 are represented by the histogram shown in Figure 4.10(a), to which was fitted the Gaussian pdf shown in Figure 4.10(b).



(a) Histogram (b) Gaussian pdf fitting

Figure 4.10: Gaussian pdf fitted to data for 500 NAND gate circuits (lambda = 0.007)

For this 500 circuit random training run with λ=0.007, the mean and standard deviation of the overall delay were found to be 6.337e-12 and 0.0633e-12 seconds

respectively.

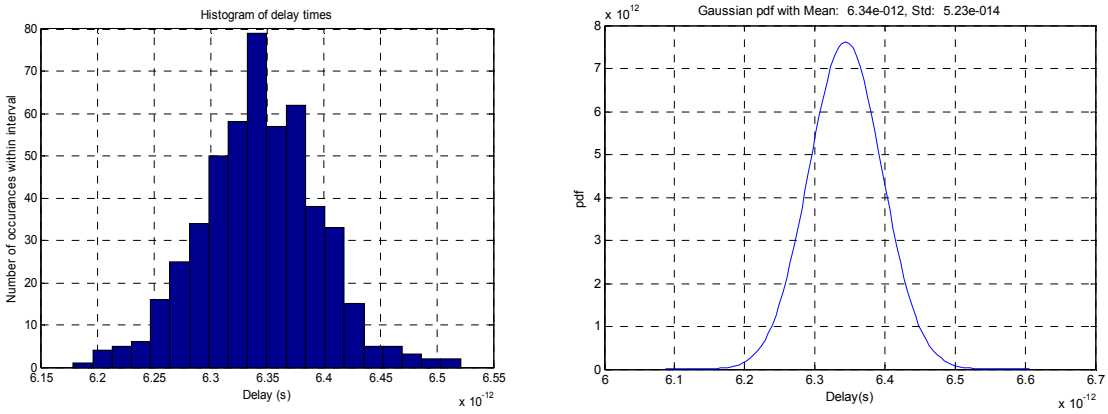
Repeating this procedure for same circuit and layout with lambda = 1, which means that there is almost no intra-die correlation, the parameter-parameter correlation matrix K and covariance matrix C became as follows:

$$K = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad , \quad C = \begin{bmatrix} 1e-4 & 0 & 0 & 0 \\ 0 & 1e-4 & 0 & 0 \\ 0 & 0 & 1e-4 & 0 \\ 0 & 0 & 0 & 1e-4 \end{bmatrix} \tag{4.25}$$

Calculating matrix A such that $A^T.A = C$ gave:

$$A = \begin{bmatrix} 0 & -0.01 & 0 & 0 \\ -0.01 & 0 & 0 & 0 \\ 0 & 0 & -0.01 & 0 \\ 0 & 0 & 0 & 0.01 \end{bmatrix} \tag{4.26}$$

This is not the expected diagonal matrix, but perfectly acceptable where the four parameters are essentially uncorrelated. The results of MC analysis of 500 randomised circuits (when λ =1) are represented by the histogram shown in Figure 4.11(a), to which was fitted the Gaussian pdf shown in Figure 4.11(b).



(a) Histogram (b) Gaussian pdf fitting

Figure 4.11: Gaussian pdf fitted to data for 500 NAND gate circuits (lambda = 1)

For this 500 circuit random training run with $\lambda = 1$, the mean and standard deviation of the overall delay were found to be 6.344e-12 and 0.05233e-12 seconds respectively. In comparison to the case where λ=0.007, the mean remains approximately the same, and the standard deviation reduces by about 17.3 %.

4.5.2 Results from the analysis of a binary full adder with behavioural models of gates

Consider the binary full adder (with seed file swbfan.seed) shown in Figure 4.6(b) whose on-chip layout is assumed to be as shown in Figure 4.12.

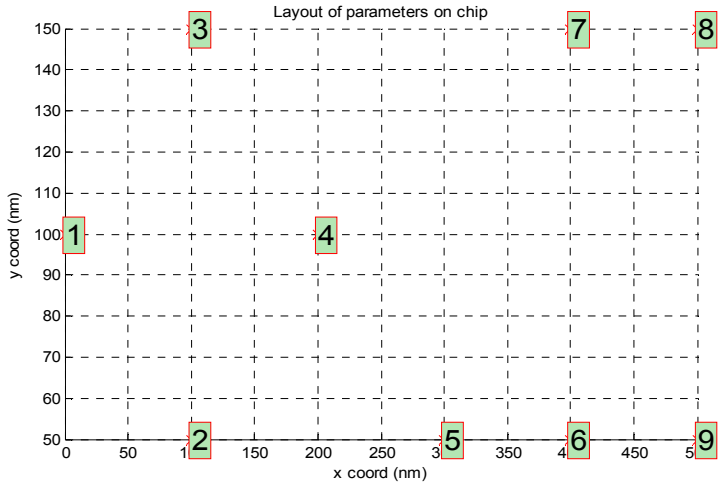
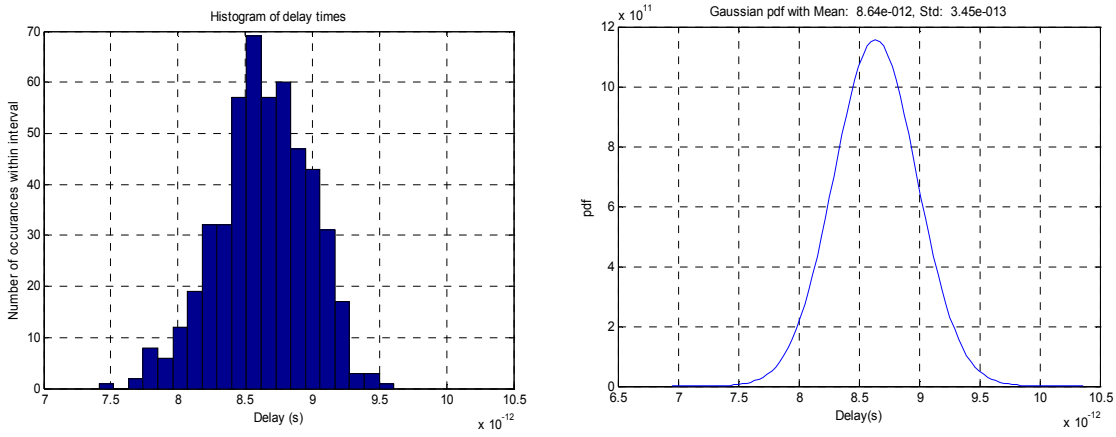


Figure 4.12: On-chip layout assumption for BFA circuit represented with NAND gates

With one delay parameter for each behavioural gate, there are nine parameters in total. Assuming the exponential model of intra-die correlation with $\lambda = 0.007$, the parameter-parameter correlation matrix K is found to be as follows:

$$K = \begin{bmatrix} 1 & 0.46 & 0.46 & 0.25 & 0.12 & 0.059 & 0.059 & 0.03 & 0.03 \\ 0.46 & 1 & 0.5 & 0.46 & 0.25 & 0.12 & 0.11 & 0.056 & 0.061 \\ 0.46 & 0.5 & 1 & 0.46 & 0.21 & 0.11 & 0.12 & 0.061 & 0.056 \\ 0.25 & 0.46 & 0.46 & 1 & 0.46 & 0.24 & 0.24 & 0.12 & 0.12 \\ 0.12 & 0.25 & 0.21 & 0.46 & 1 & 0.5 & 0.37 & 0.21 & 0.25 \\ 0.059 & 0.12 & 0.11 & 0.24 & 0.5 & 1 & 0.5 & 0.37 & 0.5 \\ 0.059 & 0.11 & 0.12 & 0.24 & 0.37 & 0.5 & 1 & 0.5 & 0.37 \\ 0.03 & 0.056 & 0.061 & 0.12 & 0.21 & 0.37 & 0.5 & 1 & 0.5 \\ 0.03 & 0.061 & 0.056 & 0.12 & 0.25 & 0.5 & 0.37 & 0.5 & 1 \end{bmatrix} \quad (4.27)$$

Knowing the standard-deviations of the parameters allows K to be converted to a parameter-to-parameter covariance matrix C. Matrix A may be computed such that $A^T.A = C$, and finally the vectors $\underline{X_j}$ of 500 random circuits may be generated and analysed using ngSPICE to produce the histogram and fitted Gaussian pdf shown in Figure 4.13 (a) and (b) . The mean is 8.64e-12 and standard deviation is 0.345e-12 seconds.

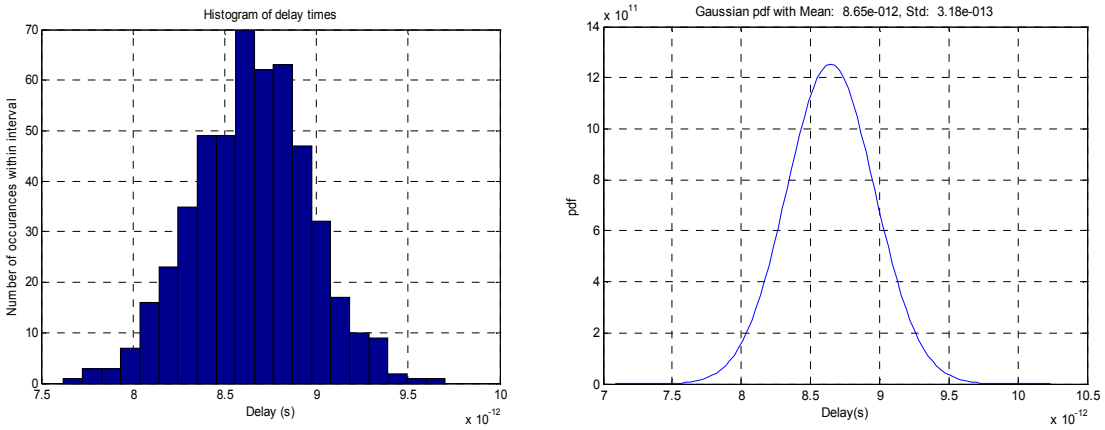


(a) Histogram (b) Gaussian pdf fitting

Figure 4.13: Gaussian pdf fitted to data for 500 BFA circuits (lambda = 0.007)

Repeating the same procedure with λ=10 (no correlation) and 500 training circuits produces the histogram and Gaussian pdf shown in Figure 4.14 (a) and (b) whose mean is 8.647e-12 and standard deviation is 0.318e-12 seconds. Once again, in

comparison to the correlated case where λ=0.007, the mean remains approximately the same. The standard deviation reduces from 0.345e-12 to 0.318e-12, that is by about 7.8 %.



(a) Histogram (b) Gaussian pdf fitting

Figure 4.14: Gaussian pdf fitted to data for 500 BFA crts (lambda = 10)

4.6 Conclusions

The ever-reducing dimensions of nano-CMOS technology mean that statistically based variability analysis will have an increasingly important role in enabling successful circuits to be designed and optimized. The means of analysing the effect of inter-die and intra-die variability is needed, and can be provided through the use of SPICE simulations of randomised versions of a circuit. This 'Monte-Carlo' type of analysis requires a randomization procedure and such a procedure is provided by the professional version of SPICE, which is HSPICE. Powerful though this is, it lacks access and flexibility for research purposes, and the use of RandomSPICE with a MATLAB harness provides a useful alternative. The author has developed a new MATLAB-based Harness for this purpose, with better functionality. It has been briefly described in this chapter, and will be the basis of research described in subsequent chapters. An immediate advantage of this approach is that the open source version of SPICE, known as NGSPICE, may be used and the work of this thesis may be useful as a contribution to the GNU 'gEDA' project.

Although the Monte Carlo error has the nice property that its convergence rate of $1/\sqrt{N}$ does not depend on the dimension, this rate is often considered quite slow. For example, to reduce the error by a factor of 10, one must increase the sample size n by 100 (on average). For this reason, a lot of work has been done on finding ways of improving the Monte Carlo error.

The computational complexity required to perform traditional Monte-Carlo type analysis on larger circuits becomes prohibitive and ways of reducing this complexity must be found. 'Quasi Monte Carlo' techniques and adaptations of 'Extreme Value Theory' can achieve complexity reduction and appear worthy of further exploration. The use of simplified 'behavioural' models of commonly used sub-circuits is another way of reducing computational complexity.

Monte Carlo algorithms compute definite integrals of functions of vectors (containing many variables) by evaluating the function for large sets of randomised vectors covering the space or range of integration. In this thesis, the function will be some circuit parameter, for example a delay, as may be estimated by SPICE simulation. The vectors will contain variables such as the parameters of transistors and other components such as wires, which, in practice, will be expected to vary randomly. The definite integral will be the volume of the 'tail' of the probability density function (PDF) where some aspect of the performance, for example the delay, falls outside some defined limit. The parameter values of transistors and other components will have particular statistical distributions and correlations determined by the physics of the fabrication process and many other effects, and these must be represented by the choice of vectors supplied to the Monte Carlo process. Therefore, repeated SPICE simulations must be performed for the randomised vectors to generate the required distribution of circuit measurements.

For the integrated circuits envisaged by the work in this thesis, the dimensions of the input vectors, i.e. the number of variables, will be extremely high. Each transistor model may have as many as 300 parameters, and there may be a very large number of transistors. Although Monte Carlo methods are known to be efficient for very high dimensional applications, the computational complexity of this application is likely to prohibitive for all but the very simplest circuits. Therefore, it is vital to

find ways of reducing computational complexity. There are many possibilities, one of which, proposed by Amith Singhee [15], makes use 'Quasi Monte Carlo' methods [25] as described in Chapter 7. The 'statistical blockade', also proposed by Amith Singhee [15][21][29] achieve computational savings by different methods, which is described in Chapter 6. The Principal Component Analysis Monte Carlo method and Statistical Behavioural Circuit Block are also investigated in this thesis as ways of reducing the computation in statistical analysis by reducing the dimensions of the problem space, as described in Chapter 5.

Before starting a simulation study, it is important to make sure the random number generator to be used is reliable and has been tested appropriately. Two examples of generators that are known to be reliable are L'Ecuyer's MRG32k3a [93], and Matsumoto and Nishimura's Mersenne-Twister [94]. The latter is implemented in MatLab®7 and will be employed in this project. It has a repetition period of 2^{19937} -1 which means that about 10^{6000} uniformly distributed independent random numbers may be generated before the sequence starts to repeat. The uniform random variables are transformed to Gaussian by the inverse cumulative Gaussian distribution function 'norminv' provided by MATLAB. Special care must be taken for distributed implementations since the pseudo-random number generator's starting point must not be allowed to be the same for each 'worker' machine, as it may be by default.

Chapter 5

Dimension Reduction of Monte Carlo Circuit Simulation

5.1 Introduction

The assertion that the error resulting from MC analyses is proportional only to the square root of the sample size does not mean that the same sample size is appropriate to any circuit no matter how complicated it is. There are clearly advantages in reducing the dimensionality of analysis problems including the simplification of the computational complexity of the analyses. This chapter deals with two methods of reducing the dimensionality of Monte Carlo analysis. The first is Principal Components Analysis (PCA) which transforms the random variables required to characterize a circuit to a reduced number of statistically independent variables. PCA is also useful as a means of introducing intra-die and inter-die correlation. The second is the use of statistical behavioural circuit blocks (SBCB) which substitute functional but computationally simpler circuit models for device level analogue sub-circuits. Both these techniques are well known and provided in user form by the commercial version of SPICE, which is HSPICE. However, they are not yet available with the current open source version, i.e. NGSPICE, and the aim of this chapter is to apply them in new ways which may be suitable for inclusion in the ongoing NGSPICE open source project [37].

5.2 Principal Component Analysis (PCA)

5.2.1 Introduction to the Concept

PCA is a technique for transforming samples of M variables into samples of a smaller number of variables by exploiting interdependency or correlation among the M variables. Referring to a set of R parameter-vectors ('feature-vectors') in an M dimensional vector-space, the transformation converts each parameter-vector to a lower dimensional vector in such a way that no information, or little information, is lost. This means that the original vectors, $\underline{X}_i$ for $i = 1, \ldots, R$, can be reconstructed from the transformed vectors either exactly or with a small error. The elements of the transformed (reduced dimensional) vectors are the coefficients (or 'loadings') of 'principal component' (PC) vectors. PC vectors are eigenvectors of the $M \times M$ covariance matrix C which are normalized and statistically independent. If there is no interdependency among the elements of the original R vectors, PCA cannot reduce the number of variables without significant loss of information.

5.2.2 Performing the Analysis

Taking the summed squared differences between the elements of an original component ('feature') vector and a reconstructed one as a measure or the error or loss of information incurred by PCA, of all possible linear transformations to a lower dimensional space PCA is optimal in minimising this error over all vectors. Further, the PC vectors are conveniently ordered in the sense that the first one has the highest variance and accounts for as much variability as possible. Each succeeding PC vector has lower variance, and therefore less importance, but has the highest variance possible while being uncorrelated to all the previous ones.

PCA may be carried out by eigenvalue/eigenvector decomposition of the M by M covariance matrix of the original vectors data matrix which is defined as:

$$C = [c_{ij}] \quad \text{where} \quad c_{ij} = \frac{1}{R-1}\sum_{k=1}^{R}(x_{ik} - m_i)(x_{jk} - m_j) \tag{5.1}$$

In this expression, x_{ik} is the ith element of vector $\underline{X}_k$ with m_i denoting the mean of x_{ik} for each variable i over all R vectors, i.e. the mean of row i of matrix $[x_{ik}]$ for k=1,2,…, R. Therefore, C is the covariance matrix of variations from the mean of each parameter, and is unaffected by the means themselves. The means of columns are not subtracted from columns. If the eigenvectors of C are $\underline{u}_1$, $\underline{u}_2$, …, $\underline{u}_M$, with corresponding eigenvalues λ_1, λ_2, …, λ_M, these are easily calculated by the MATLAB statement '[U, D] = *eig*(C)'. This statement produces the matrix:

$$U = [\underline{u}_1\ \underline{u}_2\ \ldots\ \underline{u}_M] \tag{5.2}$$

composed of all the eigenvectors as columns, and matrix D whose diagonal elements are λ_1, λ_2, …, λ_M with all other elements being zero. It follows from the definitions of eigenvectors and eigenvalues ($C\underline{u}_k = \lambda_k\underline{u}_k$ for all k) that:

$$CU = UD \qquad \therefore \qquad U^TCU = D \tag{5.3}$$

since eigenvectors are all orthogonal to each other and normal (i.e. of unit length) meaning that U^T is always the inverse of U. Therefore, $C = UDU^T$ meaning that

$$C = \lambda_1 (\underline{u}_1\ \underline{u}_1{}^T) + \lambda_2 (\underline{u}_2\ \underline{u}_2{}^T) + \ldots + \lambda_M (\underline{u}_M\ \underline{u}_M{}^T) \tag{5.4}$$

If matrix X is transformed to Y as follows:

$$Y = U^T . X \tag{5.5}$$

then:

$$X = U . Y \tag{5.6}$$

If any of the eigenvalues λ_1, λ_2, …, λ_M are zero and we remove their corresponding eigenvectors and corresponding rows of Y in equation 5.5, the remaining eigenvectors are principal components which can represent all the original data, without any loss of accuracy. Equation 5.6 will reconstruct the original vector exactly from the non-square matrix Y of reduced dimensional PC coefficient (loading) vectors. Taking this idea further, we can remove elements of Y with their eigenvectors when the corresponding eigenvalues are small on the grounds that if

they do not significantly affect C they should not significantly affect X either. It may be shown that C will always be a positive semi-definite symmetric matrix (i.e. positive definite or singular), therefore all eigenvalues will be real and positive, or zero.

The above outline of PCA hides the obvious difficulty of deciding what error is incurred by removing components with non-zero eigenvalues which are considered small, and how small an eigenvalue must be to be considered negligible. Such considerations of PCA are application specific and best related to the specific objective and how the error is to be quantified. They will be considered later.

5.2.3 Application of PCA to Modelling Statistical Variation

Assume we have a database of sets of randomised BSIM4 device parameters which have been published by a manufacturer or indeed generated on the basis of theoretical calculations as carried out by the originators of RandomSPICE. We will take the RandomSPICE Toshiba NMOS database as the example. For each device there is a set of 201 randomised parameter lists each with 300 parameters. Many of the parameters are zero or fixed in this database, but in general they need not be. The author's MATLAB PCA script computes the 300 by 300 element covariance matrix C for the parameter sets after subtracting the mean for each parameter. The 300 eigenvectors and eigenvalues of C are calculated and then the original mean-subtracted data is transformed by projecting each vector on to the set of 300 eigenvectors. As a check, reversing this transformation produces the original data but with the expected rounding error which causes a mean-squared difference of around 10^{-20} between original and reconstructed parameters. The first ten ordered eigenvalues are plotted in figure 5.1 and are seen to fall off rapidly in magnitude. Since the curve levels off at around six, it seems reasonable to delete all but the first six eigenvalues and eigenvectors to obtain the required reduced dimensional PC coefficient vectors. Figure 5.2 shows how the mean squared difference between original (mean subtracted) data and reconstructed data varies with the number of eigenvectors. This shows how the approximation improves with the number of

eigenvectors and that the mean squared error falls to around 10^{-20} when this number reaches about six. The modelling technique used by Glasgow University [41] explains this result. The leveling off at six principal components makes it reasonable to take this number for our continuing experiments. Similar results and graphs were obtained for the PMOS data provided by RandomSPICE.

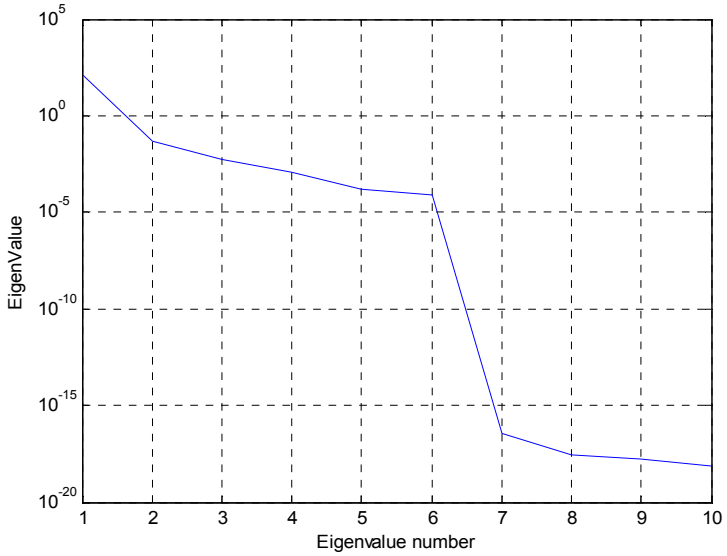


Figure 5.1: Values of first ten ordered eigenvalues for Toshiba NMOS data

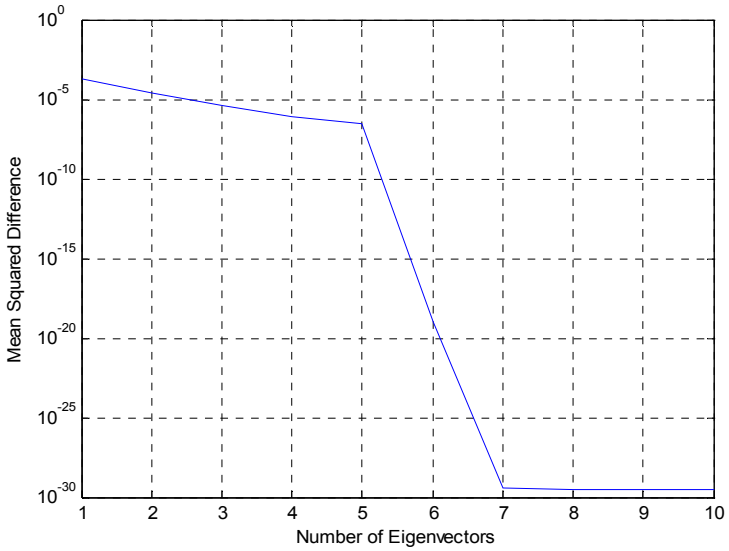


Figure 5.2: Mean square difference between original (mean-subtracted) data and PCA approximated data as number of eigenvectors increases

Only seven parameters of the BSIM4 transistor model used to produce this illustration of PCA, as provided by Glasgow University's RandomSPICE database [13], vary among the models in the set. The seven parameters are those required for modelling random discrete dopant effects (as in references [65] and [75]) and are as follows:

1. RDWMIN: Lightly-doped drain resistance per unit width at high V_{gs} and zero V_{bs} for RDSMOD=1. RDSMOD is the parameter of bias-dependent source/drain resistance model selector.
2. NFACTOR: Subthreshold swing factor.
3. DSUB: DIBL coefficient exponent in subthreshold region.
4. A1: First non-saturation effect parameter.
5. A2: Second non-saturation factor.
6. VOFF: Offset voltage in subthreshold region for large *W* and *L*.
7. LPE0: Lateral non-uniform doping parameter at V_{bs}=0.

This PCA analysis needed some consideration of numerical stability because of the wide variation of parameter values. There is clearly correlation among the seven varying parameters listed above which can be eliminated by PCA to the advantage of Monte Carlo simulation.

5.2.4 Applying PCA to Reduce Dimensionality in MC Simulation

PCA clearly has value in reducing the dimensionality of the randomization required for MC analysis. It is now possible to randomize the principal component coefficient vectors rather than the complete list of device parameters, and then transform these back to a set of parameters to be recognized by SPICE for each device. An independent randomization may clearly be performed for each device, where the effects of inter-die or intra-die variability are not required to be modeled.

However, the modelling of intra-die variability is afforded in a convenient way by the use of PC coefficient vectors, and the correlation introduced by proximity on the die can be conveniently applied to these. Inter-die variability can also be introduced in this way. The approach is to determine a set PCs for each device model and then to introduce correlation into the corresponding PC coefficient vectors

for each device within a circuit or sub-circuit, according to the exponential model outlined in Section 4.5 of the previous chapter. The correlation matrix should be partitioned into a number of smaller ones, each catering for one principal component which will be independent of all the others. If the device model has six PCs, there will be six partitions, one for each PC. A different value of λ can be used for each partition.

5.3 Behavioural Modelling

5.3.1 Introduction to the concept

The idea of behavioural modelling is to substitute functional but computationally simpler circuit models for device level analogue sub-circuits. The simpler circuits emulate, as closely as possible, the input-output behaviour of the sub-circuits they replace. Their use can reduce the computational complexity of SPICE simulations especially for large and complex integrated circuits. SPICE provides specific behavioural modelling options that are suitable for the 'mixed signal' simulation of digital circuits. The aim is to allow analogue effects to be taken into account but without the computational cost of a full device-level analogue simulation. The options include the use of controlled voltage or current sources which can be configured to emulate operational-amplifiers, switches, logic gates, delay lines and many other devices whose outputs can be represented by or approximated by continuous functions of the inputs and also time. SPICE allows such functions to be specified in many forms, including the use of look-up tables for the waveforms required and the input-output relationships. Rectangular digital waveforms may be defined as inputs and outputs with the specification of clock rise and fall times, on-off periods, and voltage or current levels. Controlled sources can be used to model gate-switching action either with close to ideal fast switching or some specific linear or non-linear behaviour which depends on nodal voltages and currents elsewhere in the circuit.

5.3.2 How SPICE Implements Behavioural Model Components

All forms of SPICE are fundamentally based on nodal analysis which characterises linear circuits in the Laplace transform domain as $Y(s)V(s) = I(s)$, The vector $I(s)$ represents all independent current sources, $V(s)$ is the vector of voltages at connection points (nodes) and $Y(s)$ is the nodal admittance matrix with elements determined by circuit components. For transient analysis, the equation is solved as a multivariate differential equation in finite difference form. The required time-span is split up into time-steps and a solution is required at each step. Where there is non-linearity, for example due to semiconductor junctions, time-varying characteristics and switching devices, a Newton-Raphson (or Raphson-Newton) iterative approach must be used at each time-step to take into account the effects of the dependencies of variables and component values on others, and of time itself. Convergence to a set of nodal voltages for each time-step must occur before moving onto the next time-step. If the time-steps are too large, the simulation may be inaccurate and even fail to converge, and if they are too small the computation time may become prohibitively long. All versions of SPICE adjust the time-step automatically according to the progress of the simulation. The step is reduced when values are changing quickly and increased when changes are slow.

The equation $Y(s)V(s) = I(s)$ is essentially Kirchhoff's Law which makes the sum of all sources of current flowing into each node equal to zero. Each element of $Y(s)V(s)$ is an ideal voltage-dependent current source (VCCS) and hence such sources can be represented directly. All other components, devices and sources must be modelled using the elements of $Y(s)$ and $I(s)$. A voltage source must be modelled by a current source and a resistor, and a junction by a VCCS and resistance. The models are non-linear and vary with circuit conditions and time. The more accurate the models, the more complicated the analysis generally.

HSPICE provides a host of devices which it represents in circuit and time-dependent nodal analysis form in proprietary ways. It provides highly versatile dependent voltage and current sources referred to as E-elements, F, G and H-elements [31]. These can be included as fundamental building blocks in models for

MOS and bipolar transistors, diodes, analogue operational amplifiers and a large variety of other circuits.

5.3.3 Using E, F, G or H Elements with Look-up Tables

Each of the E, F G and H elements can model behaviour which is a linear or non-linear function of controlling-node voltages or branch currents. These elements have many possible functions including 'behavioural' voltage or current sources (according to HSPICE documentation) and ideal delay elements. Ideal delay elements would be very useful in behavioural simulation, but their use in SPICE for MC analysis raises problems which will be discussed later. Behavioural models of a 2-input AND gate (X=A.B) and a 2-input NAND gate (X=A nand B) gate using G and E Elements to model ideal switching behaviour are presented in Tables 5.1 and 5.2. The logic functions are implemented by lookup table. Figure 5.3 shows the response of the NAND gate as defined by Table 5.2 to a voltage pulse with finite (10ns) rise-time and fall time.

```
g 0 X and(2) A 0 B 0
+0.0   0.0ma
+0.5   0.1ma
+1.0   0.5ma
+4.0   4.5ma
+5.0   5.0ma
```

Table 5.1: Netlist for 2-input AND gate (X=A.B) using g-element

```
e X 0 nand(2) A 0 B 0
+0.0   5.0v
+0.5   4.8v
+1.0   4.5v
+4.0   0.5v
+5.0   0.0v
```

Table 5.2: Netlist for 2-input NAND gate (X=A nand B) using e-element

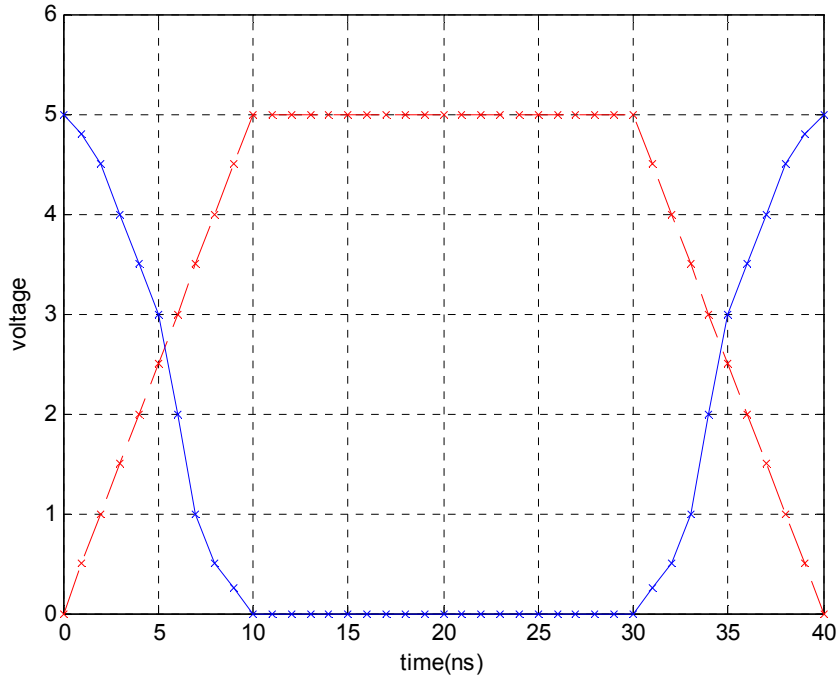


Figure 5.3: Response of 2-input NAND gate as defined in Table 5.2

The use of these ideal voltage dependent elements provides a good way to build up behavioural models suitable for augmenting with delay modelling for statistical timing analysis. It is also useful to employ voltage-controlled resistors (one of the g-element options) to implement a switch-level MOSFET.

5.3.4 Tau Models of Devices

The Tau Model of a transistor has long been used as a simple behavioural model in many transistor optimisation tools for designing integrated circuits, such as TILOS[46], COP[104] and EPOXY[118]. They are commonly used for both synchronous and asynchronous circuits [16]. It uses an ideal switch with simple RC circuitry to introduce delay, α, as determined by the transistor's gate dimensions and the technology. In its simplest form, each transistor is modelled as an ideal switch with appropriate on/off resistance, and source, gate and drain capacitance is introduced by discrete capacitors again each being determined from the gate

dimensions. The gate delays between inputs A and B and the output X of a CMOS 2-input 'pull-down' circuit as referred to in Section 2.2.3 may be modelled [16] by the RC circuit below Figure 5.4:

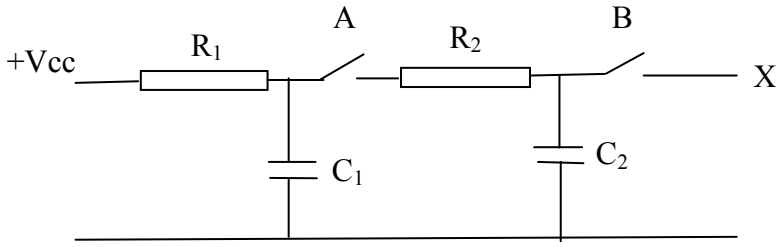


Figure 5.4: Tau model for CMOS 'pull down' sub-circuit

The following expressions are obtained where $\alpha\uparrow_{AX}$ and $\alpha\uparrow_{BX}$ are the delays in the response to rising A and B transitions respectively:

$$\alpha\uparrow_{AX} = R_1C_1 + (R_1 + R_2)C_2$$

$$\alpha\uparrow_{BX} = (R_1 + R_2)\, C_2$$

Such simple behavioural models are known to be convenient for predicting the performances of synchronous and asynchronous digital circuits and for guiding designers towards working efficient circuits. Optimisation techniques based on such models have been used to determine ideal sizes for the CMOS transistor devices that constitute a complex circuit [16].

Lookup tables as exemplified in Table 5.1 and 5.2, and 'tau models' are very simple, but each has its limitation. Look-up tables can model limited and non-linear 'slew rates' but not delay, whereas tau models are either restrictive if simple RC transition timing models are used, or complicated with more accurate ones. A single RC time-constant gives delay, but with a characteristic exponential rising or falling waveform that cannot be changed except by introducing other RC components. Here we propose the combination of these two techniques to produce better behavioural models that are still quite simple. The tau model is used to produce the

required delay and the look-up table then modifies the wave-shape with appropriately chosen entries.

Taking the $\alpha_{\uparrow BX} = (R_1+R_2)C_2$ delay as an example, if this is applied to the look-up table model in Table 2, the result is the output waveform plotted in Figure 5.5 which is affected both by delay and wave-shaping according to the look-up table.

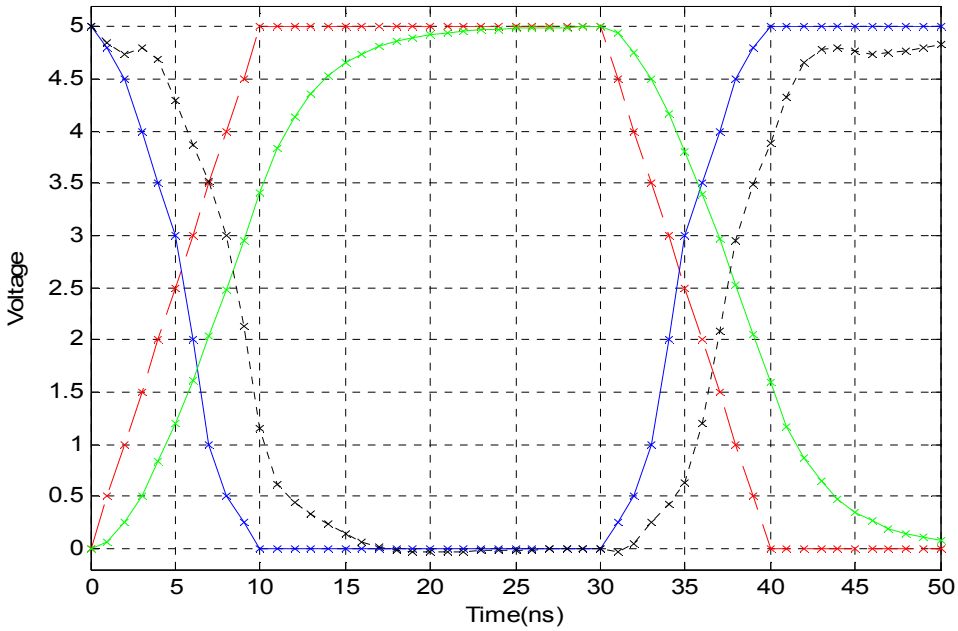


Figure 5.5: Effect of combining look-up table and simple tau model
(a) excitation applied to B (red), (b) output with look-up table alone (blue), (c) effect of tau model on excitation(green), (d) effect of combined model on output(black)

A simple MATLAB curve fitting procedure chooses the RC time-constant and the look-up table for a given wave-shape as may be obtained from a SPICE analysis of the device for which a behavioural model is required.

The Tau and delay model proposed in this thesis may be used for statistical static timing analysis as described in Section 3.5. In many ways, it is similar to the 'Composite Current Source' (CCS) modelling technique provided as part of the 'Liberty' software produced by Synopsys [124]. It is also similar to the 'Effective Current Source Model' (ECSM) provided by CADENCE Design Systems Ltd [125]. CCS and ECSM use very similar modelling techniques and are supported by

comprehensive libraries of cell and interconnect models [126] that are based on industry-supplied data. Although the commercial details of these models are not openly available, the modelling approach that is used appears to be based on, or is similar to, the 'Blade and Razor' current source based model, published in 2003 by J. F. Croix and D. F. Wong [127], and a U.S. patent by Cadence Design Systems [128] in 2004. There have been many other publications inspired by the original idea of using voltage-controlled current sources with passive delay circuits to model the behaviour of interconnections and cells with respect to their timing delay, noise generation, power consumption and statistical variability characteristics [129] [130] [131] [132].

The basic idea of current based delay models is to produce computationally simple circuit elements that consume timing waveforms and produce output timing waveforms that have approximately the right shape and delay profile when driving a further circuit element. This is essentially the idea of the 'tau and delay' modelling approach in the thesis except that voltage-controlled voltage sources (VCVS) are used in this thesis rather than voltage-controlled current sources (VCCS).

According to R. Goyal and N. Kumar of Cadence Design Systems Inc. [125], current source based delay models represent their outputs as 'look-up' tables of current against time. In principle, a different waveform is required for each combination of input waveform slew-rate and output load characteristics, though interpolation between defined data-sets allows reasonable accuracy to be achieved with reasonable resources. The model parameters are therefore determined from the input waveform and the load seen by the driver cell. The load depends strongly on the interconnect network between the driver and the driven cells and, according to Goyal and Kumar [125], interconnect delay dominates other delays in sub 90nm technology, with wiring delay accounting for at least 75% of the overall delay [127]. It is argued [131] that non-linearised Thévenin or voltage-source based driver cell models cannot easily be adapted to accurately modelling the effect of loads with highly non-linear characteristics. Voltage-controlled current source modelling is therefore preferred for modelling the non-linear aspects of the input-output relationships of cells and their interconnections. It is often pointed out (e.g. [124])

that VCCS based models can easily be converted to VCVS based models and vice-versa, but the fundamental differenced lies in the way the commercial libraries have been defined to model the required non-linearities in the source-load interconnections.

The 'tau and delay' based VCVS approach in this thesis is based on the concept of 'tau modelling' as published by Steven M. Burns [16] and the direct use of dependent voltage and current source elements provided by HSPICE with their capacity for behavioural modelling. Both VCVS and VCCS elements are provided by HSPICE, but the choice was made arbitrarily since we did not attempt to study the effect of non-linear loading by devising or utilising libraries. The dominance of interconnection delay in sub-micron technologies offers some justification for this approach.

The waveform matching approach used in this thesis produces behavioural models that generate specified wave shapes with the appropriate delay. It could have been used to generate a library of such models. However, it was used to produce a single model of each cell type derived from the output waveforms obtained from transistor level simulations of the cell (using RandomSPICE based transistor models provided by Glasgow University). Each single cell is characterised by parameters which may be randomised in Monte Carlo analysis runs to simulate the effect of statistical circuit variation. The statistical characteristics of the randomisation (distribution, mean, standard deviation.) that were used to provide the illustrations were derived from transistor level simulations performed using RandomSPICE with the supplied Toshiba 35nm transistor parameters. In principal, a different VCVS model is needed for each different delay, but the required adjustments to the VCVS parameters are small and in practice the same voltage source parameters were used for each delay.

5.3.5 Using Verilog-A in SPICE for Behavioural Modelling.

Verilog-A is a widely used mathematical language for analogue behavioral descriptions that characterize the high-level behaviour of circuits and their structure

and components. Circuits can be defined at a level of abstraction appropriate for behavioural analysis, architectural design, and verification of functionality. Versions of the Verilog-A language are supported by HSPICE and NGSPICE to allow a mixture of Verilog-A descriptions and SPICE netlists to be used to define behavioural or mixed transistor-level and behavioural simulation to be carried out. Verilog-A defined modules are loaded into the simulator with a '·HDL' netlist command and are instantiated in the same manner as HSPICE subcircuits. The use of Verilog-defined functionality is clearly a useful tool for behavioural modelling and must be mentioned here, though it became beyond the scope of the thesis as presented.

5.3.6 Behavioural Models for MC Simulation

SPICE offers a large number of different ways of defining and randomising behavioural models and we have suggested yet another alternative. Unfortunately, the most sophisticated version of SPICE, HSPICE, often does not specify how certain features are implemented and they are offered as proprietary items. The open source NGSPICE, though based on the same long-established analysis engine, does not have many of these features.

All dependent sources in HSPICE have 'ideal' delay as an option, which is most useful in behavioural models. Ideal delay is trivial in digital circuits but unachievable exactly in analogue circuits. Ideal delay does not change the shapes of waveforms, as the tau-model must do. But it must be implemented by an iterative procedure which HSPICE does not disclose to users.

The use of ideal delay for behavioural modelling has been investigated and appears viable until it is used for MC analysis. Randomising the delays, as implemented by transmission lines, even within quite modest circuits with less than 100 devices, causes SPICE to make increasingly slow progress and eventually to 'hang' (apparently). Investigations revealed that it was not the models, but the step-size selection that was to blame. MATLAB randomizes all parameters, which may include ideal delays, with high numerical precision. Hence the relative timing of events on a single chip will vary almost continuously, and delayed switching events

may become very close together. The time-step adaptation algorithm will try to model the very small timing differences and thus generate exceedingly small time-steps. Simply quantizing the Monte Carlo variation (say to two decimal places) eliminated this problem. Therefore, in practice, the randomization should ideally be done with reference to the anticipated time-step size but the effect on the results of the statistical analysis must then be investigated.

The modelling of delay by a linear time-invariant circuit, essentially a filter, with a 'look-up table' switch to modify the wave-form eliminated this problem in the examples we tried. It may be argued that a circuit approach is closer to how the delay within devices is produced anyway. Hence the modified tau-modelling approach has been adopted for the statistical behavioural circuit blocks to be described in the next sub-section. Matching MC randomization of behavioural model parameters to the step-adaptation algorithm of SPICE is a matter deserving further investigation as there may be great economies and insights to be gained.

5.3.7 Statistical Behavioural Circuit Blocks (SBCB)

A statistical behavioural circuit block (SBCB) as proposed in this thesis is a behavioural model of a device such as a transistor, or a circuit building block such as a gate or an adder, based on a combination of lookup table and tau modelling and the use of Verilog A. Its purpose is to model the most important aspects of the device's or circuit building block's behaviour, to an acceptable accuracy, with a relatively small number of parameters. It is intended to be especially suitable for modelling asynchronous circuits derived from BALSA because the models do not rely as heavily as some approaches on the non-linear processing aspects of SPICE for achieving convergence when there is lots of feedback. The delay is implemented by a linear RC circuit which eliminates convergence problems encountered with MC analysis due to difficulties with the SPICE time-step adjustment.

The statistical characteristics of SBCBs are derived by applying traditional MC analysis to true or accurately modeled devices or building blocks. The distributions, means and standard deviations of the behavioural model parameters that produce

outputs matching the true or accurately modeled device or building block outputs are then known and characterise the random variables which are generated to produce randomized copies of the SBCBs. SBCBs are used to replace the true or accurately modelled sub-blocks to reduce the dimensionality and therefore the complexity of the analysis.

The accurately modeled RandomSPICE generated PMOS and NMOS devices were used to generate SBCB models of sub-circuits containing these devices. The training procedure is illustrated in Figure 5.6.

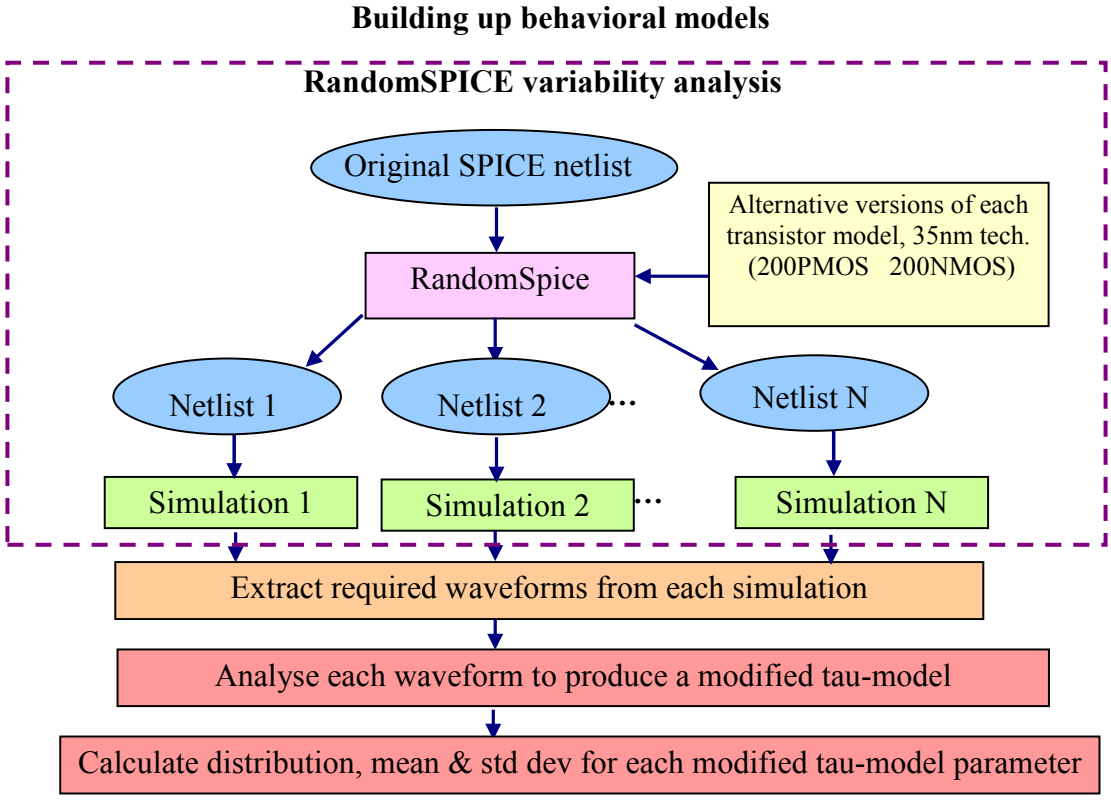


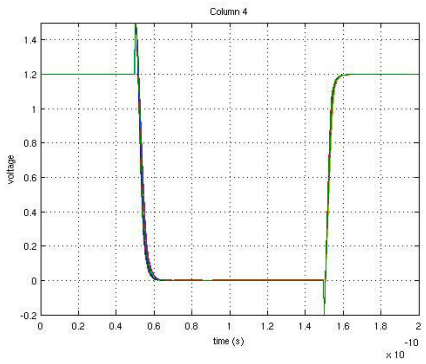
Figure 5.6: Flow chart for building up SCSB circuit blocks

Consider the parameterization of a SBCB for a 2-input NAND gate in 35nm technology. By employing RandomSPICE [12] with the 35nm randomized transistor models provided [6], [7], a set of randomised copies of the NAND gate, described with the netlist in Table 5.3, is produced. By analysing each of these circuits using NGSPICE a delay distribution, as illustrated in Figure 5.7, is obtained. Having decided that a Gaussian distribution is appropriate by a standard test, available in MATLAB, fitting a Gaussian probability density function (pdf) to this distribution allows estimates of its mean and standard deviation to be derived, as illustrated in Figure 5.7.

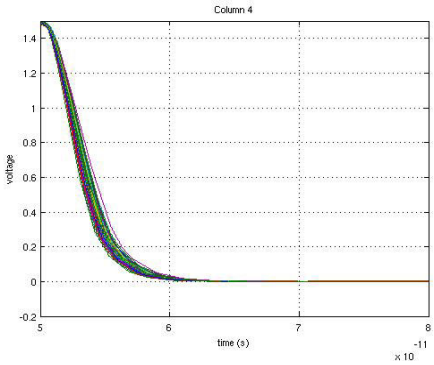
The SBCB delay model of the NAND gate, as illustrated in Figure 5.8, consists of the basic logic functionality implemented as a delay-free element with a modified tau model applied to the input. The basic 'tau' time constant will be the required delay which MC randomization will vary according to the statistics obtained above. Strictly, the look-up table parameters of the modified tau model are dependent on the delay, and should ideally be calculated for all the waveforms generated during the training. In practice as the waveforms are so similar, just one look-up table was produced and used for all delays. The MATLAB matching procedure could have been applied to all of them at little computational cost. Similar SBCB models may be produced for other logic gates, and circuit building blocks.

```
......
MMN1 Z A net3 0 atomn W=35e-9 L=35e-9
MMN2 net3 B 0 0 atomn W=35e-9 L=35e-9
MMP1 Z B vdd vdd atomp W=70e-9 L=35e-9
MMP2 Z A vdd vdd atomp W=70e-9 L=35e-9
......
```

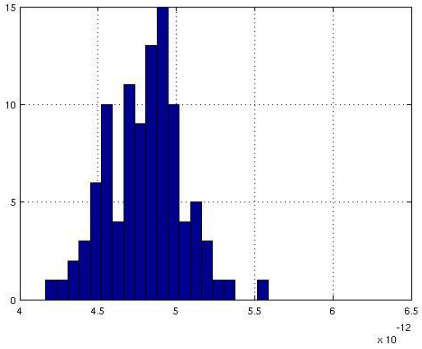
Table 5.3: 2-input NAND gate circuit netlist with transistor model.



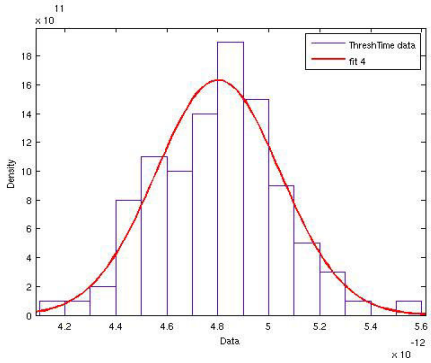
Waveform for output Z from 100 randomized circuits



Zoomed output Z falling edge from100 randomized circuits



Histogram of Z delay distribution from 5e-11s to point at 0.6V threshold voltage



Z delay fitting result
Variance: 5.96957e-26
Sigma 2.44327e-13

Figure 5.7: MC simulation on a 2-input NAND gate implemented with 35nm CMOS

.

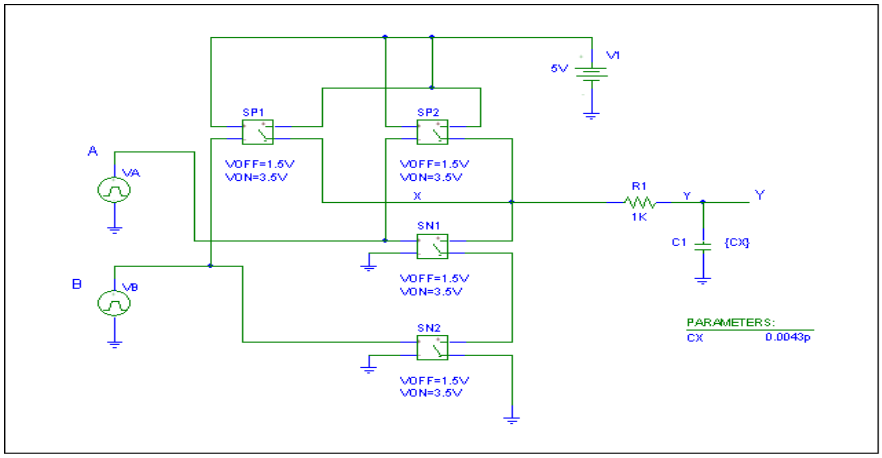


Figure 5.8: NAND SBCB model

When many logic gates and other building blocks within an integrated circuit (IC) are replaced by randomized samples of behavioral models, the dimensionality of the required parameter set will be greatly reduced since the large number of transistor model parameters will have been replaced by a much smaller number of SBCB parameters. When the IC is analysed by MC techniques, the computational complexity of each SPICE simulation will then be greatly reduced.

5.3.8 Improving the Accuracy of SBCB

As mentioned above, the look-up table parameters of the modified tau model are dependent on the delay, and should ideally be calculated for all the waveforms generated during the training.

5.4 Conclusions

This chapter deals with two methods of reducing the dimensionality of Monte Carlo analysis. The first is Principal Components Analysis (PCA) which is also useful as a means of introducing intra-die and inter-die correlation. The second is the use of statistical behavioural circuit blocks (SBCB) which substitute functional but computationally simpler circuit models for device level analogue sub-circuits. The concept of PCA is outlined and it has been applied, using MATLAB functions, to greatly reduce the dimensionality of a database of transistor parameters with small loss of accuracy.

Behavioural models are more problematic than may be anticipated. Randomizing ideal delays, even within quite modest circuits, causes SPICE to make increasingly slow progress and eventually to 'hang'. To understand the cause, it is necessary to understand how SPICE operates, though the exact details of how ideal delay is modeled is not clear. There are several ways of doing it with the facility of the timing event simulator. The step-size selection algorithm is to blame and this may point to a fundamental problem with the use of SPICE for MC simulation and even simulating large asynchronous circuits (maybe). The modelling of delay by a linear time-invariant 'tau' circuit, essentially a filter, with a 'look-up' table to modify

the wave-shape, seems to eliminate this problem for the examples we have considered. However matching the MC randomization of behavioural model parameters to the step-adaptation algorithm of SPICE is a matter deserving further investigation.

Chapter 6

Computation Reduction by Extreme Value Theory

6.1 Introduction

Extreme Value Theory (EVT) [26] is a branch of statistics concerned with the estimation of probabilities which are ‘extreme’ in the sense that the range of values of interest is many standard deviations from the mean of an assumed probability distribution. The theory is based on a comparison between pdf estimations that would be obtained for an infinite or very large sample size and for pdf estimations obtained for a much smaller set of specially selected values. The specially selected values are those for which some classifier predicts ‘extreme’ outcomes whose deviation from the mean exceeds a certain threshold. In some fields, this is referred to as a ‘Peak Over Threshold’ (POT) based approach [108]. This chapter concerns the application of EVT, in the form of an algorithm called ‘Statistical Blockade’, to statistical circuit analysis. The potential for achieving major computational savings will be demonstrated.

6.2 Statistical Blockade

EVT is concerned with the faster statistical analysis of ‘rare events’ which occur in the far tails of probability distributions. An algorithm known as ‘Statistical Blockade’ (SB) [28] applies EVT to circuit analysis by eliminating or ‘blocking out’ randomised parameter vectors that are classified as being unlikely to produce circuits

that fall in the low-probability tails. The intention is that only the ones likely to produce 'rare events' are analysed. In our application, the rare events are the circuit yield failure predictions which are extreme in the sense that they are on the 'tails' of Gaussian-like probability distributions for circuit quantities such as overall delay. Because they are designed to be rare, reliable estimates of these failures by conventional MC techniques require very large numbers of randomised parameter vectors.

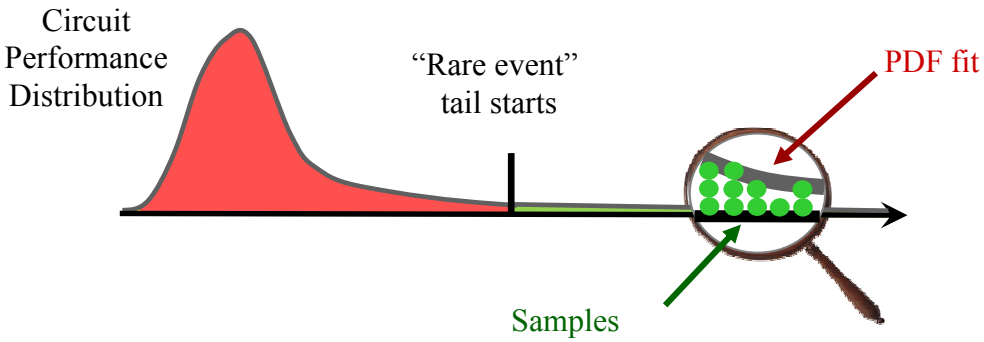


Figure 6.1: Illustration of 'rare events' in distribution tail (as in [28])

In the context of circuit simulation, the idea of SB is to try to concentrate on parameter vectors that are likely to generate the 'rare events' of failing circuits, and block out or disregard the ones that are unlikely to produce such failing circuits. Many input vectors are generated, but only the ones likely to produce 'rare events' are simulated. This partial sampling of the performance distributions is the basis of EVT. The computational complexity involved in introducing the bias, and compensating for it, is much less expensive than performing lots of uninteresting circuit simulations. The 'blockade filter' is a standard classifier as used in machine learning and data mining. It is trained by simulating a relatively small 'training set' of randomized circuits' and is further refined as more and more simulations are carried out. Statistical blockade, with this recursive updating, is intended to make estimation of rare event statistics computationally feasible [107].

The idea is similar to 'importance sampling' which was mentioned in Section 4.2, as one way of directing MC sampling effort toward the most important regions of the domain of input variables. There are some important differences, however

and the proposers of Statistical Blockade [109] are adamant that it is fundamentally different from importance sampling.

Importance Sampling, as introduced in Section 4.2, can be used to predict failure probabilities [110], and has recently [109] been applied to the modelling of 'rare' failures in arrays of SRAM cells. It addresses the problem of having too small a number of observations of 'rare events' by changing the probability distribution of the underlying random variables so that events of interest occur more often. The bias thus introduced is appropriately corrected. The concept of 'mixture importance sampling' [95][111][113] was proposed for this purpose. SPICE simulations are used at device level to estimate the probability of a single value of the performance metric exceeding a defined threshold. An estimate of the distribution tail is not computed and all performance metrics are combined in the computation of failure probability. Traditional 'importance sampling' algorithms cannot produce separate probability estimates for each metric except by re-runs of the simulations and D.E. Hocevar *et al* [112] advise against importance sampling in high dimensions because of its computational complexity.

The Statistical Blockade approach addresses the problem of generating enough 'tail' points to obtain statistical confidence for circuit failure rate estimation without prohibitive computational complexity. The aim is to obtain accurate estimations of the 'tails' of probability distributions. The approach suggested by EVT and adopted by 'Statistical Blockade', as proposed by Amith Singhee et al. [15] [28] [29], and implemented in modified form in this thesis, may be summarised as follows:

(1) Generate a limited set of representative randomised parameter vectors, sufficient for the training of a classifier which is able to predict when other parameter vectors are likely to produce circuits with a property of interest that is more than a given number of standard deviations from the mean.

(2) Generate a sufficiently large number of representative randomised parameter vectors to ensure that some are likely to produce circuits whose properties of interest fall within the tails of their distribution. This number will be far more that could be simulated with feasible computational complexity.

(3) Block out parameter vectors that are classified as being unlikely to produce

circuits that fall in the low-probability tails.

(4) Simulate only the circuits whose parameters are considered likely to produce 'rare events'.

(5) Perform 'partial sampling' of the achieved performance distributions, according to the basic theory of EVT.

6.3 Classification Techniques for Machine Learning

The classifier is critical to the success of SB and requires a form of supervised learning as often used in the general field of Machine Learning[30]. To construct a classifier, a set of classified training examples, consisting of objects and their associated classes is required. The examples are called 'labeled examples', and the set of labeled examples provided for training the classifier is called the training set. Once a classifier has been trained, its effectiveness may be determined by employing a second set of labeled examples called the 'test set'. The test set must be different from the training set. The percentage of test examples from the test set that are correctly classified becomes the 'classification rate' and the percentage of test examples misclassified is the 'misclassified rate'. Support vector machines (SVMs) [113][114] are supervised learning methods that may be used for classification and regression.

The classifier used in the author's implementation of Statistical Blockade consists of a linear estimator followed by a threshold decision maker. Such simple 'classifiers' are commonly used in machine learning and data mining. In this context the classifier will be referred to as a 'blockade filter'. It may be considered a type of vector support machine (VSM).

6.4 A Linear Estimator for Statistical Blockade

To produce a training set for the SB classifier, a set of R randomized circuits is produced and analysed using SPICE to obtain measurements of features of interest, such as delay and power consumption, which constitute the 'labels' of the circuits. Assume that each circuit has N parameters:

$$x_{r1}, x_{r2}, \ldots, x_{rN} \tag{6.1}$$

where r is the circuit index in the range 1, 2, …, R, and that one measurement feature, D_r say, is of interest for each circuit r.

The required linear estimator for Statistical Blockade will ideally be required to estimate (or predict) the value of Di for any circuit i from its known set of parameters $x_{r1}, x_{r2}, \ldots, x_{rN}$ to eliminate the need to use SPICE to analyse that circuit. This will result in a great saving of computation time since the implementation of a linear estimator is very simple indeed. If all circuits could be 'analysed' in this simple way, the time saving would be enormous, but this is not the expectation. The expectation is that the estimator will predict whether the feature of interest Di for a particular circuit i lies within the main body of the probability distribution for this feature, or whether it is likely to be in the tail. If it is predicted to be in the main body, the circuit need not be analysed as many like it will have already been analysed. But if it is likely to be in the tail, the circuit is of great interest and must be analysed. Issues of predictions that turn out to be false, either as false positives or false negatives, must of course be addressed, and will be.

The linear estimator when applied to the parameters of circuit i is defined as:

$$W_i = a_0 + a_1 x_{i1} + a_2 x_{i2} + \ldots + a_N x_{iN} \tag{6.2}$$

where $a_0, a_1, a_2, \ldots, a_N$ are the estimator's coefficients that must be trained before the estimator can be used effectively. The training requires a training set of R circuits, as mentioned above, that have all been accurately analysed, using SPICE, to a obtain the true parameter of interest D_r for each circuit r. Express the estimates W_r for each circuit r for r=1, 2,…, R and the true measurements D_r for r=1, 2,…, R , obtained from SPICE, in vector form:

$$\underline{D} = \begin{bmatrix} D_1 \\ D_2 \\ D_3 \\ \vdots \\ D_R \end{bmatrix} \qquad \text{and} \qquad \underline{W} = \begin{bmatrix} W_1 \\ W_2 \\ W_3 \\ \vdots \\ W_R \end{bmatrix} \tag{6.3}$$

The requirement of the training procedure is to choose the estimator's coefficients a_0, a_1, a_2,..., a_N to make $\underline{W}$ as close as possible to $\underline{D}$ over the whole training set.

Since,

$$\begin{aligned} W_1 &= a_0 + a_1 x_{11} + a_2 x_{12} + \ldots + a_N x_{1N} \\ W_2 &= a_0 + a_1 x_{21} + a_2 x_{22} + \ldots + a_N x_{2N} \\ W_3 &= a_0 + a_1 x_{31} + a_2 x_{32} + \ldots + a_N x_{3N} \\ &\vdots \qquad\quad \vdots \\ W_R &= a_0 + a_1 x_{R1} + a_2 x_{R2} + \ldots + a_N x_{RN} \end{aligned} \tag{6.4}$$

It follows that

$$\underline{W} = X\,\underline{a} \tag{6.5}$$

where:

$$X = \begin{bmatrix} 1 & x_{11} & x_{12} & \ldots & x_{1N} \\ 1 & x_{21} & x_{22} & \ldots & x_{2N} \\ 1 & x_{31} & x_{32} & \ldots & x_{3N} \\ 1 & x_{41} & x_{42} & \ldots & x_{4N} \\ 1 & x_{51} & x_{52} & \ldots & x_{5N} \\ \vdots & \vdots & \vdots & \vdots & \\ 1 & x_{R1} & x_{R2} & \ldots & x_{RN} \end{bmatrix} \qquad \text{and} \qquad \underline{a} = \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_N \end{bmatrix} \tag{6.6}$$

Hence the requirement is that the vector of coefficients, $\underline{a}$, must be chosen such that $\underline{W} = X.\underline{a}$ is made close as possible to $\underline{D}$. If X were a non-singular square matrix it would be possible to make $\underline{W} = X.\underline{a}$ exactly equal to $\underline{D}$ by taking

$$\underline{a} = X^{-1}\,\underline{D} \tag{6.7}$$

Here X^{-1} is the inverse of X which is easily calculated in MATLAB. However X will in general not be square and it will not be possible to find a coefficient vector $\underline{a}$ which exactly satisfies:

$$X.\underline{a} = \underline{D} \tag{6.8}$$

But it is possible to make $X.\underline{a}$ close to D by choosing vector $\underline{a}$ such that the sum of the squared differences between the elements of $\underline{W}$ and the corresponding elements of $\underline{D}$ are minimised over all possible choices of elements for $\underline{a}$. Multiplying both sides of equation (6.8) by X^T (transformed) gives:

$$X^T.\underline{D} = X^T.X.\underline{a} \tag{6.9}$$

and noting that since the dimensions of X and X^T are *R by (N+1)* and *(N+1) by R* , $X^T.X$ will be a square matrix of dimension *N+1 by N+1*. When this matrix is non-singular, we can write:

$$\underline{a} = (X^T X)^{-1} X^T \underline{D} \tag{6.10}$$

or:

$$\underline{a} = X^{\#}.\underline{D} \tag{6.11}$$

where the *N+1 by R* matrix $X^{\#}$ is defined as the 'pseudo-inverse' of the *R by N+1* matrix X:

$$X^{\#} = (X^T X)^{-1} X^T \tag{6.12}$$

Clearly, it cannot be expected that defining vector $\underline{a}$ by equation (6.11) satisfies equation (6.8) except when X is square or, in general when $R<N+1$. Neither of these cases are of interest here as they mean that there are too few training circuits (R training circuits for N parameters) to make the learning worthwhile. But let

$$E = ||X\underline{a} - \underline{D}||_2 \tag{6.13}$$

$$= (X\underline{a} - \underline{D})^T(X\underline{a} - \underline{D}) \tag{6.14}$$

$$= \sum_{i=1}^{R} (W_i - D_i)^2 \tag{6.15}$$

This is sum of squared differences between elements of $\underline{W} = X.\underline{a}$ and the corresponding elements of $\underline{D}$ and is a useful measure of the difference between $X.\underline{a}$ and $\underline{D}$, or in other words the estimation error.

For any vector $\underline{x}$ of length M say, the 'norm' is:

$$||\underline{x}||_2 = \underline{x}^T \underline{x} = x_1^2 + x_2^2 + x_3^2 + \ldots + x_M^2 \qquad (6.16)$$

Multiplying out equation (6.14) gives:

$$\begin{aligned} E &= \underline{a}^T X^T X \underline{a} - \underline{a}^T X^T \underline{D} - \underline{D}^T X \underline{a} + \underline{D}^T \underline{D} \\ &= \underline{a}^T X^T X \underline{a} - 2\underline{a}^T X^T \underline{D} + \underline{D}^T \underline{D} \end{aligned} \qquad (6.17)$$

To find the value of the estimator coefficient vector $\underline{a}$ that gives the minimum value of E and therefore the best possible estimation, we partially differentiate E with respect to each coefficient of $\underline{a}$ and then set each of the N+1 expressions to zero. In vector notation, it may be verified that the resulting gradient vector:

$$\begin{aligned} \nabla \underline{E} &= \partial \underline{E} / \partial \underline{a} = 2 X^T X \underline{a} - 2 X^T \underline{D} \\ &= \underline{0} \end{aligned} \qquad (6.18)$$

to minimise E, where $\underline{0}$ is the *N+1 by 1* zero vector. Therefore, to minimise E,

$$X^T X \underline{a} = X^T \underline{D} \qquad (6.19)$$

which means that:

$$\begin{aligned} \underline{a} &= [(X^T X)^{-1} X^T] \underline{D} \\ &= X^{\#} \underline{D} \end{aligned} \qquad (6.20)$$

MATLAB provides the function '*pinv*' for calculating the pseudo-inverse and computes it in a robust way using singular value decomposition. Therefore 'pinv' can be relied upon for very large matrices.

6.5 Application of the Linear Estimator

The linear estimator outlined above, together with a threshold detector will form the classifier as required for implementing Statistical Blockade. In N+1-dimensions, the linear estimator may be seen as a hyperplane which divides the parameter space into 2 regions as illustrated in Figure 6.2 for two dimensions:

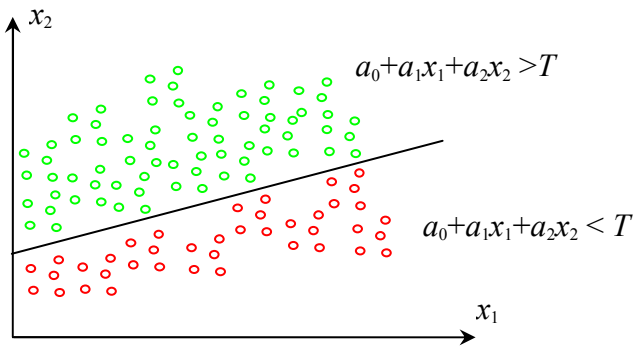


Figure 6.2: Linear classifier dividing 2-D parameter space (x_1, x_2) into 'tail' region (red) and 'body' (green) where threshold is T.

As will be seen, the estimator may be refined recursively as more and more simulations are carried out and results of simulations are fed back and used as additional training data to update the estimator. This is illustrated in Figure 6.3 for a 2D classifier with estimator updated recursively and becoming more and more accurate for the tail.

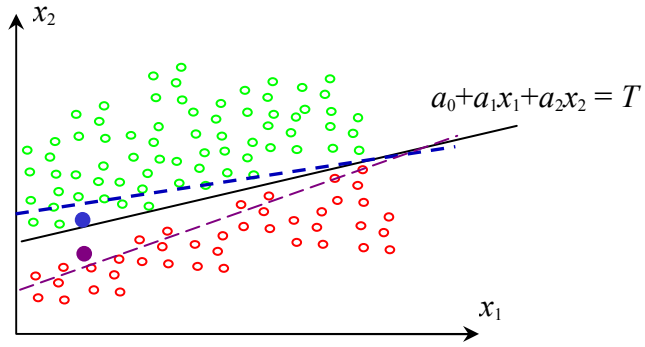


Figure 6.3: Effect of recursive adaptation of estimator coefficients.

This simple classifier should work well with Highly Replicated Circuits (HRCs), where the HRC components could be standard cells. The author's SPICE harness RandomLA introduced in Section 4.4.3 has been modified with an implementation of Statistical Blockade. As before, it requires a "seed" circuit with N parameters {x_1, x_2,

..., x_N} which may be capacitor values, transistor parameters, etc. There is a training facility, an evaluation facility and an execution facility. The training facility generates *R* randomised copies of the seed circuit, each with randomized parameters. It then runs SPICE for all of these training circuits to obtain the measurements of interest for each circuit. We consider just one measurement here, and refer to it as *Di* for each circuit *i*. This allows vector $\underline{D}$ and matrix *X* as defined by equations (6.3) and (6.6) respectively to be populated.

The pseudo-inverse method as described above is then used to calculate the coefficients, $\underline{a}$, of the linear estimator. An appropriate pdf is fitted to the *Di* measurements, the mean and standard deviation are determined and then a theshold value is specified to mark the 'start of the tail'. This threshold should be defined on the conservative side of any 'yield threshold' values that will determine the viability of circuits. This will allow for limitations in the linear estimator at the cost of having to SPICE-analyse slightly more circuits than necessary. It is now possible to calculate an initial probability of the measurement of interest (*Di*) of a randomized circuit being in the tail. The complementary error function, *erfc*, available in MATLAB, is used when the distribution is assumed to be Gaussian. This initial probability is not expected to be very accurate as it is obtained just from the measurements of the training set. This terminates the initial training phase.

An evaluation facility is provided for plotting the estimated measurement of interest against its known true value (calculated by SPICE) for a set of randomized circuits which are different from the training set. The scatter graph shown in Figure 6.4 was obtained by plotting estimated values of overall delay against the true values for the behavioural model of a binary full adder circuit as presented in Section 6.8, when the estimator has nine parameters. There were 300 training circuits and 300 different evaluation circuits. The value of 'Pearson' correlation obtained between estimated and true values of delay is clearly high and was found to be 0.9993. With a sample size of 300, the 'p-value' was found to be about 0.001 which means that the probability of getting this value of correlation by chance for a sample of 300 circuits is less than 0.1%. The 95% confidence limits on this statistic, as calculated by the MATLAB function '*corrcoef*' was found to be about 0.9993 to 0.9994.

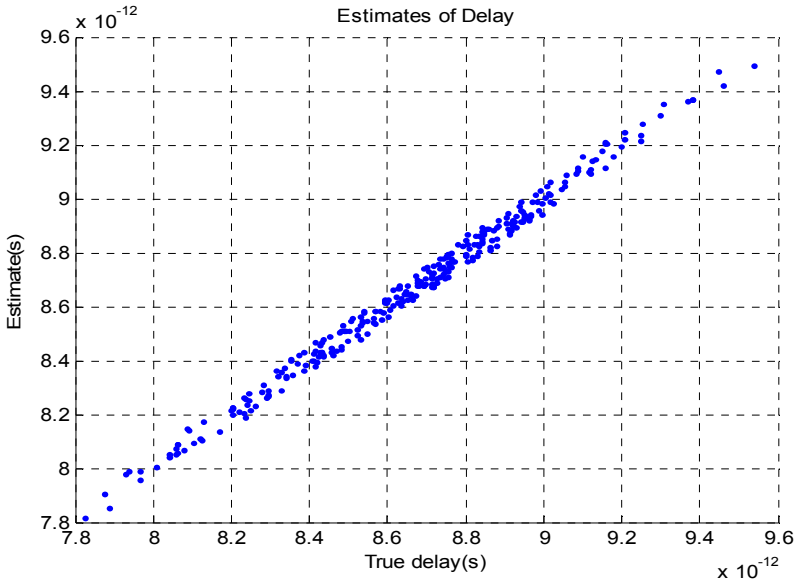


Figure 6.4: Evaluation of 9th order linear estimator of delay in a BFA

6.6 Execution Phase of Statistical Blockade

The circuit yield failure predictions are expected to be 'rare' in the sense that they are on the 'tails' of probability distributions for circuit quantities such as overall delay. Reliable estimates of these failures by conventional Monte Carlo techniques require very large numbers of randomised input vectors. For example, a 0.2% failure rate means that a failure may be expected only about twice in 1000 simulations. There are not enough failures, for a reliable estimate of such a small probability.

The SB approach is to concentrate on parameter vectors that are likely to generate the 'rare events' of failing circuits, and 'filter out' ('blockade') or disregard the vectors that are likely to produce good circuits. The bias introduced in the input data must be taken into account in the analysis. It is argued that the computational complexity involved in introducing the bias, and compensating for it, is much less expensive than performing many SPICE simulations. The complete implementation strategy is represented by the block diagram in Figure 6.5. It is initiated by firstly supplying a 'seed' netlist which specifies the basic circuit with its sub-blocks and then indicating which parameters are to be randomized and supplying the statistics (mean, standard deviation, etc.) of the required randomization for each parameter.

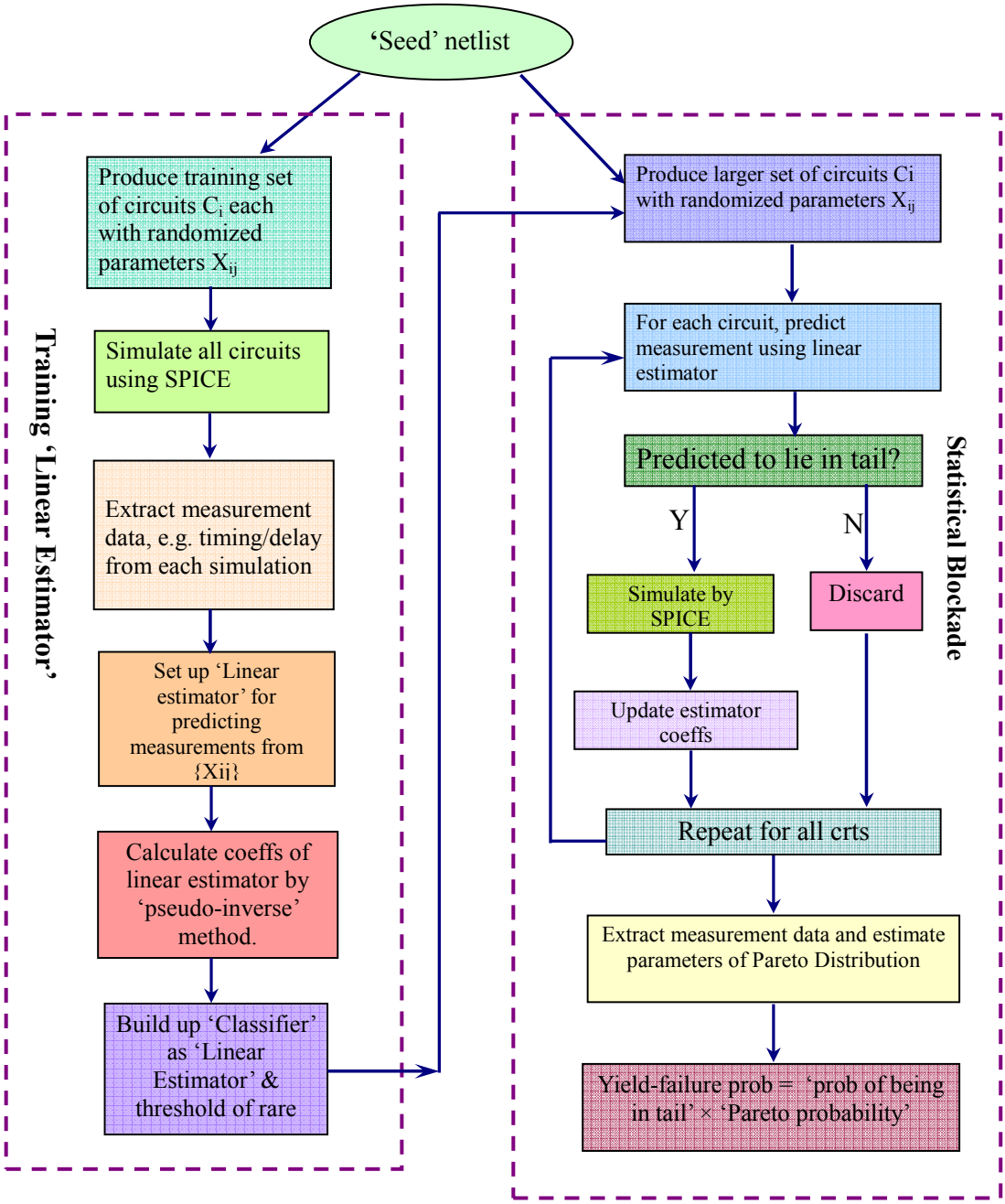


Figure 6.5: Complete SB procedure as implemented by RandomLA.

The subsequent actions are based on a standard classification technique used in machine learning and data mining [30]. The basic idea is to simulate an initial training set of circuits using a relatively small set of circuit copies with randomised parameter vectors. An estimator and classifier are then initially trained on this data as explained in the previous section. The classifier is subsequently refined as more and more simulations are carried out. The classifier is a type of 'vector support machine' and it is 'recursive' as results of simulations are fed back to an updating procedure. The classifier becomes the 'blockade filter'. The complete implementation can be divided into four parts:

(i) The initial training and evaluation of the linear estimator as described above.

(ii) The generation of a much large set of randomized versions of the circuit, and the use of a classifier to allow the program to 'block' the versions that are not likely to be within the tail. The classifier consists of the linear estimator followed a 'threshold detector' which compares the estimated value of any measurement of interest with a 'start of tail' parameter. Only the circuit copies with a measurement of interest that is estimated to fall within the tail of the so far estimated distribution will be unblocked and submitted to SPICE.

(iii) The 'recursive' refinement of the estimated distribution and the coefficients of the linear estimator as more and more SPICE simulations are carried out. The effect of the biased sampling (i.e. selecting more 'tail' circuits than would naturally occur) is corrected, to a degree, using the estimations for the 'blocked circuits' on the assumption that inevitable inaccuracies in these estimations will not be as critical in the 'body' of the distribution as they would be in the tail. The effect of such inaccuracies is, anyway, reduced by the Pareto fitting described in the next paragraph. When a sufficient number of non-blocked 'tail' has been analysed, a second estimator may calculated, again using the 'pseudo-inverse' technique outlined in Section 6.4. The second estimator will more accurate than the original estimator for the tail and may be used for more accurate blocking. Further recursion is clearly feasible with a third stage and so on. The use of recursion can allow the defined 'start of tail' parameter to be gradually moved further away from the mean: typically from 2 to 3 and then to 4 or more standard

deviations. Through recursion, we can thus get more accuracy in more extreme parts of the tail. Computational savings are possible by updating rather than recalculating the estimator coefficients from scratch with the recursion, but this feature is not currently implemented.

(iv) The fitting of a Pareto Distribution (PD) to the measurements obtained from the non-blocked ('statistical tail') versions of the circuit. This is necessary because, the Gaussian (or other pdf) assumption will be least reliable for the tails and dominated by values occurring, or estimated to occur, within the body of the pdf. Also, as already mentioned, the non-tail circuit values are estimated which introduces some inaccuracies in the general pdf shape. Few measurements will occur in the 'far tail' even when large numbers of circuits are generated. Therefore the use of PD fitting to the rarely occurring 'tail circuits' allows the prediction of likely yield without the very large number of circuit simulations that would be required with traditional MC analysis.

6.7 Fitting a Pareto Distribution

Fitting Pareto distributions to the tails of Gaussian distributions is a commonly used procedure. The pdf of a Pareto distribution with parameters k and θ is given by equation 6.22 and plotted in figure 6.6:

$$pdf\,(d) = \begin{cases} 0 & : d < \theta \\ \dfrac{k\,\theta^{k}}{d^{k+1}} & : d \geq \theta \end{cases} \tag{6.21}$$

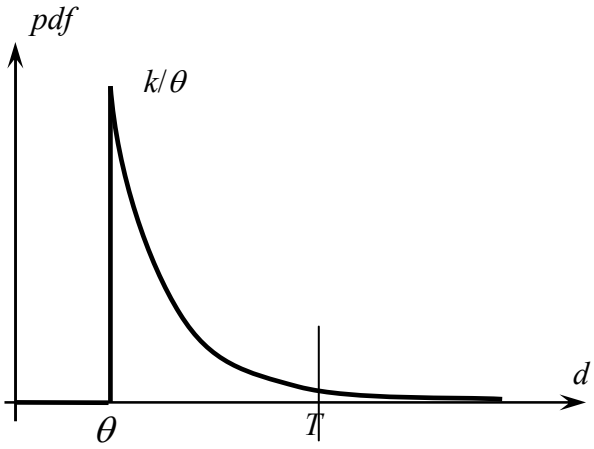


Figure 6.6: PDF of Pareto distribution

θ determines the start of the tail, and k determine the shape of the distribution. Taking θ as defined for the blockade filter, maximum likelihood estimation calculates k to be:

$$k = \frac{N_U}{\sum_{i=1}^{N_U} \log(D_i / \theta)} \tag{6.22}$$

where there are N_U unblocked measurements $D_1, D_2, \ldots, D_{NU}$

It can now be deduced that the Pareto-estimated failure probability, conditional on the measurement being in the tail, is as follows:

$$\begin{aligned} \text{p(failure} \mid \text{tail)} &= \int_T^\infty pdf\,(d)dd \\ &= (\theta / d)^k : d > \theta \end{aligned} \tag{6.23}$$

Therefore, from the Pareto distribution we can estimate probability of delay being greater than some threshold T, conditional on it being in the tail and we call this the 'Pareto probability'. The absolute failure probability is:

(Pareto probability) × (probability of being in the tail) (6.24)

The probability of being in tail is estimated from the Gaussian pdf fitted to the training circuits during the RandomLA training phase. Figure 6.7 illustrates Pareto fitting to a set of measurements whose training distribution and histogram are also shown.

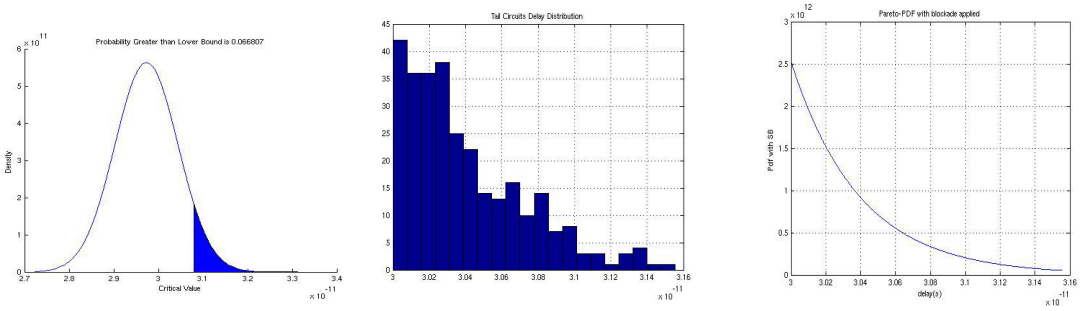


Figure 6.7: Illustration of Pareto fitting procedure.

6.8 Measurements and Evaluations

To illustrate the computation time savings that may be achieved when synchronous and asynchronous circuits employing SBCB blocks are statistically analysed by MC techniques with SB, a frequently used handshaking component in the asynchronous control circuits produced by the BALSA design package [5] mentioned in Chapter 2, i.e. a C-element [11], was considered. The intention was to compare the speed and accuracy achievable with that of straightforward MC analysis. A binary full adder, using NAND gates as building blocks, and a 4-Phase Bundled Data Muller Pipeline and a Muller 'ring', each using the C-element as building blocks, were also used as test circuits. The use of SB to analyse the switching delay of the output of a single C-element was found to reduce the computation time by about 98.5%, when the start of the distribution tail was defined to be two standard deviations (2σ) from the mean.

The accuracy of the linear estimator obtained with the start of tail defined 2σ from the mean is illustrated by the scatter-graph in Figure 6.8. It is from applying SB to 1000 copies of binary full adder circuit. The blue points represent the

measurements which are in tail and not blocked.

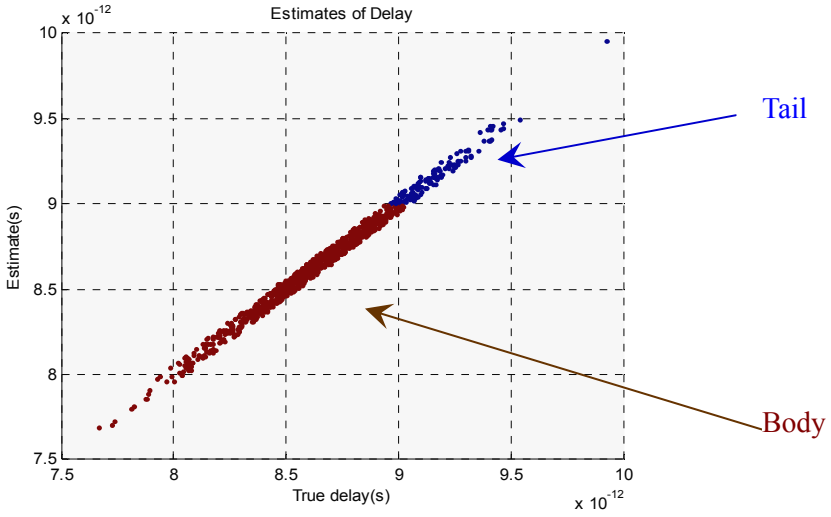
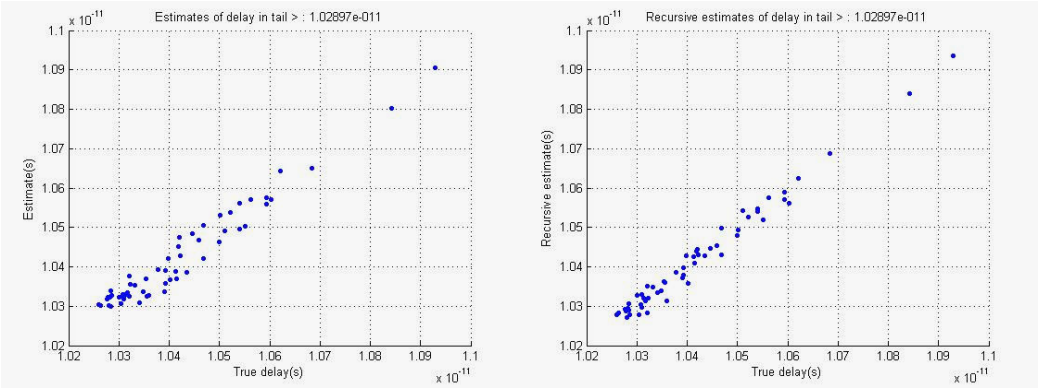


Figure 6.8: Accuracy of linear estimator . Tail defined to start at 9e-12s. Classification errors out of 1000: wrong to block: 4, wrong not to block: 2.

To obtain accurate predictions of behaviour further from the mean, recursion was employed to refine the accuracy of the original estimator using the results of non-blocked simulations. Figure 6.9(b) shows the effect of recalculating the estimator from the tail points, shown in Figure 6.9(a), as identified by the original 2σ estimator. The experiment is applying recursion to the tail points obtained from blocking 1000 copies of the 4-Phase Bundled Data Muller Pipeline circuit shown in Figure 6.11.



(a) Original (b) Refined

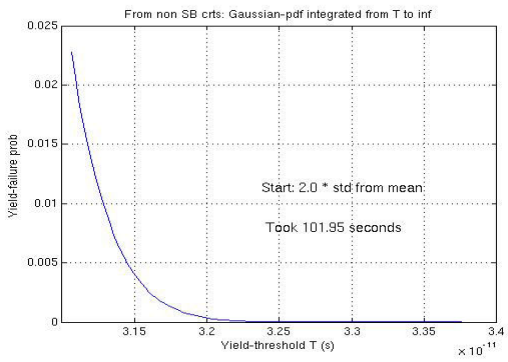
Figure 6.9: Refining linear estimator by recursion

The estimated failure probability distributions shown in figures 6.10(a) and (b) were obtained for the behavioural model of a single C-element, as shown in figure 2.6, with parameters for each behavioural gate model extracted from the transistor-level simulations outlined in Section 4.4.3, based on the 35nm transistor model set provided by RandomSPICE. The seed file is 'ngswnandceln.seed'. Figure 6.10(a) was obtained by traditional MC analysis, and plots the yield failure probability against yield threshold, showing how the failure probability reduces as the permissible delay increases. The threshold value is measured in seconds relative to a reference delay 200 ps from the start of the analysis where the trigger occurs at 100 picoseconds from the start. The graphs show the distribution tails only, which are assumed to start at two standard deviations (i.e. 2 ×0.657 ps) from the mean which is 29.7 ps relative to 200 ps. Therefore the graphs show a time-scale from 31 ps (relative to 200 ps) and extending over 5 standard deviations. The graph shown in Figure 6.10(b) was obtained from MC analysis with SB and a Pareto fit to the tail assumed to start at two standard deviations from the mean. Referring to equation 6.23, θ was set to 31 ps and k (= 126.9) was calculated according to equation 6.24 from the unblocked measurements of delay. Figure 6.10(b) may be seen to be close to figure 6.10(a) as obtained from traditional MC analysis with much greater computation. To assess the quality of fit, the two graphs are shown on the same axes in figure 6.10(c).

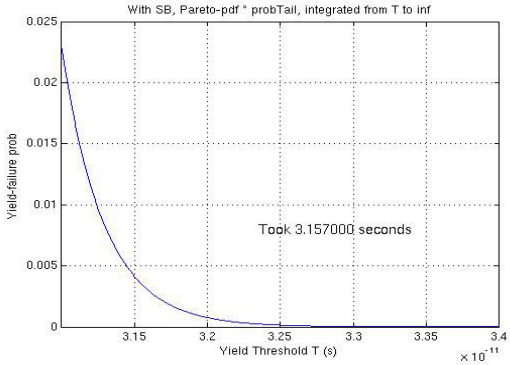
It may be seen that the maximum difference in yield threshold delay between the two graphs for any yield failure probability is about 0.06 ps seconds, which is about 0.1 standard deviations. A more useful measure of difference is the maximum difference in yield failure probability. This cannot accurately be deduced from the graphs, but a re-sampling of the data plotted in one of the two graphs (since the sampling instants are different) revealed that this maximum difference occurred at a yield threshold of 31.23 ps, and is equal to a probability difference of 0.002. This represents a discrepancy of approximately 14.2 % from the probability 0.0129 predicted by the non-SB Monte Carlo simulation being used as a reference.

Since a possible source of this discrepancy is the quality and suitability of the Pareto fit, some investigations were carried out. It was observed that one source of

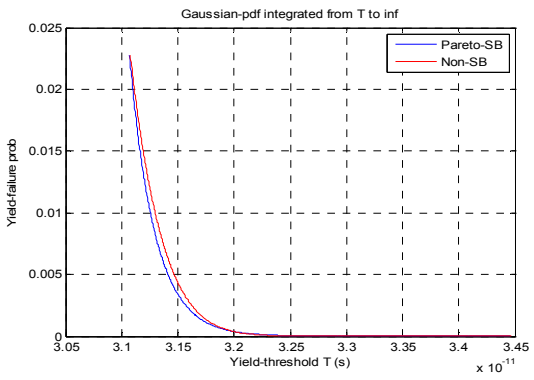
the discrepancy was the difference in mean and standard deviation of the Gaussian fits to the delay measurements produced on the one hand by the non-SB MC simulations ('meanNB' and 'sigmaNB'), and on the other by the Training procedure ('meanTR' and 'sigmaTR'). These are used to determine the 'start of tail' parameter at two standard deviations from the mean. The more accurate estimations 'meanNB' and 'sigmaNB' are available for producing the comparisons since a computationally expensive non-SB will have been carried out for test purposes. But in reality, only the less accurate 'meanTR' and 'sigmaTR' estimates (based on far fewer randomised circuits) will be available to the SB version, and were therefore used in the comparison.



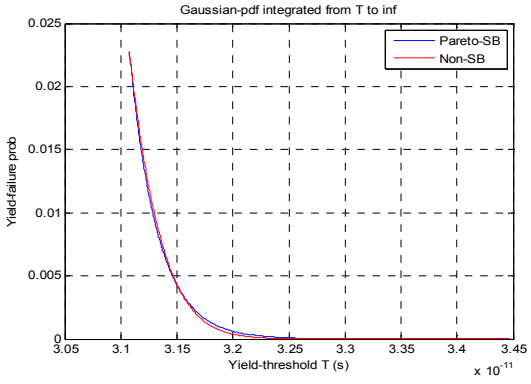
(a) Without SB



(b) With SB



(c) Comparison of (a) and (b)



(d) More accurate comparison

Figure 6.10: Failure probability for a 'C-element' realisation from 500 versions: (a) without SB, (b) with SB (2σ from mean), (c) Comparison of (a) and (b), and (d) Comparison with more accurate estimates of mean and std-dev used for Pareto-SB.

As a test, ‘meanTR’ and ‘sigmaTR’ were relaced by ‘meanNB’ and ‘sigmaNB’, thus eliminating two sources of discrepancy and allowing the suitability of the Pareto fit to be seen more clearly in Figure 6.10(d)). It was found that the two graphs did indeed become closer, the maximum discrepancy being 0.00061 in a non-SB yield probability measurement of 0.0136, i.e. a 4.5 % discrepancy. A conclusion from this investigation is that the Pareto fit is capable giving a reasonable approximation to a Gaussian tail, incurring error likely to be less than that resulting from other statistical estimates. Also, we concluded that there is scope for increased accuracy in the SB estimations, for example by updating the estimates of mean and standard deviation as statistical analysis proceeds, or perhaps by not relying on these measurements for defining the start of tail.

With a more accurate estimator, the ‘start of tail’ parameter may then be redefined as two or even three standard deviations from the mean to obtain even greater time saving since even fewer circuits need to be analysed. This increases the possibility of finding measurements yet further from the mean, i.e. ‘rarer events’, in reasonable computational time, and allows a yet more accurate estimation of the statistics of the ‘far tail’.

Table 6.1 summarises the computation time-savings that were obtained by applying SB to the statistical variability analysis of three of the circuits mentioned above. The computation was carried out on a standard desk-top PC with a dual core 2.8 GHz Intel processor. A MATLAB program that harnesses an implementation of SPICE carried out the randomisation, implementation of SB and statistical analysis. It may be seen that the most significant computation time saving, 98.7 %, was achieved for a 4-Phase 3-stage Bundled Data Muller pipeline 'ring', figure 6.11, with the start of the tail defined at two standard deviations from the mean. This table disregards the time taken for the linear estimator training phase which is, in fact, just a small proportion of the overall simulation time.

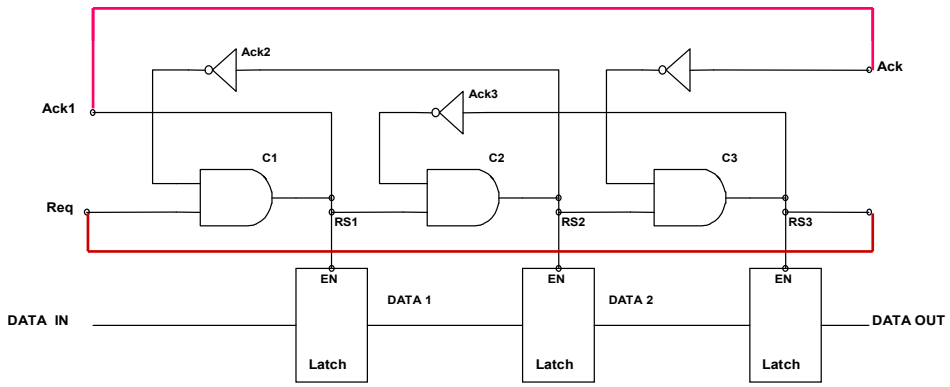


Figure 6.11: 4-Phase 3-stage Bundled Data Muller pipeline 'ring'.

Circuit	**Binary Full AddAdder 9 parameters**		**C-element 12 parameters**		**Muller Pipeline Ring 21parameters**	
Start of tail	**1.5σ**	**2σ**	**1.5σ**	**2σ**	**1.5σ**	**2σ**
1000 circuit without SB	215.99s	221.34s	250.05s	288.51s	949.59s	1003.9s
1000 circuit with SB	6.75s	3.96s	7.63s	4.24s	27.17s	13.15s
Time saving	96.9%	98.2%	96.94%	98.5%	97.1%	98.7%

Table 6.1: Time Saving Illustrated by Comparing Simulations with SB to Simulations without SB.

6.9 Conclusions

Statistical blockade, with a linear estimator and recursive updating, has the potential to simplify the estimation of rare event statistics as required when estimating circuit failure probabilities and anticipated fabrication yield. This should make the simulation of much larger circuits computationally feasible. A MATLAB implementation of a SPICE harness has been developed to implement SB in three phases: initial training, initial evaluation and execution with recursive training/adaptation. Some experiments have been performed to show that the basic approach can be made to work and that there are demonstrable benefits even for simple circuits. The approach remains compatible with the use of the open source

NGSPICE and the use of parallel programming.

Further computational savings are possible by updating rather than recalculating the estimator coefficients from scratch, at each stage of the SB recursion, but this feature is not currently implemented. Also, the nature and degree of the inaccuracy introduced into pdf shapes by the fact that SB estimates non-tail circuit values, remains to be investigated. It is argued that such inaccuracy is not likely to be critical as the pdf estimations serve mainly as a guide to allow the tails of distributions to be investigated accurately. Nevertheless it would be useful to know how accurate the pdf shapes thus obtained really are.

It is possible that quasi-Monte Carlo techniques can be combined with statistical blockade by replacing the pseudorandom generator with a low-discrepancy sequence generator. This idea will be addressed in the next chapter.

Chapter 7

Computation Reduction by Quasi Monte Carlo Techniques

7.1 Introduction

This chapter investigates the use of 'low-discrepancy' sampling to achieve further efficiency improvements, over what was achieved in earlier chapters, with Monte Carlo circuit simulation. Low-discrepancy sampling is the basis of 'quasi Monte Carlo' (QMC) techniques as often applied to multi-dimensional integration, therefore this approach to circuit simulation may be referred to as 'quasi–Monte Carlo' simulation. QMC methods are modified Monte Carlo methods where the input vectors are not totally random, but are to a degree deterministic in that they conform to 'low-discrepancy sequences' [15][36]. A low discrepancy sequence is a sequence of N-dimensional vectors which covers a finite space more uniformly than is achieved by N-dimensional vectors of independent uniformly distributed random elements. The discrepancy of a sequence of vectors is a measure of how the number of points they define in a multi-dimensional cube varies with the position and size of the cube. If the discrepancy is low, the same sized cube will always contain approximately the same number of points wherever it is located, and the number of points will be proportional to the volume of the cube. A sequence of vectors of independent uniformly distributed random elements does not give low discrepancy when the dimensionality is high. It is known that the use of low discrepancy vector

sequences can achieve significant speed gains over standard Monte Carlo integration techniques by reducing the number of input vectors needed for a given accuracy [38]. Similar gains are anticipated when QMC is used for statistical circuit simulation.

7.2 Quasi-Monte Carlo Simulation

The Monte Carlo simulation procedure for circuits as adopted in this thesis may be formulated as simply estimating the statistics of some function D say of the random vector $\underline{x}$ which characterizes the expected variation of parameters within the circuit. It is most commonly assumed that the parameters are Gaussian distributed, meaning that the multivariate pdf of $\underline{x}$ will be:

$$pdf(\underline{x}) = \frac{1}{\sqrt{(2\pi)^N \det(C)}} \exp\left(\frac{1}{2}(\underline{x}-\underline{m})^T C^{-1}(\underline{x}-\underline{m})\right) \tag{7.1}$$

where N is the dimension of $\underline{x}$, $\underline{m}$ is the vector of mean values and C is the N by N covariance matrix which specifies any inter-dependency between elements of $\underline{x}$. For a set of R vectors $\underline{x_k}$, C is defined by equation 5.1. If C is the unity matrix, the variations in all parameters of $\underline{x}$ are independent. Otherwise, if C is expressed in the form

$$C = L^T . L \tag{7.2}$$

either using Cholesky decomposition or the eigenvalue/eigenvector approach introduced in Chapter 4 (or otherwise), the MATLAB statement:

$$Y = L \; . \; randn(N,R) \tag{7.3}$$

generates R vectors of multivariate Gaussian random numbers with the required pdf . To obtain independent normally distributed random vectors, MATLAB transforms the vectors $\underline{u}_1$, $\underline{u}_2$, . . . , $\underline{u}_R$ generated by its uniform pseudo-random generator as follows:

$$\underline{v}_k = \phi^{-1}(\underline{u}_k) \tag{7.4}$$

where ϕ is the cumulative pdf of the required distribution. For Gaussian, ϕ^{-1} is provided as a function '*norminv*', and a very large number of other distributions are also catered for.

The QMC method can now be investigated by replacing the $\underline{u}_1$, $\underline{u}_2$, . . . , $\underline{u}_R$

vectors by low discrepancy vectors chosen to more effectively sample the $[0,1]^N$ hypercube.

It must be remembered that the vectors obtained from a 'pseudorandom number generator' (PRNG) should have a very long repetition period, orders of magnitude larger than the total number required. To reduce the possibility of having unwanted correlation among QMC vectors, it is standard practice to divide the generated sequence into sub-streams and to allow the next sub-stream to be started by skipping all intermediate values [95].

7.3 Low-Discrepancy Sequences

Low-discrepancy sequences or 'quasi-random' or 'sub-random' sequences, are commonly used to replace uniformly distributed random sequences. They are not random or pseudo-random, but they have properties that allow them to be used as random sequences, their lower discrepancy being an advantage.

Figure 7.1 was obtained for two and three-dimensional vectors that may be used for Monte Carlo integration, for example. Figure 7.1 (a) and (c) show 1000 points that were generated using the standard MATLAB uniform pseudorandom number generator. Figure 7.1(b) and (d) show 1000-point "deterministic" sequences from the 'Sobol' algorithm [95] as provided by MATLAB. It may be seen that the coverage is more 'uniform' or even for the Sobol' sequences and this would be more evident if we could somehow visualise higher dimensional versions. With uniform, there is more 'discrepancy' than with Sobol in the way the random points are laid out from one region to the other. The degree of discrepancy may be quantified and measured.

A comprehensive discussion on uniformity and discrepancy is given in [22]. The classic problem is to distribute the points within a hypercube (call it a hyper-box) such that any smaller hypercube within it contains a number of points proportional to its hyper-volume. This problem of 'measure theory' was solved by Roth [33]. Discrepancy is defined as the worse deviation that occurs in the percentage of points in any sub-volume divided by the exact percentage of points that would occur for a perfectly even distribution.

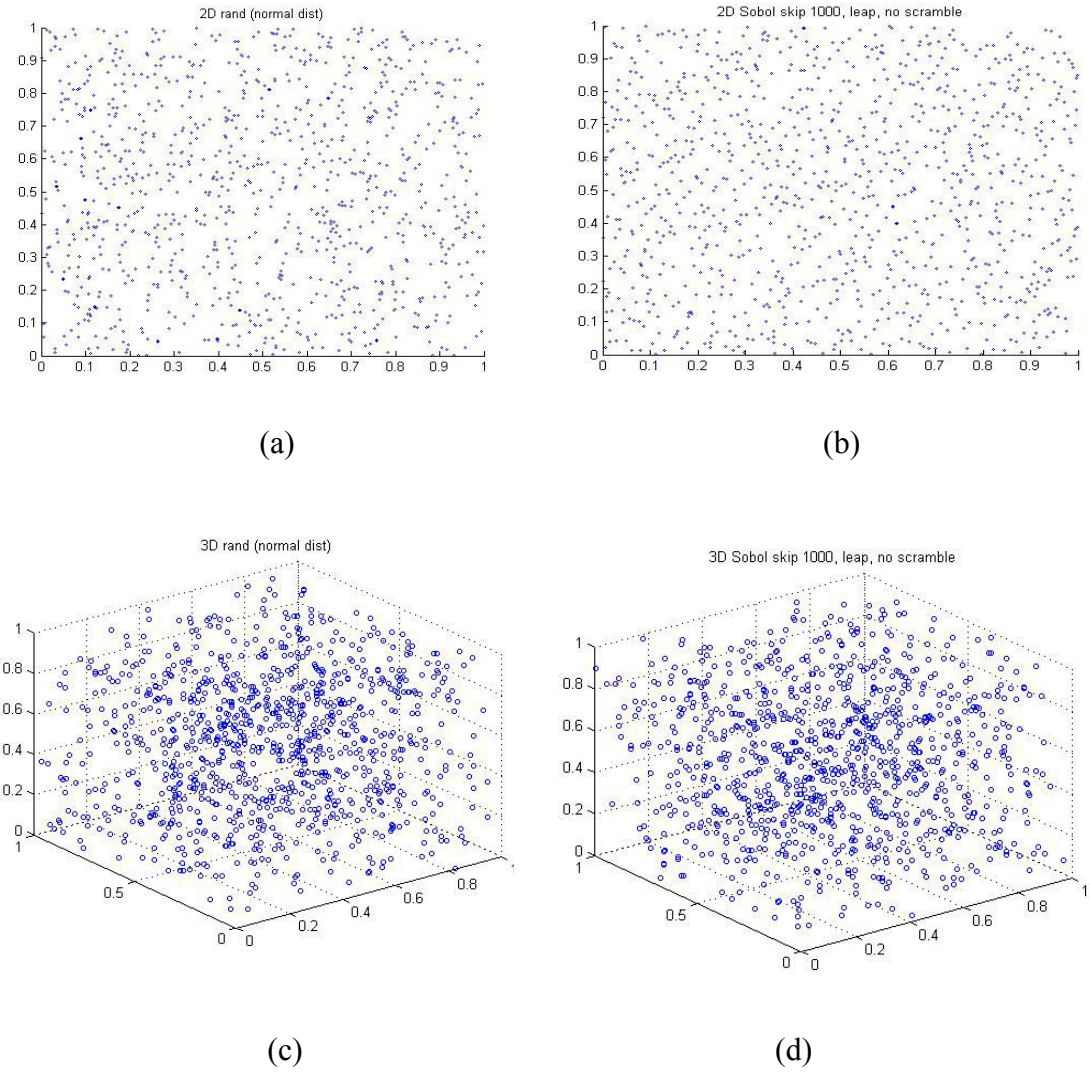


Figure 7.1: Distributions of point-sets, (a) 2D pseudo-random points, (b) 2D Sobol' points, (c) 3D pseudo-random points, (d) 3D Sobol' points.

Assume a set of points $\{\underline{x}_i\}$ are scattered throughout an N-dimensional unit hyper-box, whose volume is equal to 1. Let h be a smaller hypercube as illustrated in Figure 7.2. The discrepancy of the sequence $\{\underline{x}_i\}$ with $i = 1,2, \ldots, R$ is defined by:

$$D_R = \max_{h \in [0,1)^N} \left| \frac{r(h)}{R} - V(h) \right| \tag{7.5}$$

where $r(h)$ is the number of points (vector tips) that lie within 'h' and $V(h)$ is the

volume of this smaller hyper-cube. D_R is the maximum discrepancy over all hyper-cubes '*h*' that fit within the *N*-dimensional unit hyper-box. This concept is associated with theories of worst-case clustering and spatial gaps [95].

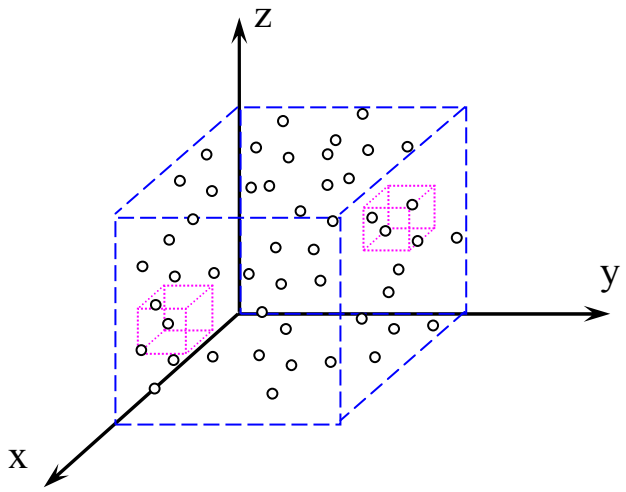


Figure 7.2: 3-dimensional hypercube and sub-hypercube with scattered points.

Uniformly distributed sequences within *N*-dimensional unit hyper-boxes have the property:

$$\lim_{R \to \infty} D_R = 0 \tag{7.6}$$

A sequence of *R* vectors with 'discrepancy constant' K_R is a sequence which satisfies the condition:

$$D_R \leq K_R \frac{(\log R)^N}{R} \tag{7.7}$$

The constant K_R is a measure of how good the sequence is; the lower the better. Researchers are still devising new sequences with ever lower constant values of K_R for high-dimensional hyper-boxes. Thompson explored the use of 'Halton sequences' [119] for the integration of a three-dimensional exponential function, and found that the constant K_R for 300 vectors was about 6.5.

Van der Corput devised one-dimensional low-discrepancy sequences (LDS) in 1935 [119]. In 1960, Halton published the first method for constructing an LDS in arbitrary dimensions [120], thus extending a previous method by Hammersley [69] which uses one-dimensional Van der Corput sequences. However, Halton sequences have poor uniformity in high dimensions because of an undesirable feature of the Van der Corput sequence for large bases. Therefore, Halton sequences are not suitable for our application. Sobol sequences [121] and Faure sequences [119] are both much better than Halton sequences and allow the practical use of QMC for large dimensional simulations. Both these sequences generalize the Van der Corput one-dimensional concept.

Halton (1960), Faure (1982), Sobol (1967), and Niederreiter (1987), are the best known low-discrepancy sequences, but new ones were still being proposed in 1997 [122]. The generation process sub-divides the unit hyper-box into smaller hyper-cubes of constant volume with all faces parallel to the faces of the hyper-box. A number of points are put into each hypercube, the grid is refined, and the process continues with smaller hyper-cubes.

Faure and Sobol sequences generate all pseudo-random numbers from just one prime number base and reorder the basic QR vectors within each dimension. Reordering eliminates possible correlations in high-dimensions. The Halton sequence uses a different prime base to generate the quasi-random numbers for each dimension. The base specifies the Galois field over which the required primitive polynomials are generated; for Sobol sequences this is always two. The higher the base, the higher the computational time and the longer the cycle period. For a Faure sequence the base is taken as the smallest prime number greater than or equal to the number of dimensions.

As reported by Galanti & Jung [123], start-up problems can occur with all methods especially in high-dimensions. To eliminate such problems, Faure suggests discarding the first $(b^4 - 1)$ points, where b is the base. As reported by Galanti & Jung, generating Faure sequences is much slower than generating Halton and Sobol sequences. Sobol sequences are simpler and faster to generate than Faure sequences due to the use of base 2 for all dimensions. However, the reordering mechanism is

more complex and requires the coefficients of irreducible primitive modulo 2 polynomials. Galanti & Jung reported that Sobol sequences remain effective at very high dimensions. Thus, they outperform both Faure and Halton sequences in this respect.

7.4 MC and QMC Convergence rates

As mentioned in Section 4.2, traditional Monte Carlo methods with statistically independent input vectors are argued to have a convergence rate proportional to $R^{-½}$ which is independent of dimensionality N. Quasi-Monte Carlo (QMC) methods are claimed to have a much better rate of convergence approaching proportionality with R^{-1} in optimal cases [115]. A theoretical upper bound for the convergence rates of multivariate low discrepancy sequences is reported to be proportionality to $(\ln R)^N/R$ [115], where N is the number of dimensions. QMC performance therefore decreases with the dimensionality N.

7.5 Implementation of QMC Circuit Simulation

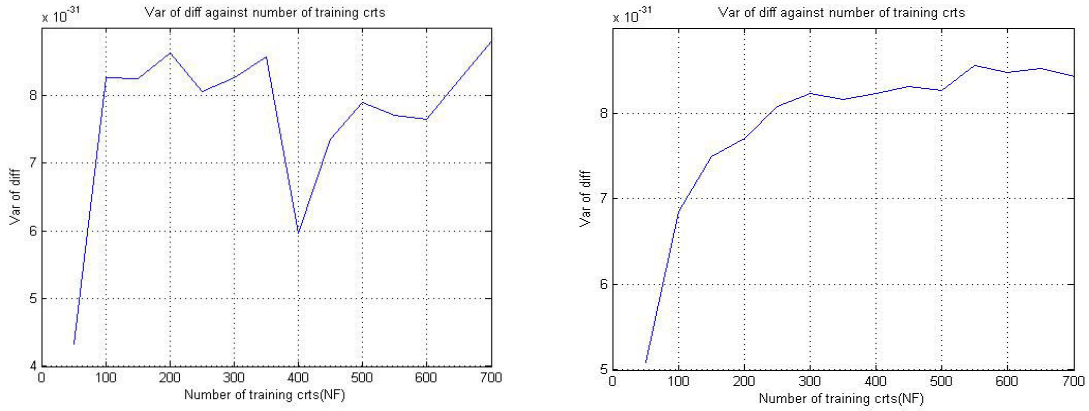
Uniformly distributed numbers in the interval (0, 1) can be generated as pseudo-random numbers or quasi-random numbers and the variables for all other distributions may be derived from these by means of the appropriate cumulative distribution function inversion. In practice the range must be restricted from (0,1) to (α, 1-α) for small positive α otherwise Gaussian variables very far from the mean may occur with QMC and cause numerical instability. Examining a Gaussian pdf graph reveals that taking $\alpha=10^{-6}$ or 10^{-11} restricts the transformed variables between 5 or 7 standard deviations respectively from the mean.

The MATLAB functions ‘haltonset’ and ‘sobolset’ are provided for constructing initial sequences of N-dimensional quasi-random vectors with the required properties. To avoid the undesirable effects of any correlations, especially in the initial segments, the random sequence obtained can be required to skip, leap over, or scramble values in the sequence as generated. Scrambling reduces correlations while also improving uniformity. These sequences use different prime bases to form

successively finer uniform partitions of the unit interval in each dimension. Latin hyper-cube sequences, as used by SPICE, are generated by the '*lhsdesign*' function. Strictly, these are not LD sequences, but they nevertheless produce uniform samples of a sort.

As suggested in Section 7.2, the idea is to use a low discrepancy sequence generator to replace the uniform random number generator as the source of randomisation in both the training and the recursive SB phases of RandomLA. The choice of LDS will be 'Sobol'. First, we present an example that compares the effect of using QMC rather than MC for training the linear estimator. Then we investigate the effectiveness of QMC for MC simulation with and without Statistical Blockade.

To provide comparison for training, an SRAM32X1 array circuit was taken as an example that is described in more detail in Chapter 8. The convergence of the linear estimator training using MC and QMC was analysed and compared. As shown in Figure 7.3(b), the QMC training converges more quickly and smoothly to an estimator producing the minimum 'variance of difference' attainable, approximately 9×10^{-31}, than MC training with pseudo-random vectors, which is shown in figure 7.3(a). The measure is just the variance of the prediction error. With QMC training, the variance becomes close to optimum with about 300 training circuits, whereas the number needed for MC is about 700 training circuits. The reason for this improved training is probably the more reliable coverage of QMC, for a given number of circuits (sample size) across the whole domain of parameters which gives us more 'tail' circuits, even without the advantages of Statistical Blockade.

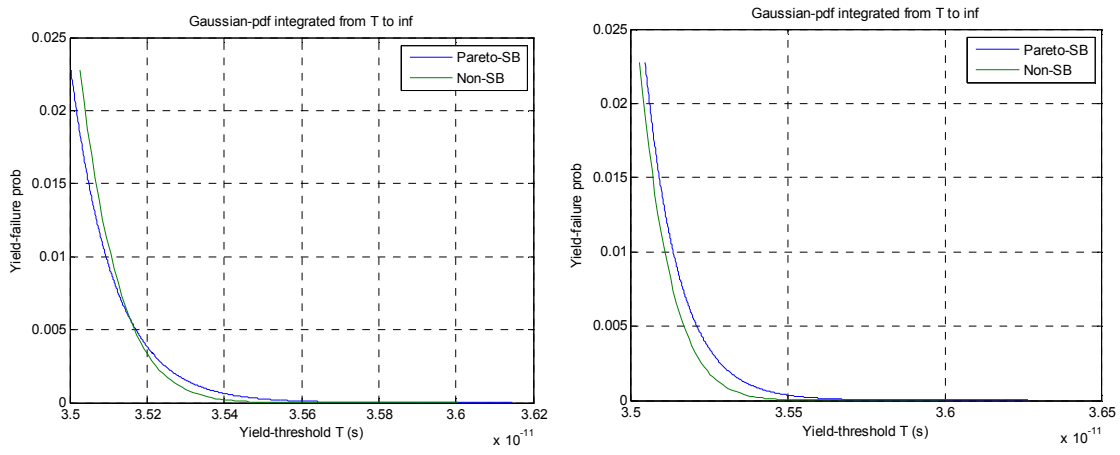


(a) (b)

Figure 7.3: Error analysis of linear estimators in RandomLA training phase, (a) MC simulation, and (b) QMC simulation.

Figure 7.4 shows the yield predictions obtained by MC and QMC analysis with and without Statistical Blockade and Pareto fitting for the 'binary full adder' circuit in figure 4.6. There are 36 transistors, and each was randomized based on two PCA components giving 72 parameters for the 'transistor-level' statistical analysis. The non-blockade analysis was carried out, with MC randomization only, on 3000 circuits which took 3497.1 seconds. The results were taken as a bench-mark for comparison with both MC-SB and QMC-SB, though a bench-mark close to this could have been obtained with about 700 fewer circuits using non-Blockade QMC. Both MC and QMC Statistical Blockade were applied using 300 training circuits in both cases. The estimator order, as always, was equal to the number of parameters, i.e. 72 in this case. The analysis time for recursive SB with MC and QMC was 146.6 s and 120.6 s respectively, achieving close to 99% savings in each case. With statistical variation from run to run, depending on how many circuits are blocked, it is not uncommon for QMC-SB to take longer than MC-SB when the same number of circuits are specified. Where the criterion is accuracy and reliability, QMC reduces the required sample size. For a given sample size, the advantages of QMC with SB over 'non-SB' are not as striking as those of MC-SB over MC without SB. More analysis is needed on this matter. As in Section 6.8, the effects of the difference in

mean and standard deviation produced by the non-SB simulations and the Training procedure can be seen in Figure 7.4. If better estimates were available, the graphs would be closer. For QMC, the maximum discrepancy in yield failure estimates between SB and non-SB is a probability difference of about 0.003 or 0.3 %. For MC the discrepancy was less, i.e. about 0.1%. Further comparisons will be presented in the next chapter.



(a) (b)

Figure 7.4: (a) MC-SB compared to MC-non-SB for BFA (3000 circuits),
(b) QMC-SB compared to MC-non-SB for BFA (3000 circuits).

7.6 Conclusions

This chapter concerns the use of Quasi Monte Carlo (QMC) techniques and 'low-discrepancy' sampling to achieve further efficiency improvements, over what was achieved in earlier chapters, with Monte Carlo circuit simulation. The effect of using a 'Sobol' low discrepancy sequence generator to replace the uniformly distributed pseudo-random number generator previously used to produce the required Gaussian parameter variation has been discussed and illustrated by example. There is a clear advantage in the convergence of the estimator training. Further analysis is needed to establish the advantages for simulation with Statistical Blockade. The interaction between the advantages of QMC and those of SB is clearly significant. The gains are not orthogonal.

Chapter 8

Results and Evaluation with SRAM Arrays

8.1 Introduction

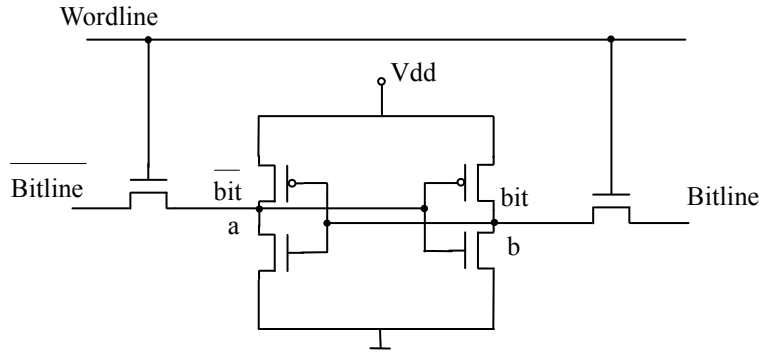
Large amounts of static random access memory (SRAM) are nowadays seen in integrated circuits. In sub-45nm MOSFET technology, the increase in fabrication variation will inevitably produce increased failure rates. Statistical testing must be performed on the designs of arrays before they are fabricated. With hundreds of millions of SRAM cells typically on a single die, the reliability of a design must be accurately estimated for output measurements up to six standard deviations from the mean [116]. The design procedure is expensive and is said to account for up to one third of the total production costs [117]. This illustrates the need for statistical analysis with the computation saving aimed for in this thesis. The use of RandomLA will be illustrated by applying the techniques it implements to statistical yield estimation for SRAM circuits, using MC and QMC analysis.

8.2 Description of the Simulations and Evaluation

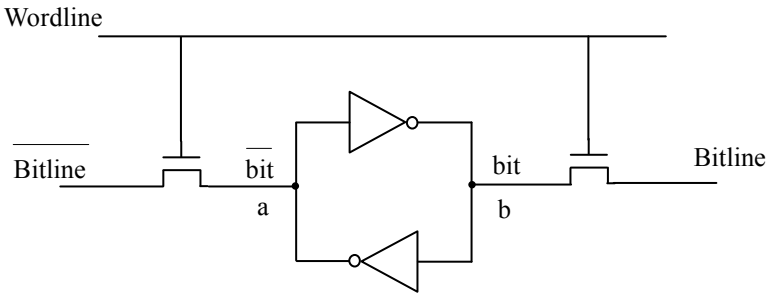
The circuits used to test the simulation facilities range from a single SRAM cell to a 32×8 array of SRAM cells. There are eight seed netlists, describing the constructions of circuits simulated. All the circuits are implemented with 35nm MOSFETs and simulations are executed on an Intel Pentium dual core processor with 1M Cache, 2.30 GHz CPU clock frequency.

8.2.1 Single SRAM Cell

MC simulation was applied to the single transistor level SRAM circuit, shown in Figure 8.1(a). A 'seed' netlist describes this circuit with six transistors; two for each inverter. The 'Bitline' was set to '1', the 'bit' node was initialised to '0' and the 'Wordline' input switched from '0' to '1' at a fixed point in time, causing the 'bit' output to switch to '1' with random delay 'write1'. The SPICE harness randomises the device parameters and makes repeated calls to SPICE to analyse the effect of these parameter variations on the 'write1' delay at the node of 'bit'. The statistics from the output of the MC simulation at pure transistor level allowed a 'SBCB' SRAM behavioural model to be constructed. After 3000 MC simulations, the resulting data set of 'write 1' delays was statistically processed to allow the required distributions, means and standard deviations to be estimated. The outputs obtained from this procedure are illustrated in Figure 8.2.



(a)



(b)

Figure 8.1: Single SRAM cell hierarchy circuits, (a) transistor level and (b) transistor - logic gate level.

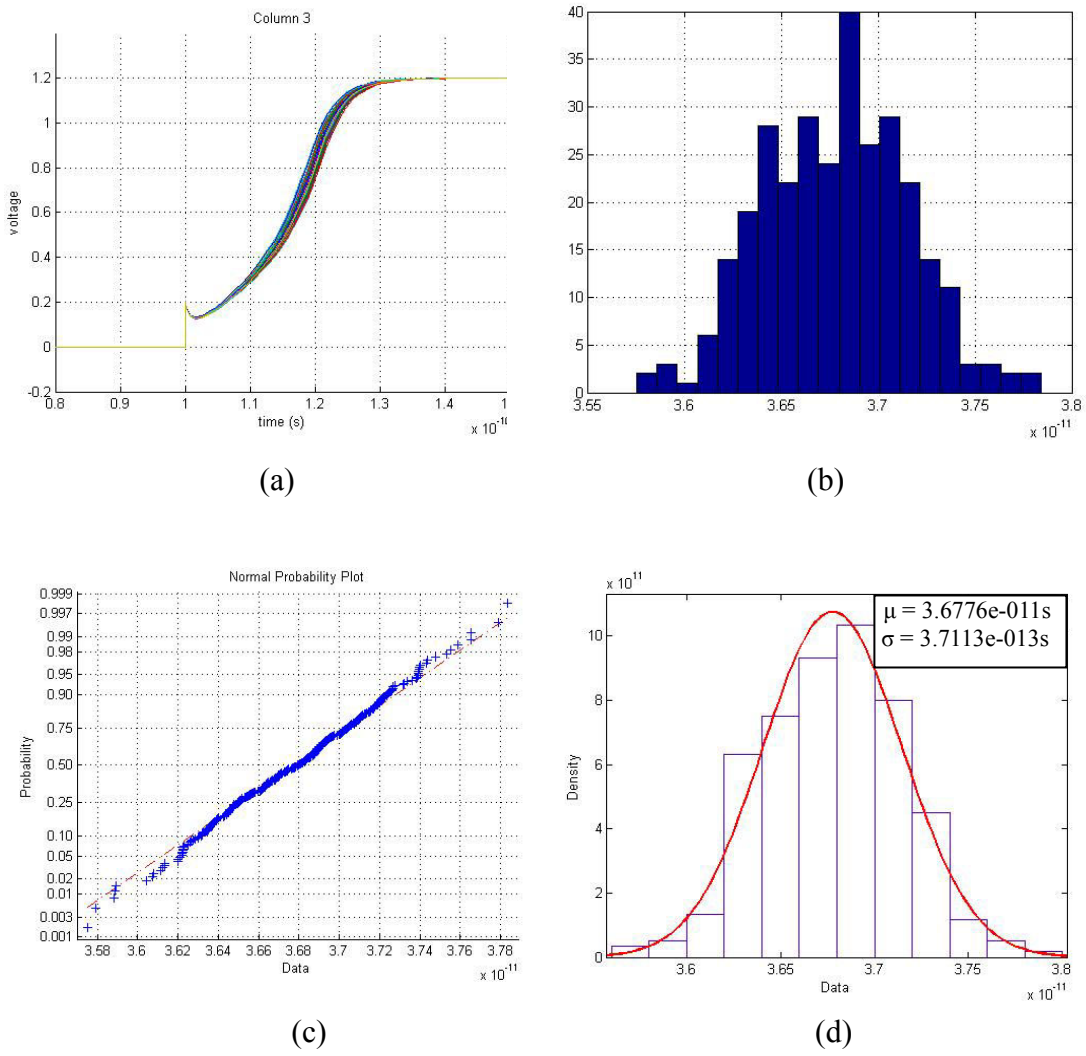


Figure 8.2: Single transistor model SRAM cell 'write 1' delay MC simulation, (a) 'write 1' signal waveforms, (b) delay time distribution histogram, (c) normal distribution evaluation, (d) Gauss fit and statistics obtained.

Figure 8.2(a) shows 'write1' delay waveforms, and plot (b) shows a delay probability distribution histogram. Figure 8.2(c) is a 'normal probability plot' to establish the degree to which it is Gaussian; the almost 'straight line' graph indicates a very strong Gaussian tendency. Figure 8.2(d) shows a Gaussian fit to the histogram. Numerical results obtained for the mean μ and standard deviation σ are indicated. In this case they are μ = 36.78e-12 s, σ = 0.371e-12 s.

The netlist, 'ngswSRAMCell.seed' in Table 1 represents the SBCB model of the single SRAM cell as constructed above . Ideally, this should be done with a tau model and switching element lookup table.

ngswSRAMCell.seed

```
****SBCB SRAM Cell netlist for NGSPICE,
***Zheng
*.GLOBAL vdd

SWmg_nmos2 bitn 0 X 0 SW OFF
SWmnmosin1 prim1 X wlprim 0 SW OFF
SWmnmosin2 bitn prim0 wlprim 0 SW OFF
SWmg_nmos1 X 0 bitn 0 SW OFF
SWmg_pmos1 X vdd vdd bitn SW OFF
SWmg_pmos2 bitn vdd vdd X SW OFF

R1 X bitp 1K
C1 bitp 0 [[3.6776e-14, 3.7113e-16]]

.IC v(bitn)=1.2 v(bitp)=0

Vdd vdd 0 1.2
Vwlprim wlprim 0 DC=0 PULSE(0 1.2 0.1n 0.01p 0.01p 0.2n 0.4n)
Vprim1 prim1 0 1.2
Vprim0 prim0 0 0

.TRAN 0.001n 0.5n
.PRINT TRAN v(bitp) v(bitn) v(wlprim)
.PROBE v(bitp) v(bitn) v(wlprim)
.OPTION POST

.MODEL SW SW
+vt = 0.8
+vh = 0.2
+ron = 0.001
+roff = 1000000

.END
```

Table 8.1: Netlist of SBCB model of single SRAM cell.

The single SRAM SBCB model now represents the transistor-level model and is suitable for 'behavioural' level statistical simulations.

8.2.2 SRAM Arrays

In this section, three SRAM arrays, 8×1, 32×1, 32×8, are taken as test-benches. Both transistor-level and SBCB-level models of the three SRAM array were statistically

simulated with the harness RandomLA in its four phases. The evaluations were as follows:

(a) 'Non-SB' MC simulation with 3000 randomised circuits to obtain failure yield probabilities without Pareto fitting: The results are presented in graphical form. Simulation run-times were recorded for comparison with the run-times for the SB simulations. MC training with 300 'uniform' randomised parameter vectors produced the estimator coefficients prior to this 'non-SB' MC simulation run.

(b) 'SB' MC simulation of 3000 randomised circuits with recursive SB and Pareto tail distribution fitting. The 'tail start' was defined as two standard deviations from the mean. The graphs and run-times obtained allowed the accuracy and computational efficiency of MC with and without SB to be compared. The same previously calculated estimator coefficients as used for the 'non-SB' MC runs were re-used. The MATLAB pseudo-random number generator was not reset for each run, therefore the 3000 randomised circuits were different for each run.

(c) 'Non-SB' QMC simulation of 3000 circuits with parameter vectors randomised by 'Sobol' vectors. The results are presented in graphical form with run-times tabulated for comparison with other simulations. QMC training with 300 'Sobol' randomised parameter vectors produced the estimator coefficients prior to this run.

(d) 'SB' QMC simulation of 3000 circuits with parameter vectors 'randomised' by Sobol vectors, and employing recursive SB and Pareto tail distribution fitting. The same previously calculated estimator coefficients as used for the 'non-SB' QMC runs were re-used. The 3000 circuits were not necessarily different for each run.

All the SB simulation runs recorded the number wrong decisions 'not to block'. The number of incorrect blocking decisions was not determined as this would have required blocked circuits to be analysed to check the blocking decision, thus invalidating the run-time comparisons.

8.2.2.1 SRAM8×1 Array

Eight copies of the circuit in Figure 8.1(b) were cascaded to construct the array shown in Figure 8.3. There are 48 transistors within the circuit. Transistor level

simulations of the array were carried out to estimate the yield failure probability for different values of yield threshold when statistical variability is included in the model for each transistor. The statistics for the parameter variation were derived from analyses of the 35nm transistor model data set provided by RandomSPICE. Two extreme cases were considered: firstly where there is assumed to be strong intra-die cell-to-cell correlation between randomised device parameters of a particular type and secondly where there is no intra-die correlation between devices from cell to cell. In each case the maximum delay or 'worst case' delay scenario applies where

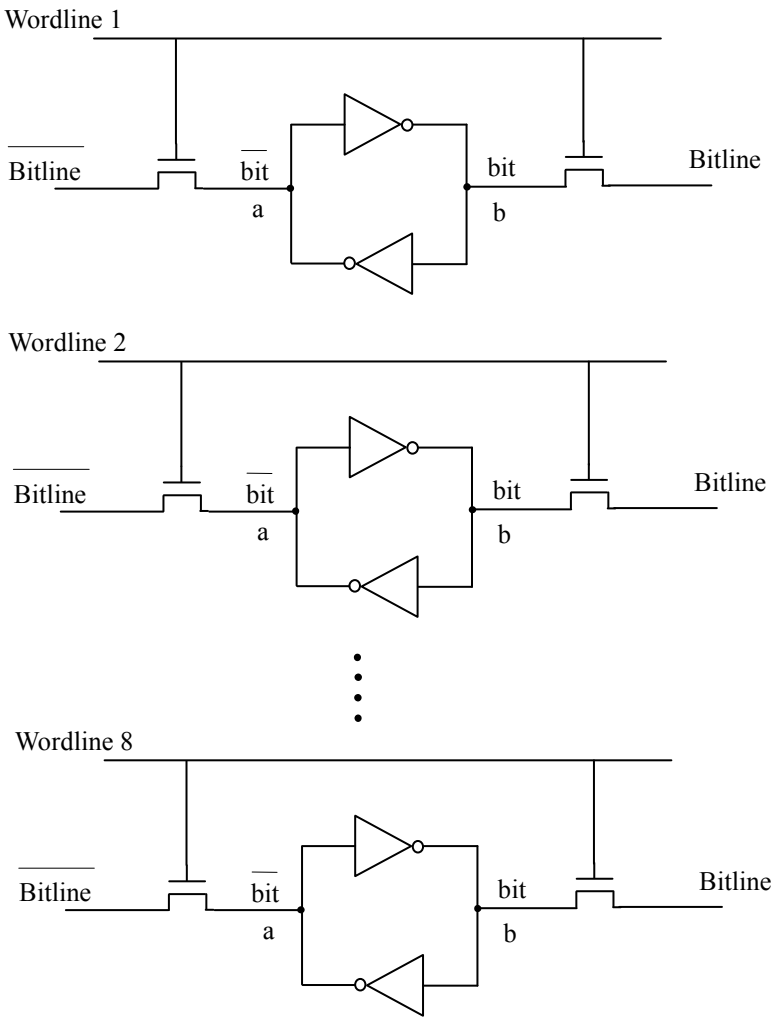


Figure 8.3: SRAM8×1 array circuit

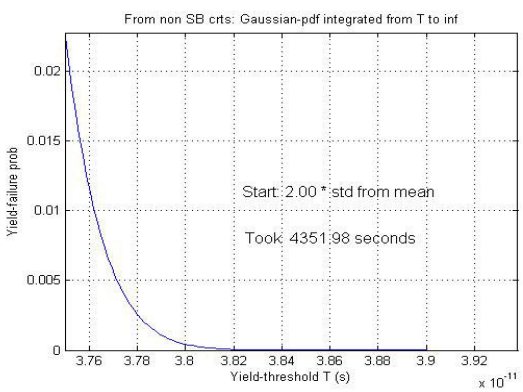
the SRAM cell with greatest delay among the array of 8 randomised cells determines the probability of yield failure. Four graphs obtained for yield failure probability against allowable delay threshold for the strong correction scenario are shown in Figure 8.4. Essentially the 'vth0' parameters of all six devices within each of the eight cells were parameterised by the same set of six random variables, generated with an appropriate value of λ, where λ is the parameter defined in Section 4.5 of this thesis. This partitions the correlation matrix into six sub-matrices (one for each device within a cell) each with $\lambda = 0$. The four graphs were obtained for (a) Traditional Monte Carlo Simulation, (b) Quasi Monte Carlo (QMC) Simulation without Statistical Blockade (SB), (c) Monte Carlo Simulation with SB and (d) QMC simulation with SB.

Figure 8.5 plots the same results that are presented in Figure 8.4, but in a way that allows the results from traditional MC analysis with and without SB, and QMC with and without SB, to be compared. In all these graphs, the timing reference was set 80 ps from the start of the run with the excitation pulse-edge occurring at 100ps from the start, which is 20 ps from the reference.

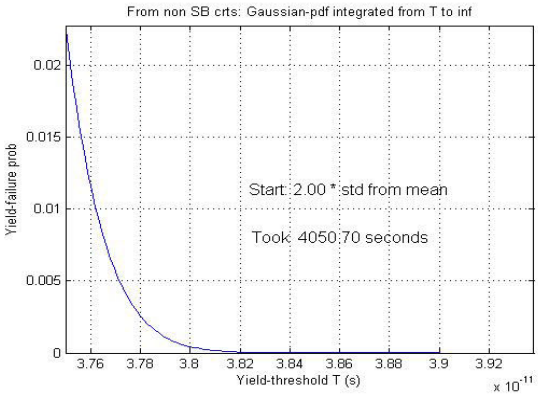
As estimated by the non-SB MC and QMC runs with 3000 randomised circuits, the mean delay 'meanNB' was found to be 36.7 ps from the reference, or 16.7 ps from the pulse-edge. The standard deviation 'sigmaNB' was found to be 0.376 ps. Therefore, the non-SB graphs, figures 8.4(a) and (b), start at the 'start of tail' which is two standard deviations from the mean, i.e. at $36.7 + 2 \times 0.376 = 37.5$ ps, from the timing reference.

As observed in Chapter 6 with reference to figure 6.10, there will be a discrepancy between the mean and standard deviation estimates 'meanTR' and 'sigmaTR', obtained from the training phase and the more accurate estimates 'meanNB' and 'sigmaNB'. This affects the accuracy of the SB results presented in figure 8.4(c) for MC and figure 8.4(d) for QMC. The differences are more clearly seen in figure 8.5(a) and (b) where the effect of QMC with SB is significantly closer to the traditional MC result than the MC with SB result. The maximum difference in predicted yield probability for MC with and without SB occurred at a yield threshold of about 37.8 ps from the timing reference (i.e. 17.8 ps from the edge). It is equal to

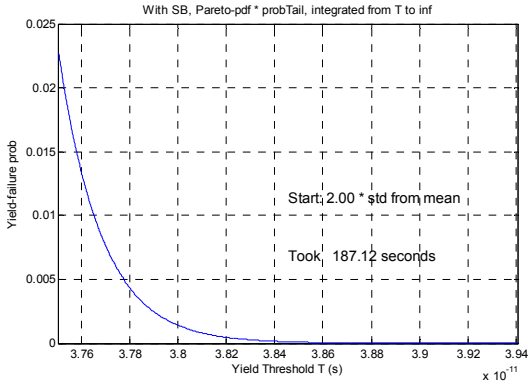
an estimated yield probability difference of about 0.0025 or 0.25 %. For QMC with and without SB, the maximum difference in estimated yield probability is less, i.e. about 0.002 or 0.2 %, this occurring at again at about 37.8 ps from the reference. The reasons for the differences, apart from the differing estimates of mean and standard deviation, may be the shape of the Pareto distribution. Improvements can be made to both these aspects of the SB programs. A summary of computation times obtained for the simulation runs represented in figures 8.4 and 8.5 for the strongly correlation case are presented in Table 8.2. The parameter 'W' is the number of circuits incorrectly blocked by the SB method. In these examples W was always zero for QMC simulations, while several non-zero values usually occurred for MC. This supports the view that QMC training appears to improve the accuracy of the estimator training.



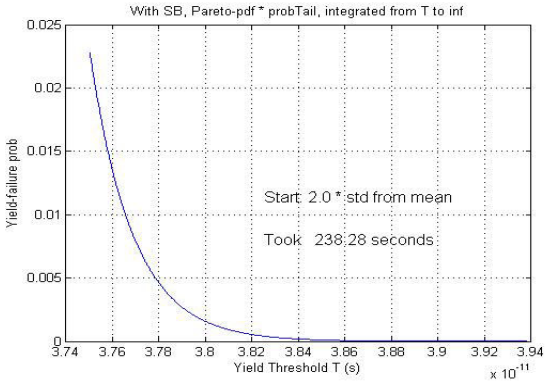
(a) MC without SB



(b) QMC without SB

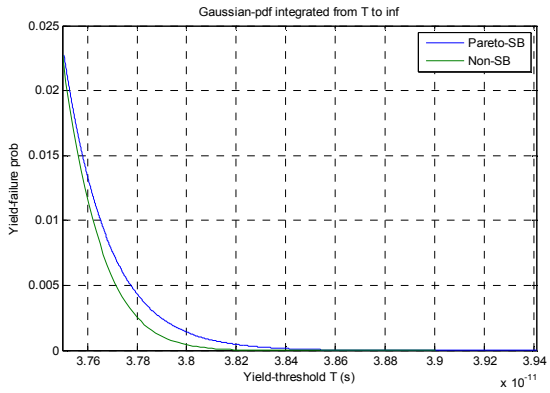


(c) MC with SB

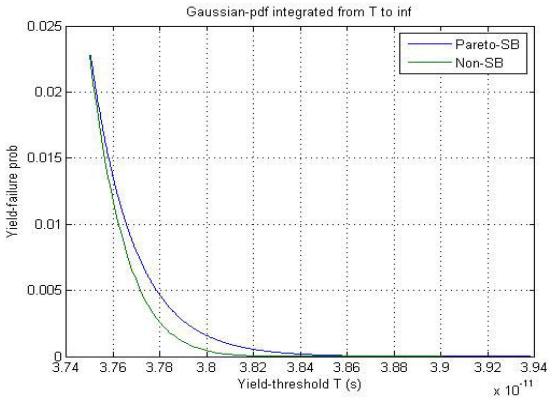


(d) QMC with SB

Figure 8.4: Yield obtained from 3000 transistor level simulations of SRAM8×1 for strongly correlated case,
(a) MC without SB, (b) QMC without SB, (c)MC with SB and (d) QMC with SB.



(a) MC

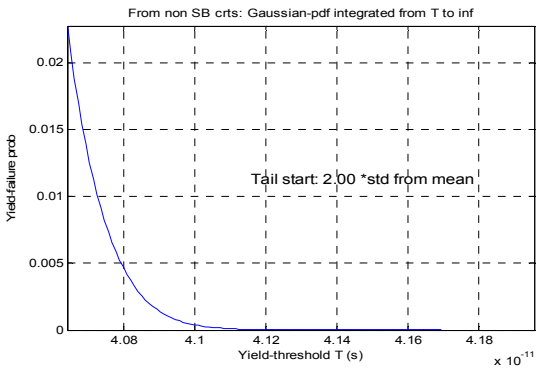


(b) QMC

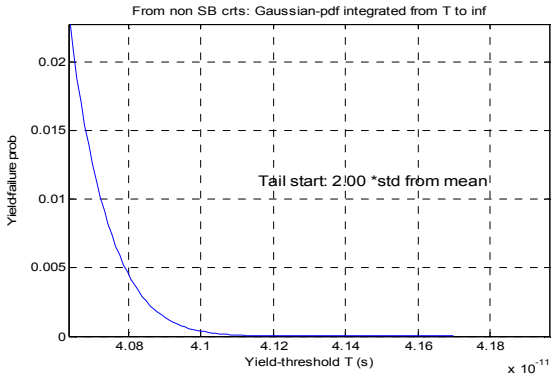
Figure 8.5: Comparison of yield analysis results of SB and non-SB for transistor level SRAM8×1 simulations for strongly correlated case, (a) MC and (b) QMC.

The results obtained from the non-correlation case are presented in Figures 8.6 and 8.7. These were obtained by connecting the eight outputs from the array, all initialised to zero, to a behaviourally modelled NAND gate. A CMOS NAND gate cannot be used here because of the possibility of introducing random switching speed variations in the 'pull down' NMOS transistor chain, depending on the order that the randomised input changes occur. If the data inputs to all eight cells are set to 1, a transition from 1 to 0 in the NAND gate output will occur only when all eight SRAM cells have correctly changed state to '1' in response to a single trigger applied simultaneously to all of them. The eight cells were randomised by independent variables without correlation.

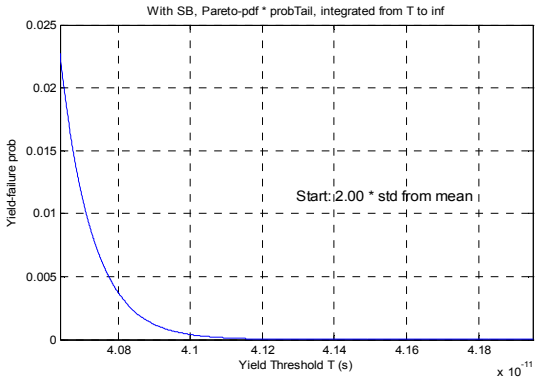
For the non-correlated case, the non-SB calculated mean was observed to be 40.1 ps from the timing reference (i.e. 20 ps from the edge) and the standard deviation was 0.263 ps. There is a reduction in standard deviation from 0.376 ps for the strongly correlated case to 0.263 ps, which is a factor of 0.69. The increase in the mean delay from 16.7 ps to 20.1 ps is explicable. With independent Gaussian parameter variations, the distribution of worst case cell delay which determines the yield failure probability is no longer Gaussian. It becomes a type of 'Gumbel' distribution [133]. It may be shown that, for eight cells, the standard deviation of the worst case delay measurements may be expected to reduce by a factor of about 0.64 in comparison with the strongly correlated case, and the mean may be expected to increase by the addition of about 1.43 times the standard deviation obtained for the strongly correlated case. Therefore we expect the mean to increase to $36.7 + 1.43 \times 0.376 = 37.24$ ps from the timing reference, and the standard deviation to decrease to about 0.376×0.64 which is 0.24 ps. The 'worst case' pdf for eight cells is close to Gaussian and is approximated as such for the non-SB and as Pareto for the SB case. Ideally a more appropriate Gumbel-type distribution should have been explored. The results obtained are close to these predictions; i.e. the standard deviation reduces by a factor of 0.69, and the mean increases by 1.436 standard deviations. Comparing the strongly correlated case (Figure 8.4) with the non-correlated case (Figure 8.6) it can be seen that the effect of intra-die correlation can have a significant effect on predictions of yield failure probability.



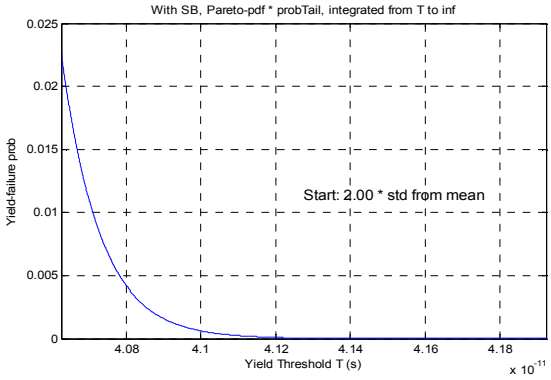
(a) MC without SB



(b) QMC without SB

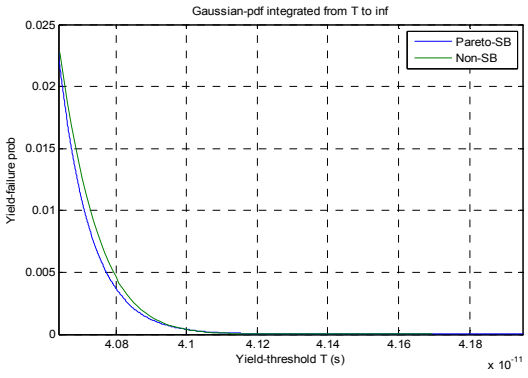


(c) MC with SB

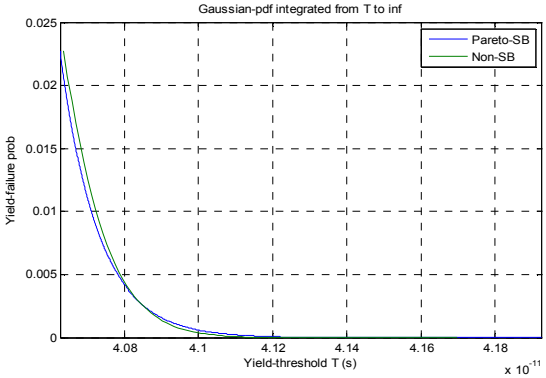


(d) QMC with SB

Figure 8.6: Yield obtained from 3000 transistor level simulations of SRAM8×1 for non-correlated case,
(a) MC without SB, (b) QMC without SB, (c) MC with SB and (d) QMC with SB.



(a) MC



(b) QMC

Figure 8.7: Comparison of yield analysis results of SB and non-SB for transistor level SRAM8×1 simulations for non-correlated case, (a) MC and (b) QMC.

Comparing the yield probability graphs, e.g. Figures 8.4(a) and 8.6(a), it may be seen that, given the same means and standard deviations for the parameter variations, the yield appears higher for the strongly correlated case than for the non-correlated case for a given permitted delay threshold. Hence, assuming no intra-die correlation where it exists could give a pessimistic estimation of yield.

Then, each of the eight cells were replaced by a behavioural model based on SBCB 'tau and delay' models of the six transistors shown in figure 8.1(a) with parameters derived from a statistical analysis of the RandomSPICE Toshiba 35nm transistor model set. The simulations carried out at transistor level, as described above, were now repeated at behavioural level. The results obtained for the strongly correlated case were very similar to those presented in Figures 8.4 and 8.5 and are not reproduced. The computation times required for the strongly correlated case are summarised in Table 8.2 and compared with the computation times required for the corresponding transistor level analyses. The overall computational time-saving due to the use of SBCB modelling with MC and QMC, for each approach with and without SB are summarised in this table. Comparing traditional 'non-SB' MC with QMC employing SB, the computational time reduced from 4351.98 to 21.63 seconds, giving an overall time saving of about 99.5%. Similar accuracy and computational savings were obtained for the non-correlated case, though with longer run-times.

8.2.2.2 SRAM32×1 Array

Thirty two copies of the circuit in Figure 8.1(b) were cascaded to construct the array shown in Figure 8.5. There are 192 transistors within the circuit. Again, transistor-level statistical simulations were carried out. Then an SBCB model of the 32×1 SRAM array was produced using the SBCB model established previously, and it was also simulated. Since the graphical results are quite similar to the ones in Section 8.2.2.1, except with different delays, only run-time results are presented and appear in Table 8.3.

	Transistor model: ngSRAM 8X1.seed				SBCB model: ngswSRAM 8X1.seed			
	MC		QMC		MC		QMC	
	NonSB	SB	NonSB	SB	NonSB	SB	NonSB	SB
CPU time/s	4351.98	187.12	4050.7	238.28	623.56	22.4	693.14	21.63
		W = 5		W = 0		W = 0		W = 0
Time saving	95.7%		94.12%		96.41%		96.88%	
Overall time saving = (4351.98-21.63)/4351.98 = 99.50%								

Table 8.2: Computational times and time savings for MC/QMC simulations of the SRAM8X1 array (strongly correlated case).

	Transistor model: ngSRAM32X1.seed				SBCB model: ngswSRAM32X1.seed			
	MC		QMC		MC		QMC	
	NonSB	SB	NonSB	SB	NonSB	SB	NonSB	SB
CPU time/s	10785.34	681.61	10144.14	615.40	740.80	27.54	823.07	31.2
		W = 0		W = 0		W = 18		W = 0
Time saving	93.68%		93.93%		96.28		96.21%	
Overall time saving = (10785.34 - 31.2)/ 10785.34 = 99.71%								

Table 8.3: Run-times and run-time savings for MC/QMC simulations of the SRAM32×1 array (strongly correlated case).

8.2.2.3 SRAM32×8 Array

Finally, thirty-two copies of figure 8.1(b) were cascaded to construct the 32×8 array shown as Figure 8.8. There are 1536 transistors within the circuit. The SBCB model of this array is established from the single cell model as in Table 8.1, and connected together as shown in Figure 8.8. The simulations were undertaken for the strong correlation case only.

Figures 8.9 and 8.10 present the graphs obtained from the simulations; the run-times are given in Table 8.4. The observed delay mean and standard deviation are very close to the ones in figure 8.4 and 8.5 for the SRAM8×1 array, which is as expected for the strongly correlated case with all cells in parallel.

Analysis of the results revealed that:

(a) SB with Pareto fitting version is reasonably accurate and much faster in comparison to the "non-Blockade" version.

(b) For the SB simulations with strong correlation, the number of wrong decisions not to block was always close to zero for QMC with Sobol points, while the number for MC fluctuated between zero and about 20 when there were 3000 circuits being analysed. This is an indication of the accuracy of the linear estimator which was found to be better for 'Sobol' vector training with under 200 training circuits, than for MC with pseudo-random parameter vectors. The results also demonstrate that QMC with Sobol vectors can make non-Blockade simulation more efficient, reaching a given accuracy with fewer runs than are required with MC.

(c) For the non-correlation examples, the estimator was much less accurate for both MC and QMC training. This caused many more wrong decisions not to block. The results of these wrong decisions are discarded for the Pareto tail fitting procedure with some loss of efficiency. The behaviour of the linear estimator when adapting to the maximum delay criterion explains this loss of efficiency.

(d) The use of QMC with 'Sobol' vectors makes non-Blockade more efficient than with MC in that a given accuracy is achieved with fewer runs.

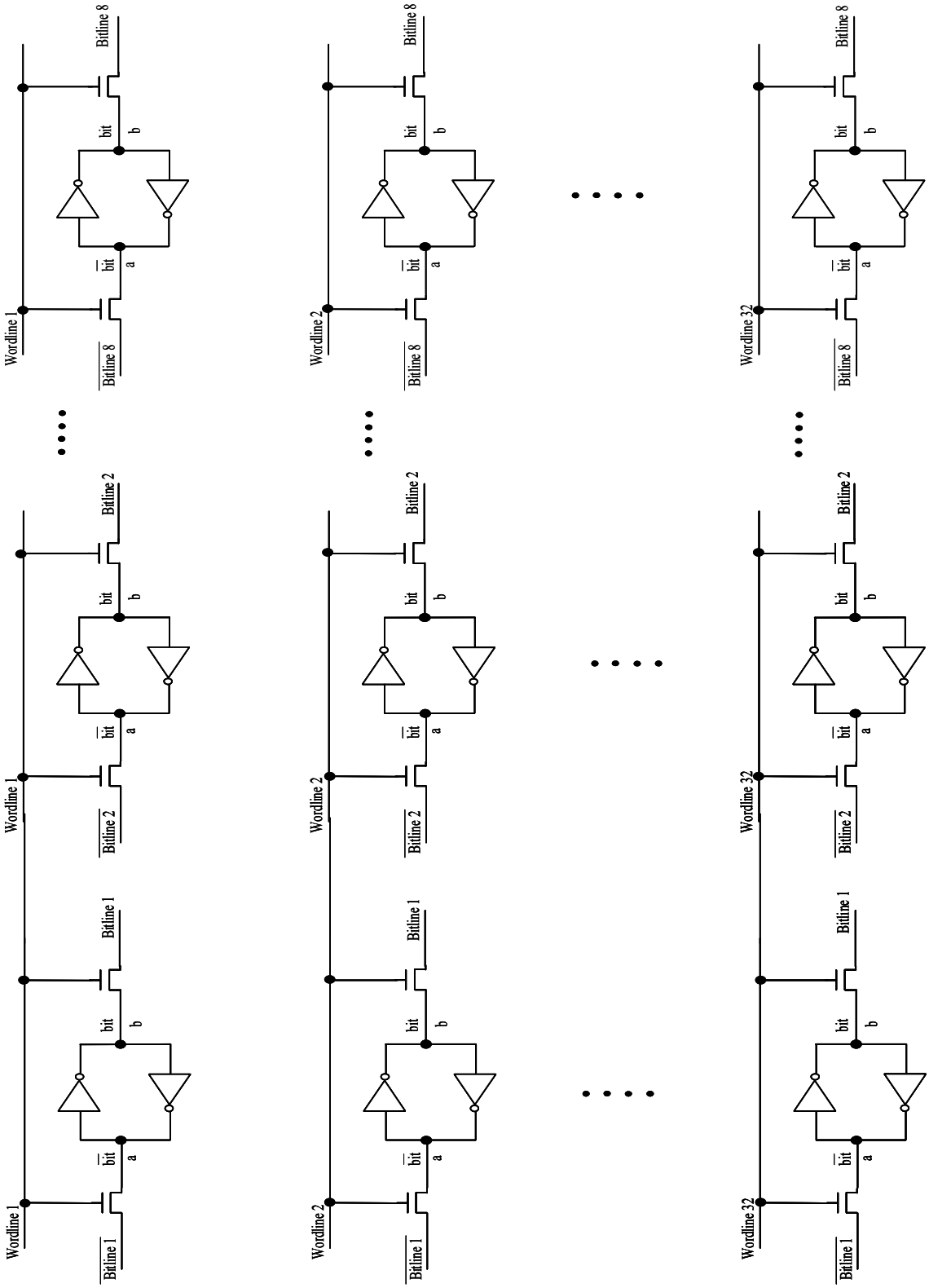
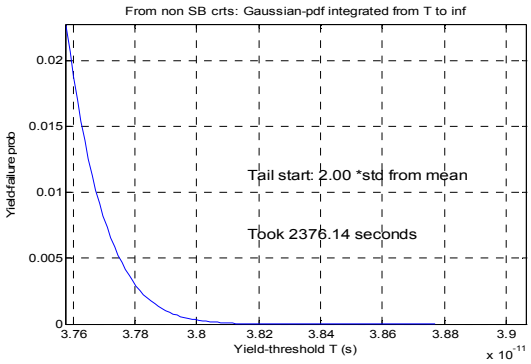
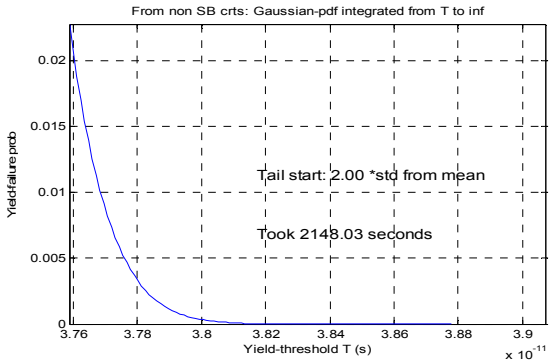


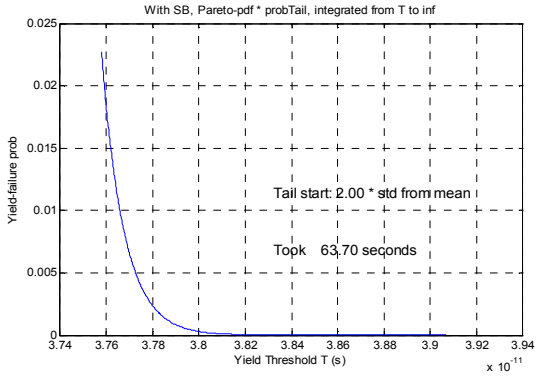
Figure 8.8: SRAM32×8 array circuit.



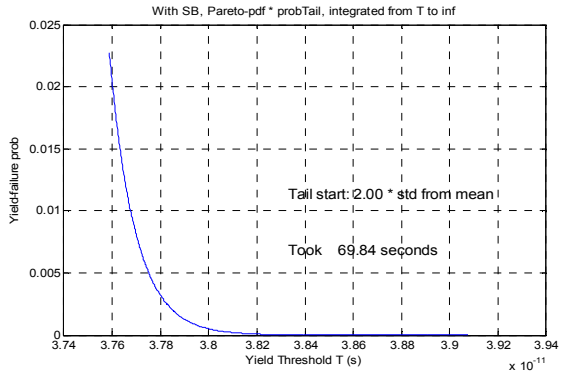
(a) MC without SB



(b) QMC without SB

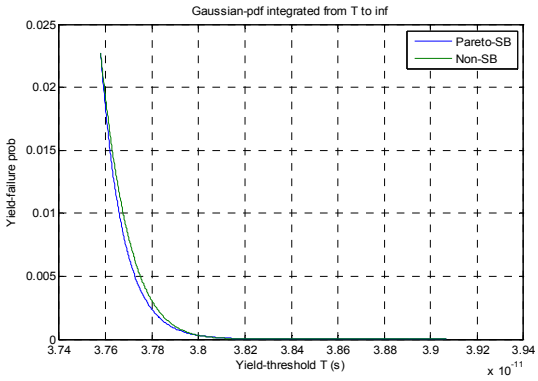


(c) MC with SB

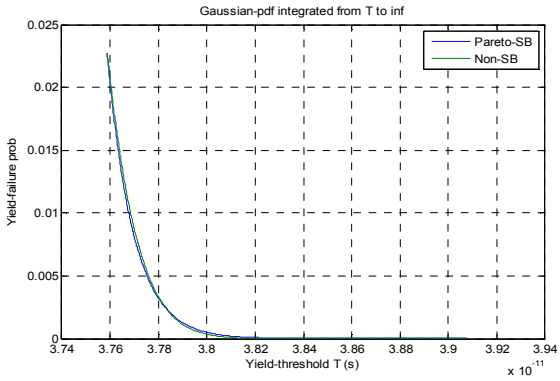


(d) QMC with SB

Figure 8.9: Yield obtained from 3000 behavioural level simulations of SRAM32×8 for strongly correlated case,
(a) MC without SB, (b) QMC without SB, (c) MC with SB and (d) QMC with SB.



(a) MC



(b) QMC

Figure 8.10: Comparison of yield analysis results of SB and non-SB for behavioural level SRAM32×8 simulations for strongly correlated case, (a) MC and (b) QMC.

	Transistor model: ngSRAM32X8.seed				SBCB model: ngswSRAM32X8.seed			
	MC		QMC		MC		QMC	
	NonSB	SB	NonSB	SB	NonSB	SB	NonSB	SB
CPU time/s	47263.96	1481.69 W = 13	39373.99	1183.59 W = 0	2376.14	63.70 W = 6	2148.03	69.84 W = 0
Time saving	96.87%		96.99%		97.32%		96.75%	
Overall time saving = (47263.96-69.84)/47263.96 = 99.85%								

Table 8.4: Run-times and time-savings for MC/QMC simulations of the SRAM32×8 array (strongly correlated case).

(e) QMC with SB compared with QMC alone offers further savings, but these remain to be fully analysed.

(f) When comparing overall time-savings for MC with SB and QMC with SB, both the training and the analysis times must be taken into account. A lower number of training circuits were required for QMC simulations than for MC to reach a given estimator accuracy.

From the Gaussian distribution shown in Figure 3.1, it may be deduced that if the delay distribution is Gaussian and the tail is assumed to start at two standard deviations from the mean, the percentage of unblocked circuits may be expected to be about 2.1 %. Therefore, out of 3000 randomly generated circuits about 63 unblocked circuits should be observed. Out of the first 500 circuits, about ten unblocked circuits should occur, and this observation suggests a simple adaptation mechanism for countering inaccuracy in the estimator. After a certain number of random circuits, say 500, have been generated, if the number of unblocked circuits is significantly different from what is expected, say 10, the tail threshold can be decreased or increased accordingly. The decision can be revisited later in the run, say after 1000 circuits, 2000 circuits and so on. This adaptation was found useful in the non-correlated examples presented in this chapter where the accuracy of the

estimator was found to be lower than for the strongly correlated examples. Decreasing the threshold does not greatly affect the computation run-time if the intention is to base the tail estimation on a specific number of circuits, say 2.1 % of the total. This approach appears even more advantageous when higher deviations from the mean are to be examined, say three or more standard deviations. Instead of specifying a fixed number of randomised circuits, the simulations could be allowed to continue until a suitable number of unblocked circuits have been produced to allow a reliable estimation of the tail distribution.

The timing results quoted in this chapter are for single core non-distributed computation. The RandomLA SPICE harness has been developed in such a way that it may be run on multi-core machines and distributed frameworks such as Condor. Using parallel or distributed computing facilities can achieve great time-savings. For example, re-running the simulations in this chapter on a dual core PC achieve a time saving which is very close to 50%, i.e. a factor of two reduction in runtime. Using Condor, the run-time of the transistor level simulations of SRAM32×8 reduced from 47263.96s (13.13 hours) to 720s (12 minutes).

8.3 Conclusions

These investigation are based on the use of RandomSPICE with an early version of the RandomLA (Random LSI analysis) statistical analysis harness which has four phases:

1. RandomLA-Nonblockade (either MC or QMC)
2. RandomLA-Training (either MC or QMC)
3. RandomLA-Evaluation (either MC or QMC)
4. RandomLA-RecursiveSB (either MC or QMC)

The results of accuracy evaluations of the linear estimator among all the experiments undertaken indicated that, as expected, the number of runs required with traditional MC training does not have to increase as dimensionality increases. As a given type of circuit, e.g. SRAM arrays with strong intra-die correlation, gets more and more complicated it seems to be still reasonable to use the same number of runs

for a certain simulation error. The SPICE computation time increases greatly with dimensionality, but the number of runs can remain approximately the same. However, the estimator's accuracy has been found to vary for different types of simulations. For example, it is lower for SRAM circuits with no intra-die correlation. A modification to the RandomLA software has therefore been made, whereby the 'start of tail' is adapted according to the number of circuits being blocked. This makes the accuracy of SB results less critically dependent on the accuracy of the estimator. Given unlimited run-times, the accuracy of the results obtained from SB need not be affected by the accuracy of the estimator, but where run-times must be minimised, estimator accuracy becomes important.

For yield estimates due to 35nm MOSFET variability, RandomLA, as developed in this thesis, has been found to provide Monte Carlo and QMC simulations for ICs containing up to 1536 transistor devices. Simulations and statistical analysis both at device level and behavioural model level give compatible results. The results indicate that assuming no intra-die correlation where it exists could give a pessimistic estimation of yield.

The results obtained from the simulations of SRAM arrays demonstrate the potential of RandomLA to achieve computation reduction for yield analysis with a delay specification. The RandomLA software is highly suitable for parallel and distributed implementations, which have already been shown to achieve great time-savings.

Chapter 9

Conclusions and Further Work

9.1 Introduction

This chapter reviews the research aims, objectives and achievements of the work presented in this thesis and draws some general conclusions. It then suggests some ideas for follow-up work.

9.2 Review of Research Aims, Objectives and Achievements

As defined in Chapter 1, the aims of this thesis were to reduce the computational complexity of traditional Monte Carlo (MC) methods for modelling the effects of variability in deep sub-micron CMOS circuits, and to enable a deeper understanding of these effects. To realize these aims, three objectives were defined, as reviewed in the following three sections along with the achievements gained with respect to each objective.

9.2.1 Design and Implementation of a Statistical Simulation Method

The first objective was the design and implementation of a statistical simulation method using traditional MC methods with facilities for including the effect of inter-die and intra-die correlation in the variability. In fulfillment of this objective, a statistical simulation method using traditional MC methods was implemented as a

MATLAB harness. This harness calls NGSPICE repeatedly to analyse a series of different circuits generated by randomly varying the parameters of a 'seed' netlist in ways that reflect the variability of fabricated circuits. The results of the analyses are combined to estimate statistical properties describing the effects of the variability. This thesis focuses on the effects on the overall delay of a circuit which must be less than some declared 'yield threshold' if the circuit is to be considered viable. The MATLAB harness has facilities for including the effects of intra-die correlation in the variability and has been shown to be suitable for distributed or parallel computation. Examples are presented to show that the traditional MC technique, as implemented in software, is capable of producing statistical estimates of yield where viability is determined by a threshold of overall delay. It has been possible to demonstrate the effect of intra-die correlation in some of these examples.

9.2.2 Dimension Reduction Techniques

The second objective was to investigate dimension reduction techniques for MC simulation, focusing on the use of Principal Components Analysis (PCA) to exploit any correlation that exists between device parameters, and the use of behavioural modelling for replacing device level analogue sub-circuits by computationally simpler circuit models. A study of the fundamental theory of Monte Carlo analysis techniques, concentrating initially on integration for simplicity, emphasized the importance of statistical independence among the 'source of randomisation' parameters. Well known results concerning convergence and error analysis take this for granted, and become invalid otherwise. PCA allows independent random vectors, used as the source of randomization, to be transformed to back to circuit parameters with the appropriate degree of correlation. It was shown how specified degrees of correlation may be introduced into component values and device parameters, with illustrations based on the 'exponential model' of proximity correlation. The same approach is applicable to principal components resulting from the PCA analysis of component and device parameters. Only intra-die correlation is considered in detail, though a similar approach may be used for inter-die correlation. The usefulness of

PCA has been outlined in this thesis with a basic example indicative of the analysis procedure. A behavioral modelling technique based on the 'tau delay and source modelling' approach has been introduced, and examples have been devised to explore its advantages. The technique has been compared to similar 'current source modelling' approaches proposed in the literature [124] [127] [129]. The use of 'look-up table' switches each with a 'Tau model' of delay [16], with the RC time-constant and the look-up table elements optimized to match the required statistically variable switching waveform has been implemented and evaluated. This approach has proved well suited to the computational methods adopted by SPICE and the demands of simulating asynchronous circuits whose behaviour relies on many 'C-elements' with highly non-linear bistable operation, switching at close but different instants of time.

9.2.3 Further Computation Reduction Methods

The third objective was to investigate two further computation reduction methods which are a technique known as 'Statistical Blockade' (SB) based on published ideas of 'extreme value theory' [15], and the use of Quasi MC techniques based on the use of 'low discrepancy sequences'[123]. The thesis investigates to what extent computation reduction can be achieved by these two methods both individually and in combination.

The Statistical Blockade (SB) algorithm applies Extreme Value Theory (EVT) to circuit analysis by eliminating randomised parameter vectors that are considered unlikely to produce 'rare event' circuits that are of interest because they are likely to fail. The process of SB requires a classifier which, in this thesis, is implemented as a 'least squares' trained linear estimator combined with a threshold comparator. After a period of initial training, the classifier is trained recursively from only the unblocked circuits as the simulation proceeds. Experiments with sample circuits confirm that the computational complexity involved in introducing the biased sampling, and compensating for it, can be significantly less expensive than performing many unnecessary circuit simulations as happens with traditional MC. It was shown that significant reduction is achievable with some cost in accuracy that has been

estimated. For a selection of SRAM arrays containing up to 1536 transistors modeled with parameters appropriate to 35nm technology, significantly faster statistical analysis has been shown to be possible when the aim is to obtain practical predictions of the yield for fabrication. Saving of up to 99.87% in computation time was obtained with these circuits. Causes of inaccuracy have been identified, for example the mean and variance estimates obtained from the training phase, the nature of the Pareto approximation to the Gaussian distribution tail and the Gaussian assumption itself. There are possible remedies to all these problems, which proved to be beyond the scope of this thesis.

It is known that the use of low discrepancy vector sequences can achieve significant speed gains over standard Monte Carlo integration techniques by reducing the number of input vectors needed for a given accuracy [15]. Our results indicate that similar gains may be obtained when QMC is used with 'low-discrepancy' 'Sobol' sequence sampling for statistical circuit simulation. The use of SB with traditional MC or quasi-MC has been shown to offer considerable promise for computation reduction. The gains of each of these approaches are not orthogonal, but there are still good reasons for using quasi-MC with SB to maximize computational savings.

As fulfilled within the scope of this thesis, the three objectives mentioned above have achieved greater understanding of the effects of variability in nano-CMOS circuits, and how they may be statistically modelled. The objectives have also led to new insight into ways of achieving reduced computational complexity, and have produced illustrations of what is achievable in the context of delay specifications.

9.2.4 Overall Conclusions

The causes and effects of variability in integrated circuits have been studied, and it is clear that anticipating the effect of variability must be a critical aspect of design procedures. In nano-scale technology, 'intra-die' variability, has become a very important consideration, and with dimensions approaching atomic scales, intrinsic atomic scale variations such as line edge roughness and dopant granularity have

become the main sources of this variation. Traditional design methodologies based on worst case corner analysis are no longer acceptable and methods based on information obtained by statistical circuit analysis techniques are now required. The effect of correlation in 'intra-die' and 'inter-die' variation which must be given due consideration

Because of the high dimensionality of the parameter space, analytical methods are ruled out. However, the use of Monte Carlo simulation is an option that has been widely explored. Monte Carlo analyses are particularly suitable to nano-scale IC statistical simulation to achieve statistical estimates of properties of interest. A major problem is the computational cost of carrying out sufficient simulations to produce statistically reliable results for all but the most trivial circuits.

Statistical variability analysis has been available in the commercial package 'HSPICE' for some time, but it is not comprehensive and HSPICE is not suited to a research project because it implements proprietary approaches without the flexibility needed for investigating new research ideas. NGSPICE is an open source mixed-signal (analogue and digital) circuit simulator that is under continuing development as part of a GNU project. The work in this thesis is intended be relevant to this project.

Although a software package called 'RandomSPICE' [14] was employed initially for the randomization process required, a new randomization package called "RandomLA" had to be developed to implement the research in this thesis. A major consideration was the need to perform the simulations and analyses with reasonable computation and to allow the use of parallel computation as provided by MATLAB and CONDOR [33] [34] [35] for circuits of realistic complexity. Hence the need for a number of complexity reduction techniques and the use of NGSPICE as explored and evaluated in the thesis.

The randomization can reflect both intra-die and inter-die variation of devices and other circuit components such as wires. Intra-die transistor parameter variation can be based on measurements and the results of 3D device modelling as carried out by our collaborators at Glasgow University [20]. Applying principal components analysis (PCA) to such sets of device parameters reduces their dimensionality and

provides a convenient way of introducing intra-die correlation. The subsequent computation reduction methods investigated in this thesis, i.e. behavioural modelling, 'Statistical Blockade' based 'extreme value theory' [15], and the use of 'low discrepancy' SOBOL sequences have been shown to have significant potential for computational complexity reduction.

9.3 Further work

The use parallel processing for efficiently undertaking the intensive computation required for statistical simulation remains to be fully explored taking into account the intrinsically parallel nature of massive Monte Carlo simulations [34]. MATLAB itself and the 'CONDOR' distributed computing facility provide all the facilities needed for this. All that is required is an implementation of NGSPICE and MATLAB on all worker machines. The fact that NGSPICE is open source and readily installed on any machine, without license, is a great advantage. MATLAB is site licensed at Manchester University, though there are restrictions on parts of its functionality.

As mentioned in Chapter 5, the use of Verilog-defined functionality is clearly a useful tool for behavioural modelling but became beyond the scope of the thesis as presented. Verilog is widely used and can define circuits at a level of abstraction appropriate for behavioural analysis, architectural design, and verification of functionality. The Balsa tool described in Chapter 3 for designing asynchronous circuits produces appropriate output. Verilog-A is supported by HSPICE and NGSPICE to allow a mixture of Verilog-A descriptions and SPICE netlists to be used to define behavioural or mixed transistor-level and behavioural simulation to be carried out. As further work, there is scope for extending the behavioural models presented in Chapter 5 to include Verilog descriptions.

The modelling of intra-die variability is achieved by a method which is applicable directly to component values and device parameters, and also indirectly to principal components which are transformed back to such values and parameters. An approach to determining representative intra-die correlation models is described by

[3] and elsewhere. In future work real foundry data will be used to improve such models.

As discovered in Chapter 5, randomizing ideal delays, even within quite modest circuits, causes SPICE to make increasingly slow progress and eventually to 'hang' (apparently). The step-size selection algorithm is to blame and this may point to a fundamental problem with the use of SPICE for MC simulation and even simulating large asynchronous circuits (maybe). Depending on how they are modeled, different switching events occurring at random throughout a large circuit may cause the step adaptation algorithm to try to model the very small timing differences and thus generate exceedingly small time-steps. Therefore, in practice, the randomization should ideally be done with reference to the anticipated time-step size. The modelling of delay by a linear time-invariant circuit, essentially a filter, with a 'look-up' table to modify the wave-shape, seems to eliminate this problem for the examples we have considered. However matching MC randomization of behavioural model parameters to the step-adaptation algorithm of SPICE is a matter deserving further investigation as there are great economies and insights to be gained. There may be a need for an asynchronous version of SPICE where iteration step-size is localised.

The use of recursion in Statistical Blockade can allow the defined 'start of tail' parameter to be gradually moved further away from the mean: typically from 2 to 3 and then to 4 or more standard deviations. Through recursion, we can thus get more accuracy in more extreme parts of the tail. This has been implemented but in a sub-optimal way. Computational savings are possible by updating rather than recalculating the estimator coefficients from scratch with the recursion. It would be useful to achieve this.

While it was clear that, in the examples considered, there were advantages in using QMC rather than MC in the estimator training, as measured by the convergence of the prediction error, further analysis is needed to establish the advantages of QMC for simulation with Statistical Blockade and Pareto 'tail fitting'. The interaction between QMC and SB has raised interesting questions, and is clearly significant. The gains are not independent, and more experiments are needed to explore the interaction.

References

[1] Hidetoshi Onodera, "Variability: Modeling and Its Impact on Design," *IEICE Trans. Electron*., Vol.E89-C, NO.3, pp.342-348, March 2006.

[2] A. Asenov, A. R. Brown, J. H. Davis, S. Kaya, and G.. Slavcheva, "Simulation of Intrinsic Parameter Fluctuations in Decananometer and Nanometer-scale MOSFETs," *IEEE Trans. Electron Devices*, vol.50, no.9, pp.1837-1852, Sept.2003.

[3] A. Srivastava, D. Sylvester, D. Blaauw, "Statistical Analysis and Optimization for VLSI: Timing and Power," *Springer*, pp.13-16 2005.

[4] A. J. Martin, P. Prakash, "Asynchronous Nano-electronics: Preliminary investigation," *Proceedings of 14th IEEE International Symposium on Asynchronous Circuits and Systems*, Newcastle Upon Tyne, UK, pp.58-68, 7-11 April, 2008.

[5] D. Edwards, A. Bardsley, L. Janin, W. Toms, "Balsa: A Tutorial Guide," School of Computer Science, University of Manchester, UK, http://apt.cs.man.ac.uk/projects/tools/balsa/, accessed Jan 2008.

[6] J. Sparsø, S. Furber, "Principles of Asynchronous Circuit Design," *Kluwer Academic Publishers*, 2005.

[7] R. Sinnott, A. Asenov, D. Berry, D. Cumming, S. Furber, C. Millar, A. Murray, S. Pickles, S. Roy, A. Tyrrell, M. Zwolinksi, "Meeting the Design Challenges of Nano-CMOS Electronics: An Introduction to an Upcoming EPSRC Pilot Project," http://www.Allhands.org.uk/2006/proceedings/papers/603.pdf.

[8] ''International Technology Road Map for Semiconductors'', http://www.itrs.net. Access date: June 2009.

[9] G. E. Moore, ''Cramming more components onto integrated circuits'' Originally published *in 'Electronics Magazine'* 1965, retrieved on 11 Nov 2006.

[10] Cadence Design Systems: http://www.cadence.com/, accessed Jan 2008.

[11] ''SPICE: Simulation program with integrated circuit emphasis'', Professional version: http://www.synopsys.com/products/mixedsignal/hspice/hspice.html.

[12] R. H. J. M. Otten and R. K. Brayton, ''Planning for performance,'' *In DAC '98:Proceedings of the 35th annual conference on Design automation*, pp. 122-127, New York, NY, USA, 1998. ACM Press.

[13] ''Meeting the challenges of nano-CMOS electronics (EPSRC pilot project)'' http://www.nanocmos.ac.uk, accessed June 2008.

[14] RandomSPICE: unpublished but available to participants of EPSRC Pilot Project: www.nanocmos.ac.uk.

[15] Amith Singhee, ''Novel Algorithms for Fast Statistical Analysis of Scaled Circuits,'' *PhD thesis*, Carnegie Mellon University, 2007.

[16] Steven M. Burns, "Performance Analysis and Optimization of Asynchronous Circuits," *Ph.D. thesis,* California Institute of Technology, Pasadena, California, 1990.

[17] William J. Morokoff , Russel E. Caflisch, "Quasi-Monte Carlo integration," *Journal of Comput. Phys*., no. 2, 218- 230, 1995,

[18] L. Kuipers, H. Niederreiter, "Uniform distribution of sequences," *Dover Publications*, pp. 129,158, ISBN 0-486-45019-8, 2005.

[19] S. Borkar, ''Design reliable systems from unreliable components: the challenges of transistor variability and degredation,'' *IEEE Micro*, vol.25, no. 6, pp. 10-16, Dec 2005.

[20] G. Roy, A. R. Brown, F. Adamu-lema, S. Roy, and A. Asenov, ''Simulation study of individual and combined sources of intrinsic parameter fluctuations in conventional nano-MOSFETs,'' *IEEE Trans. Electron Devices*, vol. 53, no. 12, pp. 3036-3037, Dec 2006.

[21] H. Niederreiter, "Quasi-Monte Carlo methods and pseudo-random numbers," *Bull. Amer. Math. Soc*., 1978.

[22] Harald Niederreiter, "Random Number Generation and Quasi-Monte Carlo Methods," *Society for Industrial and Applied Mathematics*, ISBN 0-89871-295-5, 1992.

[23] M. Hane, T. Ikezawa, and T. Ezaki, "Atomistic 3d process/device simulation considering gate line-edge roughness and poly-si random crystal orientation effects," In *Proc. IEEE Int. Electron Devices Meeting*, 2003.

[24] P. Glasserman, "Monte Carlo Method in Financial Engineering," *Springer*, 2004

[25] C. P. Robert, G. Casella, "Monte Carlo Statistical Methods (2nd ed.)," *Springer*, New York, 2004.

[26] S. I. Resnick, "Extreme Values, Regular Variation and Point Processes," *Springer*, New York, 1987.

[27] E. Castillo, "Extreme value theory in engineering," *Academic Press*, Inc., New York, 1988.

[28] A. Singhee and R. A. Rutenbar, "Statistical Blockade: a novel method for very fast Monte Carlo simulation of rare circuit events, and its application," In *Proc. Design Autom. Test Europe*, 2007.

[29] A. Singhee, J. Wang, B. H. Calhoun, and R. A. Rutenbar. “Recursive statistical blockade: an enhanced technique for rare event simulation with application to SRAM circuit design,” In *Proc. Int. Conf. VLSI Design*, 2008.

[30] T. Hastie, R. Tibshirani, and J. Friedman. “The Element of Statistical Learning: Data Mining, Inference, and Prediction,” *Springer*, 2001.

[31] HSPICE User Guide: Simulation and Analysis, Version A-2007.12, December 2007, Synopsys.

[32] C. Millar, D. Reid, G. Roy, S. Roy, and A. Asenov, “Accurate Statistical Description of Random Dopant-Induced Threshold Voltage Variability,” *IEEE Electron Device Letters*, vol. 29, no. 8, August 2008.

[33] J. Beck, "Roth's estimate of the discrepancy of integer sequences is nearly sharp," *Combinatorica*, pp. 319-325, 1981

[34] L. Han, A. Asenov, D. Berry, C. Millar, G. Roy, S. Roy, R. Sinnott, G. Stewart, “Towards a Grid-Enabled Simulation Framework for Nano-CMOS Electronics,” *Proceedings of the Third IEEE internatioinal Conference on e-Science and Grid Computing*, pp. 305-311, 2007.

[35] R. Sinnott, A. Asenov, A. Brown, C. Millar, G. Roy, S. Roy, G. Stewart. “Grid Infrastructures for Electronics Domain: Requirements and Early Prototypes from and EPSRC Pilot Project,” *Proceedings of the UK e-Science All Hands Conference*, National e-Science Centre, 2007. website: http://www.allhands.org.uk/2007/, ISBN/ISSN: 978-0-9553988-3-4.

[36] R. E. Caflisch. “Monte Carlo and quasi-Monte Carlo methods,” *Acta Numerica*, vol. 7, pp.1-49, Cambridge University Press, 1998.

[37] ‘NGSPICE release 23’ website: www.sourceforge.net, 5th June 2011.

[38] L. Kuipers, H. Niederreiter, “Uniform distribution of sequences,” *Dover Publications*, ISBN 0-468-45019-8, 2005.

[39] MATLAB Parallel Computation Toolbox, MathWorks United Kingdom, http://www. mathworks.co.uk /products/parallel-computing/.

[40] Doulas Thain, Todd Tannenbaun, and Miron Livny, "Distributed Computing in Practice: The Condor Experience," *Concurrency. Prac.*, John Wiley & Sons, Ltd, 2004.

[41] B. Bindu, B. Cheng, G. Roy, X. Wang, S. Roy, A. Asenov, "Parameter set and data sampling strategy for accurate yet efficient statistical MOSFET compact model extraction," *Solid-State Electronics,* Volume 54, Issue 3, pp. 307-315, March 2010.

[42] Sheu, Scharfetter, Ko, and Jeng, "BSIM: Berkeley Short-Channel IGFET Model for MOS Transistors," *IEEE Journal of Solid State Circuits*, vol. SC-22, pp. 558-566, August 1987.

[43] "CMC-Compact Model Council", website: http://www.geia.org, accessed September 2011.

[44] S. M. Hisayo, N. Shin-ichi, O. Tatsuya, Y. Takashi, M. Eiji, M. Toyota, K. Yasuhiro and Hiroshi I., "Study of the Manufacturing Feasibility of 1.5-nm Direct-Tunneling Gate Oxide MOSFET's: Uniformity, Reliability, and Dopant Penetration of the Gate Oxide," *IEEE Transactions on Electron Devices*, vol. 45, no. 3, pp. 691-700, March 1998

[45] David A. Gates, "Design-Oriented Mixed-Level Circuit and Device simulation," *Technical Report*, no. UCB/ERL M93/51, University of California, Berkeley, 1993.

[46] J. P. Fishburn and A. E. Dunlop, "TILOS: A Polynomial Programming Approach to Transistor Sizing," *IEEE ICCAD*, pp. 326-328, November 1985.

[47] H. C. deGraaf, F. M. Klaassen, "Compact Transistor Modelling for Circuit Design," *Springer-Verlag/Wein*, New York, pp. 351, ISBN 0-387-82136-8, 1990.

[48] J. R. Pierret, "A MOS Parameter Extraction Program for the BSIM Model," *U. C. Berkeley Electronics Res. Lab.*, Memo No. UCB/ERL M84/99&100, November 1984.

[49] C. N. Berglund, "A unified yield model incorporating both defect and parametric effects," *IEEE Trans. Semiconductor Manufacturing*, vol. 9, no. 3, pp. 447- 454, August 1996.

[50] A. Lochtefeld and D. A. Antoniadis, "On experimental determination of carrier velocity in deeply scaled NMOS: How close to the thermal limit?," *IEEE Electron Device Letter.*, vol. 22, pp. 95-97, February 2001.

[51] J. Welser, J. L. Hoyt, and J. F. Gibbons, "Electron mobility enhancement in strained-Si n-type metal-oxide-semiconductor field-effect transistors," *IEEE Electron Device Letter*, vol. 15, pp. 100-102, March 1994.

[52] M. Saitoh, N. Yasutake, Y. Nakabayashi, K. Uchida, and T. Numata, "Understanding of strain effects on high-field carrier velocity in (100) and (110) CMOSFETs under quasi-ballistic transport," *IEDM Tech. Dig.*, pp. 469-472, 2009.

[53] T. Ashley, M. T. Emeny, D. G. Hayes, K. P. Hilton, R. Jefferies, J. O. Maclean, S. J. Smith, A. W-H. Tang, D. J. Wallis, and P. J. Webber, "High-performance InSb based quantum well field effect transistors for low-power dissipation applications," *IEDM Tech.Dig.*, pp. 849-852, 2009.

[54] T. Ghani, K. Mistry, P. Packan, S. Thompson, M. Stettler, S. Tyagi, and M. Bohr, "Scaling challenges and device design requirements for high performance sub-50nm gate length planar CMOS transistors," *Symp. VLSI Tech. Dig.*, pp. 174-175, 2000.

[55] G. D. Wilk, R. M. Wallace, and J. M. Anthony, "High- K gate dielectrics: Current status and materials properties considerations," *Journal. Application Physics.*, vol. 89, pp. 5243-5275, May 2001.

[56] J. Kedzierski, D. M. Fried, E. J. Nowak, T. Kanarsky, J. Rankin, H. Hanafi, W. Natzle, D. Boyd, Y. Zhang, R. Roy, J. Newbury, C. Yu, Q. Yang, P. Aunders, C. Willets, A. Johnson, S. Cole, H. Young, N. Carpenter, D. Rakowski, B. A. Rainey, P. Cottrell, M. leong, and P. Wong, "High-

performance symmetric-gate and CMOScompatible VT asymmetric-gate FinFET devices," *IEDM Tech. Dig.*, pp. 437- 440, 2001.

[57] S. Saxena, C. Hess, H. Karbasi, A. Rossoni, S. Tonello, P. McNamara, S. Lucherini, S. Minehane, C. Dolainsky, and M. Quarantelli, "Variation in transistor performance and leakage in nanometer-scale technologies," *IEEE Trans. Electron Devices*, vol. 55, pp. 131-144, January 2008.

[58] C. H. van Berkel, M. B. Josephs, and S. M. Nowick, "Scanning the technology: Applications of asynchronous circuits," *Proceedings of the IEEE*, vol. 87(2), pp. 223-233, February 1999.

[59] J. B. Dennis, "Data Flow Computation," *Proc. of the NATO Advanced Study Institute on Control flow and data flow: concepts of distributed programming*, Springer-Verlag, New York, 1986.

[60] C. H. van Berkel, R. Burgess, J. Kessels, A. Peeters, M. Roncken, and F. Schalij, "Asynchronous circuits for low power: a DCC error corrector," *IEEE Design & Test*, vol. 11(2), pp. 22-32, 1994.

[61] T. E. Williams and M. A. Horowitz, "A zero-overhead self-timed 160 ns. 54 bit CMOS divider," *IEEE Journal of Solid State Circuits*, vol. 26(11), pp. 1651-1661, 1991.

[62] N. C. Paver, P. Day, C. Farnsworth, D. L. Jackson, W. A. Lien, and J. Liu, "A low-power, low-noise configurable self-timed DSP," In *Proc. International Symposium on Advanced Research in Asynchronous Circuits and Systems*, pp. 32- 42, 1998.

[63] L. S. Nielsen, C. Niessen, J. Sparsø, and C. H. van Berkel, "Low-power operation using self-timed circuits and adaptive scaling of the supply voltage," *IEEE Transactions on VLSI Systems*, vol. 2(4), pp. 391- 397, 1994.

[64] A. R. Brown, A. Asenov and J. R. Watling, "Intrinsic Fluctuations in Sub 10 nm Double-Gate MOSFETs Introduced by Discreteness of Charge and Matter," *IEEE Trans. on Nanotechnology*, Vol.1, pp. 195-200, 2002.

[65] G. Roy, F. Adamu-Lema, A. R. Brown, S. Roy and A. Asenov, "Simulation of Combined Sources of Intrinsic Parameter Fluctuations in a 'Real' 35 nm MOSFET," *Proc. European Solid-State Device Research Conference (ESSDERC)*, 12-16 September, Grenoble, France, pp. 337-340, 2005

[66] National Microelectronics Institute. *International Conference CMOS Variability: "The impact of Variability on Design*," Royal College of Physicians, London, October 2007.

[67] Asen Asenov, "Variability in the next generation CMOS technology and impact on design," *International Conference of CMOS Variability*, Key Note, 2007.

[68] David T. Reid, "Large-scale simulations of intrinsic parameter fluctuations in nano-scale MOSFETs," *PhD thesis*, University of Glasgow, 2010.

[69] P. Oldiges, Q. Lin, K. Pertillo, M. Sanchez, M. Ieong, and M. Hargrove, "Modeling line edge roughness effects in sub 100 nm gate length devices," *Proc. SISPAD*, pp. 131–134, 2000.

[70] S. Kaya, A. R. Brown, A. Asenov, D. Magot, and T. Linton, "Analysis of statistical fluctuations due to line edge roughness in sub 0.1um MOSFET's," In s*imulation of Semiconductor Processes and Devices 2001*, Springer-Verlag, Vienna, Austria, pp. 78–81, 2001.

[71] S. Winkelmeier, M. Sarstedt, M. Ereken, M. Goethals, and K. Ronse, "Metrology method for the correlation of line edge roughness for different resists before and after etch," *Microelec. Eng.*, vol. 665, pp. 57-58, 2001.

[72] G. F. Cardinale, C. C. Henderson, J. E. M. Goldsmith, P. J. S. Mangat, J. Cobb, and S. D. Hector, "Demonstration of pattern transfer into sub-100 nm polysilicon line/space features patterned with extreme ultraviolet lithography," *J. Vac. Sci. Tech. B*, vol. 17, pp. 2970–2974, 1999.

[73] A. Asenov, S. Kaya, A. R. Brown, "Intrinsic Parameter Fluctuations in Decananometer MOSFETs Introduced by Gate Line Edge Roughness," *IEEE Transactions on Electron Devices*, vol. 50, no. 5, May 2003.

[74] S. Xiong, J. Bokor, Q. Xiang, P. Fisher, I. Dudley, P. Rao, "Study of Gate Line Edge Roughness Effects in 50 nm Bulk MOSFET Devices," *Proceedings of SPIE*, vol. 4689, 2002.

[75] G. Roy, A. R. Brown, F. Adamu-Lema, S. Roy, A. Asenov, "Simulation Study of Individual and Combined Sources of Intrinsic Parameter Fluctuations in Conventional Nano-MOSFETs," *IEEE Transactions on Electron Devices*, vol. 53, no. 12, December 2006.

[76] A. Asenov, S. Kaya, J. H. Davies, "Intrinsic Threshold Voltage Fluctuations in Decanano MOSFETs Due to Local Oxide Thickness Variations," *IEEE Transactions on Electron Devices*, vol. 49, no. 1, pp. 112-119, January 2002.

[77] M. Niva, T. Kouzaki, K. Okada, M. Udagawa, and R. Sinclair, "Atomicorder planarization of ultrathin SiO /Si(001) interface," *Jpn. J. Appl.Phys*, vol. 33, pp. 388–394, 1994.

[78] D. Z. Y. Ting, E. S. Daniel, and T. C. McGill, "Interface roughness effects in ultrathin gate oxides," *VLSI Syst. Des.*, vol. 8, pp. 47-51, 1998.

[79] E. Cassan, P. Dollfus, S. Galdin, and P. Hesto, "Calculation of direct tunnelling gate current through ultrathin oxide and oxide/nitride stacks in MOSFETs and H-MOSFETs," *Microelectron. Reliab.*, vol. 40, pp. 585–588, 2000.

[80] A. Asenov and S. Kaya, "Effect of oxide roughness on the threshold voltage fluctuations in decanano MOSFETs with ultrathin gate oxide," *Proceedings of SISPAD*, pp. 135-138, 2000.

[81] M. Koh, W. Mizubayashi, K. Ivamoto, H. Murakami, T. Ono, M. Tsuno, T. Mihara, K. Shibahara, S. Miyazaki, and M. Hirose, "Limit of gate oxide thickness scalling in MOSFETs due to apparent threshold voltage fluctuation introduced by tunnelling leakage current," *IEEE Trans. Electron Devices*, vol. 48, pp. 259-264, Jan. 2001.

[82] L. W. Nagel and P. O. Pederson, "SPICE (Simulation Program with Integrated Circuit Emphasis)," *Memorandum* No. ERL-M382, University of California, Berkeley, Apr. 1973.

[83] A. J. Bhavnagarwala, X. Tang, and J. D. Meindl, "The impact of intrinsic device fluctuations on CMOS SRAM cell stability," *IEEE Trans. Solid-State Circuits*, vol. 36(4), pp. 658-665, 2001.

[84] B. Hargreaves, H. Hult, S. Reda, "Within-die process variations : How accurately can they be statistically modeled?," *Proc. IEEE*, ASPDAC 2008, pp. 524-530.

[85] R. K. Brayton, G. D. Hachtel, A. L. Sangiovanni-Vincentelli, "A survey of optimization techniques for integrated circuit design," *Proc. IEEE*, vol. 69, pp. 1334-1362, 1981.

[86] S. R. Nassif, A. J. Strojwas and S. W. Director, "A methodology for worst-case analysis of integrated circuits," *IEEE Trans. Computer Aided Design*, vol. CAD-5, no. 1, pp. 104-113, January 1986.

[87] P. Tuohy, A. Gribben, A. J. Walton and J. M. Robertson, "Realistic worst-case parameters for circuit simulation," *IEEE Proceedings*, vol.134, no.5, pp. 137-140, October 1987.

[88] G. E. Muller-L., "Limit-parameters: the general solution of the worst-case problem for the circuit simulation," *Proc. IEEE Int. Symp. Circuits and Systems*, pp. 2256-2259, 1990.

[89] M. Orshansky, K. Keutzer, "A general probabilistic framework for worst case timing analysis," In *Proceedings of 39th Design Automation Conference*, pp. 556-561, 2002.

[90] C. Visweswariah, K. Ravindran, K. Kalafala, S. G. Walker and S. Narayan, "First-order incremental block-based statistical timing analysis," *Proc. IEEE/ACM Design Auto. Conf.*, pp. 2170-2180, June 2004.

[91] H. Chang and S. Sapatnekar, "Statistical timing under spatial correlations," *IEEE Trans. Computer-Aided Design*, vol. 24(9), pp. 1467-1482, 2005.

[92] N. Metropolis and S. M. Ulam, "The Monte Carlo method," *Journal of the American Statistical Association*, vol. 44, pp. 335-341, 1949.

[93] P. L'Ecuyer, "Good parameters and implementations for combined multiple recursive random number generators," *Operations Research*, Vol. 47(1), pp. 159-164, 1999.

[94] M. Matsumoto and T. Nishimura, "Mersenne Twister: A 623-dimensionally equidistributed uniform pseudo-random number generator," *ACM Transactions on Modeling and Computer Simulation*, vol. 8(1), pp. 3-30, 1998.

[95] Christiane Lemieux, "Monte Carlo and Quasi-Monte Carlo Sampling," *Springer Series in Statistics*, New York, 2009.

[96] R. Bellman, "Adaptive Control Processes: A Guided Tour," *Princeton University Press*, Princeton, NJ, 1961.

[97] S. K. Chaudhary, "Acceleration of Monte Carlo methods using low discrepancy sequences," *PhD thesis*, UCLA, 2004.

[98] D. I. Asotsky, E. E. Myshetskaya and I. M. Sobol', "The average dimension of a multidimensional function for quasi-Monte Carlo estimates of an integral," *ComputationalMathematics and Mathematical Physics*, vol. 46, pp. 2061-2067, 2006.

[99] H. Niederreiter, "Random Number Generation and Quasi-Monte Carlo Methods," *Regional Conference Series in Applied Mathematics*, vol. 63 of SIAM CBMS-NSF, SIAM, Philadelphia, 1992.

[100] A. B. Owen, "Scrambled net variance for integrals of smooth functions," *Annals of Statistics*, vol. 25(4), pp. 1541-1562, 1997.

[101] W. H. Press, G. R. Farrar, "Recursive Stratified Sampling for Multidimensional Monte Carlo Integration," *Computers in Physics*, vol. 4, pp. 190-195, 1990.

[102] G. P. Lepage, "A new algorithm for adaptive multi-dimensional integration," *Journal of Computational Physics*, vol. 27, issue 2, pp. 192-203, 1978.

[103] G. P. Lepage, "VEGAS: An adaptive multi-dimensional integration program," *Cornell preprint*, CLNS 80-447, March 1980.

[104] David P. Marple and Abbas El Gamal, "Optimal Selection of Transistor Sizes in Digital VLSI," *Advanced Research in VLSI, Proceedings of the 1987 Stanford Conference*, pp. 151-172, MIT Press, Cambridge, MA, 1987

[105] W. H. Press, G. R. Farrar, "Recursive Stratified Sampling for Multidimensional Monte Carlo Integration," *Computers in Physics*, vol. 4, pp. 190-195, 1990.

[106] High Throughput Computing using Condor at Manchester, http://condor.eps.manchester.ac.uk/, accessed October 2011.

[107] Zheng Xie, Doug Edwards, "Computation Reduction for Statistical Analysis of the Effect of Nano-CMOS Variability on Asynchronous Circuit," *Proceedings of the 13th IEEE International Symposium on Design and Diagnostics of Electronic Circuits and Systems (DDECS)*, pp. 161-166, April 14-16, 2010, Vienna, Austria. ISBN 978-1-4244-6610-8.

[108] G. Lindgren and H. Tootzen, "Extreme value: Theory and Technical Applications," *Scandinavian Journal of Statistics*, vol.14, pp. 241-279, 1987.

[109] Amith Singhee, Jiajing Wang, Benton H. Calhoun, Rob A. Rutenbar. "Recursive Statistical Blockade: An Enhanced Technique for Rare Event Simulation with Application to SRAM Circuit Design," *Proceedings of 21st International Conference on VLSI Design (VLSID)*, pp. 131-136, 4-8 January 2008, Hyderabad.

[110] R. Srinivasan, "Importance sampling - Applications in communications and detection," *Springer-Verlag*, Berlin, 2002.

[111] R. Kanj, R. Joshi, S. Nassif, "Mixture importance sampling and its application to the analysis of SRAM designs in the presence of rare failure events," *proceedings of 43rd ACMIEEE Design Automation Conference*, pp.

69-72, 2006.

[112] D. E. Hocevar, M. R. Lightner, T. N. Trick, "A Study of Variance Reduction Techniques for Estimating Circuit Yields," *IEEE Trans. CAD*, vol. 2(3), July, 1983.

[113] T. C. Hesterberg, "Advances in Importance Sampling," *PhD Dissertation*, Dept. of Statistics, Stanford University, 2003.

[114] W. H. Teukolsky, S. A. Vetterling, W. T. Flannery, "Numerical Recipes: The Art of Scientific Computing (3rd ed.) - Section 16.5. Support Vector Machines," *Cambridge University Press*, New York, ISBN: 978-0-521-880608-8, 2007.

[115] J. Spanier, "Quasi-Monte Carlo methods for particle transport problems," In H. Niederreiter and P. J.-S. Shiue, editors, Springer-Verlag. *Monte Carlo and Quasi-Monte Carlo Methods in Scientific Computing*, pp. 121-148, New York, 1995.

[116] G. Zheng, C. Andrew, P. Liang-Teck, D. Kenneth, L. Tsu-Jae King, N. Borivoje, "Large-Scale Read/Write Margin Measurement in 45nm CMOS SRAM Arrays," *IEEE Symposium on VLSI Circuits Digest of Technical Papers*, pp. 42- 43, 2008.

[117] Lawrence T. Clark, Yu Cao, "Fast Parallel Test of SRAM Arrays," *Patent,* Arizona Technology Enterprises, LLC (AzTE), http://azte.technologypublisher.com, accessed January 2012.

[118] Fred W. Obermeier, "An Open Architecture for Improving VLSI Circuit Performance," *Ph.D. Thesis*, University of California, Berkekey, 1989.

[119] L. Kuipers, H. Niederreiter, "Uniform distribution of sequences," *Dover Publications*, pp. 129-158, ISBN 0-486-45019-8, 2005.

[120] J. Halton, "Algorithm 247: Radical-inverse quasi-random point sequence," *Communications of the ACM*, vol. 7, pp. 701-702, 1964.

[121] I. M. Sobol', "The distribution of points in a cube and the approximate evaluation of integrals (English translation)," *Math. and Math. Phys.,*U.S.S.R. Comp., issue 7, vol.4, pp. 86-112, 1967.

[122] M. Boyle, N. Bonini, S. DiNardo, "Expression and function of clift in the development of somatic gonadal precursors within the Drosophila mesoderm," *Journal of Development*, issue 124, vol.5, pp. 971-982, 1997.

[123] Galanti Silvia and Jung Alan Robert, "Low-Discrepancy Sequences: Monte Carlo Simulation of Option Prices," *Journal of Derivatives*, pp. 71-80, 1997.

[124] George Mekhtarian, "Composite Current Source (CCS) Modeling Technology Backgrounder," ©2005 Synopsys, Inc. 11/05.KF.WO .05-13816.

[125] Ratnakar Goyal, Naresh Kumar, "Current Based Delay Models: A Must For Nanometer Timing," Cadence Design Systems, Inc.

[126] Synopsys, "Liberty Library Modeling," http://www.synopsys.com/community/interoperability/pages/libertylibmodel.aspx., accessed date: 3rd July 2012.

[127] John F. Croix, D. F. Wong, "Blade and Razor: Cell and Interconnect Delay Analysis Using Current-Based Models," *Proceedings of the IEEE Design Automation Conference,* pp. 386-389, June 2-6, 2003.

[128] Jun Li, Hong Zhao, Hsien-Yen Chiu, "Accuracy Timing Models for Integrated Circuit Verification," U.S. patent number 6721929, 13th April 2004.

[129] Debasish Das, William Scott, Shahin Nazarian, Hai Zhou, "An Efficient Current-Based Logic Cell Model for Crosstalk Delay Analysis," *Proceedings of the IEEE Quality of Electronic Design Conference,* pp. 627 - 633, March 16-18, 2009.

[130] Michał Rewien´ski, Jacob White, "A Trajectory Piecewise-Linear Approach to Model Order Reduction and Fast Simulation of Nonlinear Circuits and Micromachined Devices," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 22, no. 2, pp. 155-170, February 2003.

[131] P. Li, Z. Feng, E. Acar, "Characterizing Multistage Nonlinear Drivers and Variability for Accurate Timing and Noise Analysis," *IEEE Transactions on very large scale integration (VLSI) Systems*, vol. 15, no. 11, pp. 1205-1214, November 2007.

[132] C. Visweswariah, K. Ravindran, K. Kalafala, S. G. Walker, S. Narayan, D. K. Beece, J. Piaget, N. Venkateswaran, J. G. Hemmett, "First-Order Incremental Block-Based Statistical Timing Analysis," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 25, issue10, pp. 2170-2180, October 2006.

[133] E. J. Gumbel "Statistical theory of extreme values and some practical applications," *Applied Mathematical*, Series 33, US Department of Commerce, National Bureau of Standards, 1954.

[134] J. Keiner and U. Waterhouse, "Fast Principal Components Analysis Method for Finance Problems with Unequal Time Steps," In P. L'Ecuyer and A. B. Owen editors, *Monte Carlo and Quasi-Monte Carlo Methods 2008*, Springer-Verlag, New York, 2010.

[135] Silicon Integration Initiative, "Statistical Methods For Semiconductor Chip Design," *Inc. (Si2™)*, Version 1.0, ISBN:1-882750-35-7, 2nd December 2008.

[136] Scott Roy, Private communication on PhD viva, 25th April 2012, School of Computer Science, the University of Manchester.