Efficient Algorithms for Fundamental Statistical Timing Analysis Problems in Delay Test Applications of VLSI Circuits

Von der Fakultät Informatik, Elektrotechnik und
Informationstechnik der Universität Stuttgart
zur Erlangung der Würde eines
Doktors der Naturwissenschaften (Dr. rer. nat.)
genehmigte Abhandlung

Vorgelegt von

Marcus Wagner

aus Jena, Deutschland

Hauptberichter:	Prof. Dr. rer. nat. Hans-Joachim Wunderlich
Mitberichter:	Prof. Dr. rer. nat. Bernd Becker
Tag der mündlichen Prüfung:	03. November 2016

Institut für Technische Informatik
der Universität Stuttgart

2016

Contents

List of Figures

Chapter 5

Chapter 6

Chapter 7

List of Tables

List of Abbreviations and Acronyms

ATPG	Automatic Test Pattern Generator
BVND	FORTRAN routine to compute the CDF of the bivariate normal distribution, developed by Alan Genz [Genz04]
CDF	Cumulative Distribution Function
CMOS	Complementary Metal-Oxide-Semiconductor
CMP	Chemical Mechanical Polishing
CPU	Central Processing Unit
DPPM	Defect Parts Per Million
FDP	Fault Detection Probability
FinFET	Fin Field Effect Transistor
GPGPU	General Purpose Computation on Graphics Processing Unit
HDL	Hardware Description Language
ITRS	International Technology Roadmap for Semiconductors
LER	Line Edge Roughness
LHS	Left-Hand Side
MPU	Microprocessor unit
MVNDST	FORTRAN routine to compute the CDF of the multivariate normal distribution, developed by Alan Genz [Genz92]
NR-test	Non-Robust test
NXP	NXP Semiconductors N.V. global semiconductor manufacturer
OPC	Optical Proximity Correction
PDF	Probability Density Function
PPI	Pseudo-Primary Input
PPO	Pseudo-Primary Output
PSA	Probabilistic Sensitization Analysis
RAM	Random-Access Memory
RDF	Random Dopant Fluctuation
RHS	Right-Hand Side
SDF	Standard Delay Format (IEEE standard for the representation and interpretation of timing data in electronic design process)
TPDFP	Target Paths Delay Fault Probability
TVTL	FORTRAN routine to compute the CDF of the trivariate normal distribution, developed by Alan Genz [Genz04]
VLSI	Very Large Scale Integration
XNOR	Digital logic gate whose function is the logical complement of the exclusive OR
XOR	Digital logic gate that implements the exclusive OR function

Units

GiB	gibibyte, 2^{30} bytes
KiB	kibibyte, 2^{10} bytes
MiB	mebibyte, 2^{20} byte

List of Mathematical Symbols and Notation

Symbol	Explanation
$\mathbf{0}$	zero vector
$\mathbf{0}_{k,l}$	$(k \times l)$ zero matrix
I_n	$(n \times n)$ identity matrix
A^T	transpose of matrix A
A^{-1}	inverse of matrix A
A^{-T}	inverse of transpose of matrix A
θ_0	nominal circuit instance $\theta_0 \in \Theta$
θ	randomly chosen circuit instance $\theta \in \Theta$
Θ	infinite sample space of circuit instances
M	skewness matrix
L	lower triangular Cholesky factor
Ψ	target paths delay fault probability
Ξ	delay fault detection probability
X	random variable X
$\phi(x)$	probability density function of standard normal distribution $\mathcal{N}(0,1)$
$\Phi(x)$	cumulative distribution function of standard normal distribution $\mathcal{N}(0,1)$
$\mathbb{P}(X \leq 0)$	short hand notation for probability of event $\{\theta \in \Theta : X(\theta) \leq 0\}$
$\mathbb{E}(X)$	expected value of random variable X
$\mathrm{Var}(X)$	variance of random variable X
$\boldsymbol{X}$	random vector $\boldsymbol{X}$
$\phi_2(x_1, x_2; \rho)$	probability density function of bivariate standard normal distribution with correlation coefficient ρ
$\phi_n(\boldsymbol{x}; \boldsymbol{\mu}, \Sigma)$	probability density function of n-dimensional normal distribution with mean vector $\boldsymbol{\mu}$ and full rank covariance matrix Σ
$\Phi_2(x_1, x_2; \rho)$	cumulative distribution function of bivariate standard normal distribution with correlation coefficient ρ
$\mathrm{Var}(\boldsymbol{X})$	covariance matrix of random vector $\boldsymbol{X}$
$\mathrm{Cov}(X_1, X_2)$	covariance between random variables X_1 and X_2
$\kappa_3(\boldsymbol{X})$	third multivariate cumulant of random vector $\boldsymbol{X}$
$\mathcal{N}_n(\boldsymbol{\mu}, \Sigma)$	n-dimensional normal distribution with mean vector $\boldsymbol{\mu}$ and covariance matrix Σ
$\mathcal{SN}_n(\boldsymbol{\mu}, \Sigma, \boldsymbol{\lambda})$	n-dimensional skew-normal distribution with mean vector $\boldsymbol{\mu}$, covariance matrix Σ and shape vector $\boldsymbol{\lambda}$

Acknowledgments

I would like to take this opportunity to thank all individuals who supported me during my time as a PhD student. I would particularly like to thank my thesis supervisor, Prof. Dr. rer. nat. Hans-Joachim Wunderlich, for his continuous support, ideas and advice. I also would like to thank my co-supervisor Prof. Dr. rer. nat. Bernd Becker for his commitment to assess my thesis in a timely manner.

A further thank you goes to my colleagues and students who at some time were involved in my research.

I would especially like to thank my parents for their kindness and unlimited support throughout my educational career. Without them, this work would not have been possible.

Abstract

Tremendous advances in semiconductor process technology are creating new challenges for the delay test of today's digital VLSI circuits. The complexity of state-of-the-art manufacturing processes does not only lead to greater process variability, it also makes today's integrated circuits more prone to defects such as resistive shorts and opens. As a consequence, some of the manufactured circuits do not meet the timing requirements set by the design specification. These circuits must be identified by delay testing and sorted out to ensure the quality of shipped products.

Due to the increasing process variability, key transistor and interconnect parameters must be modelled as random variables. These random variables capture the uncertainty caused by process variability, but also the impact of modelling errors and variations in the operating conditions of the circuits, such as the temperature or the supply voltage.

The important consequence for delay testing is that a particular delay test detects a delay fault of fixed size in only a subset of all manufactured circuits, which inevitably leads to the shipment of defective products. Despite the fact that this problem is well understood, today's delay test generation methods are unable to consider the distortion of the delay test results, caused by process variability. To analyse and predict the effectiveness of delay tests in a population of circuits which are functionally identical but have varying timing properties, statistical timing analysis is necessary. Although the large runtime of statistical timing analysis is a well known problem, little progress has been made in the development of efficient statistical timing analysis algorithms for the variability-aware delay test generation and delay fault simulation.

This dissertation proposes novel and efficient statistical timing analysis algorithms for the variability-aware delay test generation and delay fault simulation in presence of large delay variations. For the detection of path delay faults, a novel probabilistic sensitization analysis is presented which analyses the impact of process variations on the sensitization of the target paths. Furthermore, an efficient method for approximating the probability of detecting small delay faults is presented. Beyond that, efficient statistical SUM and MAX-operations are proposed, which provide the fundamental basis of block-based statistical timing analysis.

The experiment results demonstrate the high efficiency of the proposed algorithms.

German Abstract
—Zusammenfassung—

Die rasanten Fortschritte bei der Halbleiterprozesstechnologie führen zu immer neuen Herausforderungen beim Test heutiger hochintegrierter digitaler Schaltungen. Die steigende Komplexität der Herstellungsprozesse vergrößert nicht nur die Prozessvariabilität, sie erhöht auch die Anfälligkeit der Schaltungen gegenüber Fabrikationsdefekten, wie z.B. widerstandsbehafteter Kurzschlussdefekte oder widerstandsbehafteter Leiterunterbrechungen. In Folge dessen erfüllen einige der produzierten Schaltungen nicht die durch die Designspezifikation vorgegebenen zeitlichen Anforderungen. Diese Schaltungen müssen durch Verzögerungstests identifiziert und aussortiert werden, um die Qualität der ausgelieferten Produkte zu garantieren.

Aufgrund der zunehmenden Prozessvariabilität müssen wesentliche Parameter von Transistoren und Verbindungsleitungen durch Zufallsvariablen modelliert werden. Durch diese Zufallsvariablen wird die Unsicherheit dieser Parameter aufgrund von Prozessvariationen ausgedrückt, aber auch der Einfluss von Modellierungsfehlern und Schwankungen in den Betriebsbedingungen der Schaltung, wie z.B. der Temperatur oder der Versorgungsspannung, berücksichtigt.

Die wesentliche Konsequenz für den Verzögerungstest ist, dass ein bestimmter Verzögerungstest einen Verzögerungsfehler fester Größe in nur noch einem Teil der fehlerhaften Schaltungen erkennen kann, was unweigerlich zur Auslieferung defekter Schaltungen führt. Obwohl dieses Problem inzwischen gut bekannt ist, berücksichtigen die bestehenden Verfahren zur Testmusterzeugung die Verfälschung der Testergebnisse aufgrund von Prozessvariationen nicht. Um die Wirksamkeit von Verzögerungstests in einer Population von Schaltungen zu bewerten, die funktional identisch sind aber unterschiedliches Zeitverhalten besitzen, bedarf es einer statistischen Timing-Analyse. Obwohl der enorme Rechenaufwand einer solchen statistischen Timing-Analyse ein wohlbekanntes Problem ist, wurden bislang nur geringe Fortschritte bei der Entwicklung effizienter statistischer Timing-Analyse Algorithmen für die Berücksichtigung von Prozessvariationen bei der statistischen Verzögerungstestmustererzeugung und Verzögerungsfehlersimulation erzielt.

Die vorliegende Dissertation stellt neue und effiziente statistische Timing Analyse-Algorithmen für die statistische Verzögerungstestmustererzeugung und Verzögerungsfehlersimulation unter Berücksichtigung großer Prozessvariationen vor. Für die Erkennung von Pfadverzögerungsfehlern wird eine neuartige probabilistische Sensibilisierunganalyse präsentiert, welche den Einfluss von Prozessvariationen auf die Sensibilisierung der zu testenden Pfade analysiert. Ebenso wird ein effizientes Verfahren zur Bestimmung der Erkennungswahrscheinlichkeit kleiner Verzögerungsfehler vorgestellt. Überdies hinaus wurden effizientere statistische SUM und MAX-Operationen entwickelt, welche die fundamentale Grundlage der blockbasierten statistischen Timing-Analyse bilden. Die experimentellen Ergebnisse zeigen die hohe Effizienz der vorgestellten Algorithmen.

CHAPTER 1

Introduction

The increasing complexity of manufacturing processes makes today's integrated circuits more prone to defects such as resistive shorts and opens and also exacerbates the problem of accurately predicting the timing properties of the manufactured circuits. In order to ensure the quality of shipped products, delay testing must identify the manufactured circuits which fail the timing requirements.

The uncertainty introduced by the manufacturing process, modelling errors and variability in the circuit's operating conditions causes some delay tests to be effective in only a subset of all manufactured circuits, which can result in a large number of test escapes. Today's delay test generation tools are incapable of analysing the detrimental impact of process variations on the delay tests, which requires sophisticated statistical timing analysis algorithms.

The focus of this work is the development of efficient statistical timing analysis algorithms for the variability-aware delay fault simulation and delay test generation.

This chapter briefly introduces the reader to the challenges of semiconductor manufacturing and the main concepts of delay testing. The following sections highlight three fundamental problems of statistical timing analysis, which must be solved to analyse and compare the effectiveness of delay tests considering uncertain gate and interconnect delays. Finally, the organisation and contributions of this work are presented.

1.1 Trends and Challenges in Semiconductor Manufacturing

Over the past decades, the complexity of the semiconductor manufacturing process has been increasing rapidly. Whereas in 2000 a $0.25\mu m$ CMOS manufacturing process involved 25 mask levels with around 30 lithography steps [Frans04, p.260, 356], a state-of-the-art $18nm$ process requires 65 mask levels [ITRS12] and over 1000 manufacturing

process steps [Goel13]. Where a gate electrode was once formed by a doped silicon polysilicon film, now a stack of up to seven precisely formed metal films is required to achieve the desired threshold voltage, gate resistance and device stability [Schue13].

One of the key challenges of the manufacturing process is the availability of suitable lithography solutions. Today's state-of-the-art immersion 193nm lithography systems can only resolve features of approximately 40nm and extreme ultraviolet lithography (13.6 nm) is still not available due to the limited power and reliability of the available light sources. To continue scaling, reticle enhancement techniques such as optical proximity correction (OPC) with predistorted mask patterns are used. In addition, conventional single lithography exposure is supplanted by multi-patterning lithography which involves several exposure, etch and deposition steps [Bench09, Ma11]. Multi-patterning increases the number of design rule constraints, restricts cell placement and routing and ultimately increases the process complexity significantly.

Each additional manufacturing step bears the risk of introducing a defect by some contamination or impurity. Two of the most frequent types of physical defects are metal mousebites and metal slivers.

With continued scaling, transistor feature dimensions are approaching fundamental limits given by the atomic nature of matter. This does not only lead to greater variability of the electrical transistor parameters, the transistor performance also becomes more sensitive to variations of these parameters. For example, the saturation current I_{on} becomes more sensitive to variations of the effective channel length L_{eff}, carrier mobility and supply voltage U_{dd} over technology generations [Zhao06].

These so called *process variations* are widely known to play an increasingly important role in future technology scaling [Kuhn10, Leung12b, Kang13, Wang13]. The 3σ variability goals set by the international technology roadmap for semiconductors are shown in table 1.1. While the physical gate length is decreasing, the threshold voltage (U_{th}) and consequently the gate delay variability is expected to increase significantly. This delay variability will cause increasingly large circuit performance variations and additional complications for the circuit design sign-off.

▼ **Table 1.1 —** Variability limits set by International Technology Roadmap for Semiconductors (ITRS) [ITRS12]

	2015	2018	2021	2024	2026
MPU physical gate length (nm)	17	12.8	9.7	7.4	tbd
U_{dd} variability	10%	10%	10%	10%	10%
U_{th} variability	23%	28%	30%	35%	38%
channel length variability	12%	12%	12%	12%	12%
circuit performance variability	45%	50%	52%	57%	60%
contribution of variability to sign-off delay	18%	20%	26%	32%	38%

1.2 Delay Testing of VLSI circuits

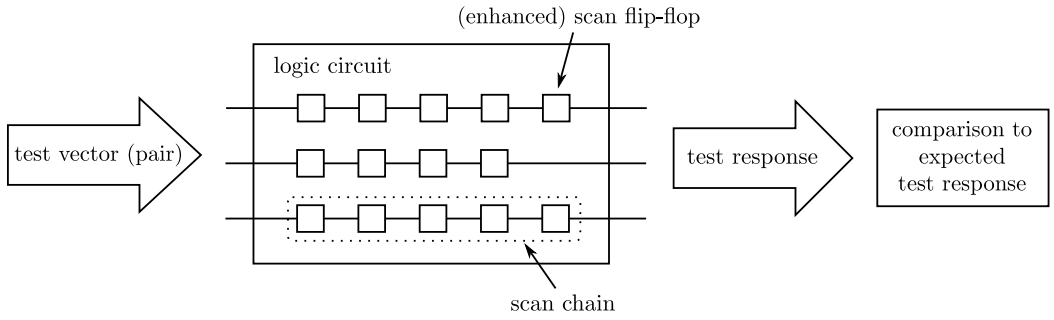
The purpose of delay testing is to detect timing defects in the circuit and to check that the circuit meets the specified timing requirements. Delay testing has become an essential part of the VLSI design testing process, due to the advances in process technology and the aggressive timing requirements of modern high speed designs. Delay testing is performed by loading the circuit into automatic test equipment, applying a test vector-pair to the circuit inputs and comparing the logic values observed after the clock cycle time at the circuit outputs with the expected fault free values. If an unexpected value is observed at any of the outputs, the circuit does not meet the timing requirements and is sorted out. Otherwise, the next test vector-pair is applied.

A circuit which does not fail for any applied test vector-pair is not necessarily defect free as it might fail under different operating conditions or when different test vector-pairs are applied. This is because modern VLSI circuits can have an extremely large number of paths and only a small subset can be tested by each test vector-pair.

The following subsections introduce the reader to important concepts of scan testing, important delay fault models and variation-aware delay-testing.

1.2.1 Scan Delay Test

One of the most widely used methods for VLSI circuit testing is scan test [Eiche77]. In scan test, the state elements like flip-flops and latches are joined together into shift registers called scan chains. During the normal operation of the circuit, the scan chain is disabled. However in scan mode, the scan chain is used to shift in the next test vector while shifting out the combinational output response to the previous test vector [Wang06]. In practise, often multiple scan chains are used in parallel to reduce the number of shift cycles, as shown in fig. 1.1.



▲ **Figure 1.1 —** Delay testing using (enhanced) scan testing

The test vector-pair can be applied in three different ways: launch-on-capture [Savir94], launch-on-shift [Savir93] and enhanced-scan [Dervi91]. In *launch-on-capture*, the first test vector is shifted into the scan chain at a slow speed. The second test vector is defined by the next state values, which are computed by the circuit one clock cycle after

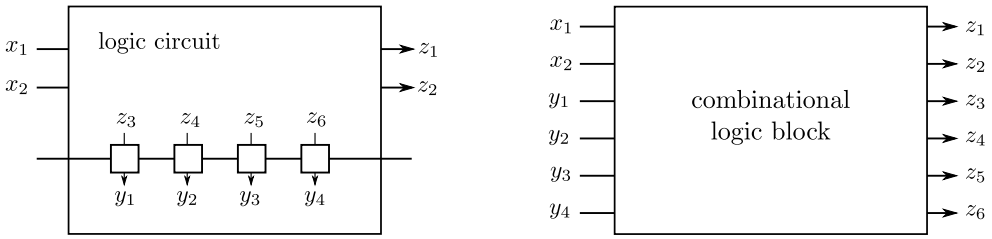
the application of the first test vector. The test response is captured after the functional at-speed clock cycle time.

In *launch-on-shift*, the first $k-1$ bits of the first k-bit test vector are shifted into the scan chain at slow speed. The second test vector is obtained by shifting the last bit into the scan chain, which creates the desired input transitions. Again, the test response is captured after the functional at-speed clock cycle time.

Finally, in *enhanced-scan testing*, both vectors of the test vector-pair are shifted into the scan chain. At first, the first test vector is shifted into the scan chain and it is immediately applied to initialize the circuit. The outputs of the scan chain are preserved while the second test vector is shifted into the scan chain. Once the last bit of the second vector has been shifted into the scan chain, the scan chain outputs are changed to the second test vector and the test response is captured after the functional at-speed clock cycle time. For enhanced-scan, the flip-flops are required to stored two bits that can be applied consecutively to the combinational logic driven by the scan cells.

After the application of a test vector-pair, the captured test response is shifted out of the scan chain and compared to the expected fault free test response. If the test response differs from the expected test response, then a fault has been detected.

In a full scan design, every state element in the circuit is part of a scan chain. In this case, delay test algorithms can ignore the state elements and instead focus on the combinational logic block between the state elements. This idea is illustrated by an example. Figure 1.2a shows a logic circuit, where all state elements are arranged into a scan chain. Because the value of these state elements can be directly observed and controlled, the inputs of these state elements can be considered as pseudo-primary outputs and the outputs of the state elements can be considered as pseudo-primary inputs of the combinational logic block [Bushn00, p.135], which is shown in fig. 1.2b.



(a) Full scan design (b) Combinational equivalent

▲ **Figure 1.2 —** Conversion of full scan design to combinational equivalent circuit

It is important to observe that this so called *combinational equivalent* circuit is independent of the number and structure of the scan chains. Hence, by assuming a full scan design, the algorithms discussed and presented in this thesis can ignore the state elements and operate directly on the combinational equivalent circuit without loss of generality.

A *structural path* of a circuit is an ordered list of gates $g_1, \ldots, g_n$ such that the output of gate g_i is connected to an input of gate g_{i+1} for $1 \leq i < n$. Furthermore, an input of g_1 is connected to an input of the circuit and the output of g_n is connected to an output of the circuit. The term *logical path* refers to a structural path with either a rising or a falling transition at the beginning of the path. This work is mostly concerned with logical paths so that in the following, "path" is used to refer to a logical path.

Definition 1.1 (sensitized path)**.** A logical path is *sensitized* by a test vector-pair if the transition at the start of the path is propagated along the path to the end of the path.

Definition 1.2 (target path)**.** A *target path* of a test vector-pair is a logical path that is sensitized by the test vector-pair in the circuit with nominal gate delays (nominal circuit instance). In this case, it is said that the test vector-pair *targets* the (logical) path.

1.2.2 Delay Fault Models

In this work, it is assumed that the delay of each gate is defined depending on a rising or falling transition at the gate output and the path along which a transition is propagated from a gate input to the gate output (pin-to-pin delay). The interconnect delay model also distinguishes between rising and falling transitions. For simplicity, the term "gate delay" will be used to denote the sum of pin-to-pin and interconnect delays for a rising or a falling transition at the gate output.

The defects introduced during the manufacturing process are modelled at higher levels of abstractions to alleviate the test generation complexity [Wang06]. A fault model represents many physical defects with a single fault at a suitable abstraction level. Two delay fault models are considered in this work: the gate delay fault model and the path delay fault model.

The *gate delay fault model* assumes that the defect affects only the delay of a single gate in the circuit [Iyeng88a, Iyeng88b]. The additional propagation delay of the gate introduced by the defect is called the size of the delay fault. A gate delay fault whose size is smaller than the clock cycle time is called *small delay fault*, otherwise it is called *gross-delay fault*.

The *path delay fault model* assumes that a circuit is faulty if any of its paths exceeds a specified delay, which in most cases is the circuit's functional clock cycle time [Smith85]. The delay of a path is the sum of the corresponding gate delay values along the path. The path delay fault model is more comprehensive than the gate delay fault model as it captures the combined effects of physical defects and process variations. Thus, a path delay fault may be caused by process variations, by a defect or by a combination of both. However, the path delay fault model may require much more test generation and test application time because VLSI circuits can have a huge number of paths.

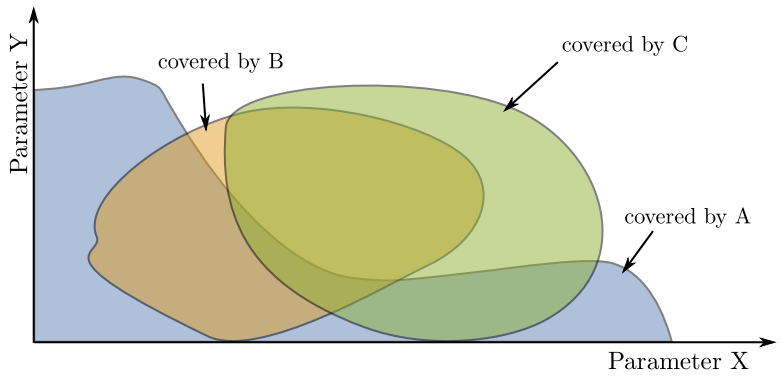
Many more specialized fault models exist. For example, the *resistive bridging fault model* assumes that a defects (e.g. metal sliver) connects two or more interconnects with some specific resistance, which should not be connected [Ingel11]. This connection may then cause a gate delay fault [Hopsc10].

1.2.3 Variation-Aware Delay-Testing

The increasing manufacturing process variations can no longer be ignored in delay testing and must be considered during delay test generation to achieve a sufficiently high defect coverage [Becke10]. Under the impact of process variations, the detectability of delay faults can be very different in different manufacturing samples. A given test vector-pair might detect a particular delay fault of fixed size in one circuit sample, but fail to detect the same fault in another circuit sample. In the latter, the fault might still be detectable by a different test vector-pair or be provably undetectable.

Classical delay test quality measures fail to capture the detrimental impact of process variations on the delay test and new quality measures have been proposed for variation-aware delay testing, which are based on the probability of detecting each delay fault by a given set of test vector-pairs. The efficient computation of a large number of these probabilities requires very sophisticated statistical timing analysis algorithms. This is not only because multiple test vector-pairs may be required to detect a particular delay fault of fixed size with sufficiently high probability, but also because each test vector-pair can detect multiple delay faults and the total number of test vector-pairs may not exceed some given upper bound to limit test cost. Hence, statistical timing analysis must compute these probabilities for a large number of alternative choices of test vector-pairs to guide the delay test optimization process.

The optimization problem is illustrated by an example. Figure 1.3 visualizes the circuit samples in which test vector-pairs 'A', 'B' and 'C' can detect a particular delay fault of fixed size. If the initial set of test vector-pairs {'A'} is extended by test vector-pair 'B', then the probability of detecting the delay fault increases significantly. By adding another test vector-pair 'C', this probability increases further. However, now the contribution of the test vector-pair 'B' to the detection of the delay fault becomes negligible and 'B' can be removed to limit test cost. Hence, a good solution to the given optimization problem is the set of test vector-pairs {A,C}.



▲ **Figure 1.3 —** Circuit samples in which the same delay fault of fixed size is detected by test vector-pair 'A', 'B' and 'C'

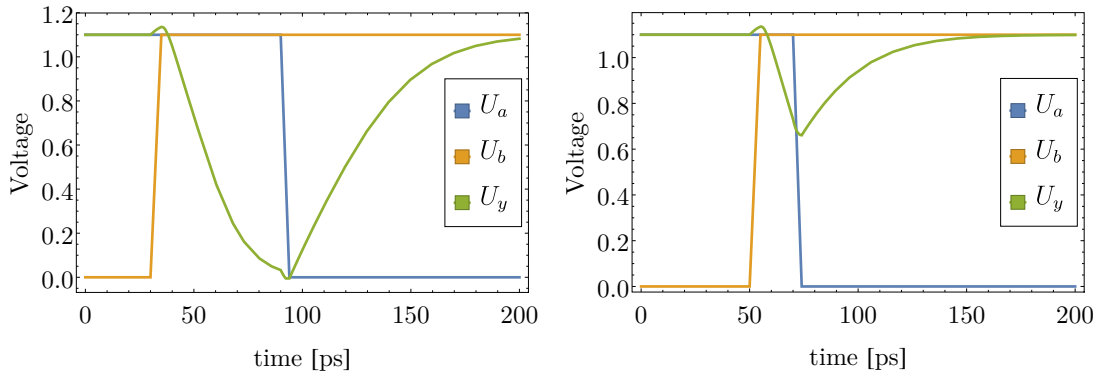
1.3 Statistical Timing Analysis for Path Delay Fault Testing

Path delay fault tests are typically applied to a selected set of near critical paths to characterize the speed of a circuit after manufacturing. These tests can also be repeated at multiple different clock frequencies for speed-binning.

1.3.1 Transition Propagation Condition

A physical gate can only propagate a subset of all transitions at the gate inputs to the gate output. Whether or not a particular input transition is propagated depends on many factors including the gate input and output voltages at the time of the input transition and the capacitive load at the gate output. One important condition for an input transition to be propagated is that the logic function of the gate must imply a change in the gate output logic value in response to the input transition. For example, a NAND gate with two inputs will only propagate a transition at one of its inputs if the other input has logic value '1'.

Another important condition applies to glitches, which are electrical pulses of short duration. During the application of a test vector-pair, one or multiple glitches may appear at a gate output before the output settles to the correct logic value. If a gate output transition is part of a glitch, then this glitch must be sufficiently long so that the gate is able to fully charge and fully discharge the capacitive load at the gate output. For example, fig. 1.4 depicts the waveforms of the input and output voltages of a NAND-gate with two inputs. It shows that if the falling transition at input 'a' (blue) occurs slightly earlier and the rising transition at input 'b' (orange) is slightly delayed, then the glitch at the gate output disappears.



(a) glitch occurs (b) glitch is filtered

▲ **Figure 1.4** — SPICE simulation results of a 2-input NAND gate with a falling transition at input 'a', a rising transition at input 'b' and a glitch at output 'y'

1.3.2 Invalidation of Path Delay Fault Tests

During the operation of the circuit, the transitions at the inputs of a gate can interact in complex ways depending on the order and the time intervals between the transitions [Konuk00]. As a consequence of the gate delay variability, the propagation of a transition may become blocked by another transition or redirected along a different path. In other words, a path which is sensitized by a test vector-pair in the circuit with nominal gate delays might not be sensitized under the impact of process variations.

Definition 1.3. A path delay fault test is called *invalidated*, if the delay test fails to sensitize the path it is supposed to test.

Path delay fault test invalidation causes two major problems for the delay test:

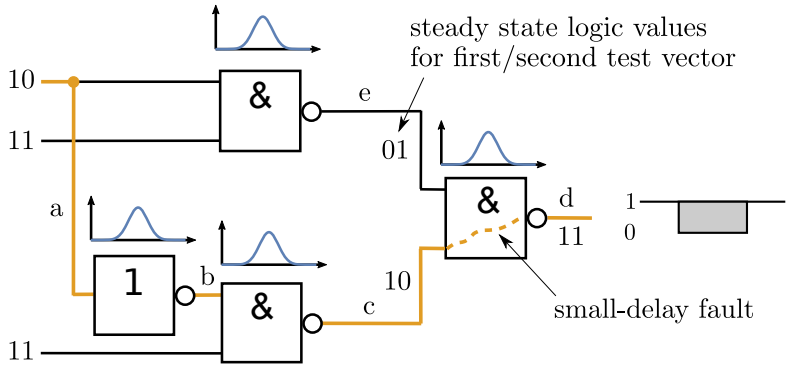
- A test vector-pair which sensitizes a particular path in the circuit with nominal gate delays (nominal circuit instance) might not sensitize the same path under the impact of process variations, leading to test escapes.
- A path which cannot be sensitized during the normal operation of the circuit is not required to meet the timing specification of the circuit. However, such a path might become sensitized by some test vector-pair due to delay variations, resulting in yield loss [Cheng93].

Invalidation of a path delay fault test can occur if no other path delay fault exists in the circuit and an off-path input stabilizes to the non-controlling value after the transition at the on-path input [Konuk00]. If the inertial delay property of physical gates is also considered, then test invalidation is possible even if the off-path transition occurs before the transition at the on-path input [Devad92]. In the following, one of the invalidation mechanisms of a path delay fault test is illustrated by an example.

The circuit in fig. 1.5 contains a small delay fault in the last NAND gate which is activated by a falling transition at 'c' that causes a rising transition at 'd'. The path delay fault test of path a-b-c-d creates a falling transition at 'c' and a rising transition at 'e', resulting in a glitch at the output 'd' of the last NAND gate. The faulty logic value '0' caused by the glitch can be detected during the delay test.

It is assumed, that the clock cycle time is 1.75ns. Due to the small delay fault, the path a-b-c-d has a nominal delay of 1.85ns and therefore a path delay fault. All other paths have delays less than 1.75ns. In those manufactured circuits in which, during the application of the test vector-pair, the falling transition at 'c' occurs when the off-path input 'e' still has the logic value '0', does the gate output 'd' remain constant '1' so that the path delay fault test is invalidate. Only in those manufactured circuits, where the off-path input has already stabilized to the non-controlling value when the transition at 'c' occurs, can the path delay fault be detected.

It is important to note that even if the off-path input 'e' has stabilized to '1' well before the falling transition at the on-path input 'c', the glitch might not appear at the gate output 'd', as explained in section 1.3.1. As shown in fig. 1.4, if the gate output voltage



▲ **Figure 1.5 —** Invalidation of a path delay fault test for the orange path a-b-c-d, which has a path delay fault [Konuk00]

has not dropped below 50% by the time the falling transition appears at the on-path input 'c', then the gate output voltage will rise immediately and no glitch will occur at the gate output.

The reader should note that the small delay fault in fig. 1.5 is not necessary for the path a-b-c-d to have a path delay fault, because process variability alone can cause the delay of each gate (pin-to-pin delay) along the path a-b-c-d to increase so that the path delay exceeds the clock cycle time. This slightly modified example shows that path delay fault test invalidation can occur regardless of any small delay fault in the circuit.

1.3.3 Probabilistic Sensitization Analysis

It is well known that often many different path delay fault tests exist for a sensitizable path [Cheng94, Egger11]. To reduce the number of test escapes due to path delay fault test invalidation, the tolerance of different path delay fault tests towards delay variations must be evaluated to identify those tests with a sufficiently small probability of being invalidated. In particular, statistical timing analysis is required to solve the following problems:

Detection of test invalidation mechanisms: Given sufficiently detailed knowledge of the test invalidation mechanism, such as the location of the affected target path off-path input, the test generation process can carefully optimize the test vector-pair to minimize the risk of test invalidation.

Computation of test invalidation probability: Under the impact of large process variations, all possible path delay fault tests of a target path may be affected by path delay fault test invalidation. In this case, it is necessary to select those path delay fault tests with a sufficiently low probability of test invalidation or to apply a suitable combination of path delay fault tests to the target path.

The underlying fundamental statistical timing analysis problem is stated in section 4.1.

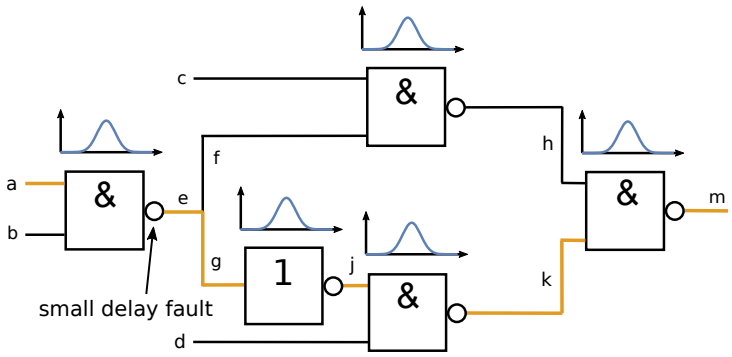
1.4 Statistical Timing Analysis for Small Delay Fault Testing

Recent studies show that a large fraction of delay faults in the latest technology nodes are caused by small-delay faults [Matti09, Ahmed06], which are also becoming increasingly difficult to detect [Ryan14]. While a small-delay fault of size 20 ps would unlikely affect the circuit operation at 100 MHz (0.2% of the clock cycle time), the same fault is much more likely to cause a timing failure at 5 GHz (10% of clock cycle time). Furthermore, certain types of small-delay faults can grow over time [Segur02] and therefore pose a reliability risk. Hence, delay testing for small-delay faults is extremely important for the quality and reliability of CMOS circuits.

1.4.1 Testing for Small Delay Faults

For a gate delay fault to be detectable, the fault size must be sufficiently large to cause at least one path through the small delay fault site to have a path delay fault. The smallest detectable delay fault size is equal to the slack of the longest sensitizable path through the delay fault site. The slack of a path is defined as the difference between the clock cycle time and the delay of the path.

The necessity of testing for small delay faults by exercising the longest path has been demonstrated in high volume production manufacturing test [Daasc07]. For example in fig. 1.6, a small delay fault at line 'e' may be missed by testing the path a-e-f-h-m because the delay of the path is too small [Goel13]. By testing the path a-e-g-j-k-m, which has a much larger delay, can the small delay fault be detected.



▲ **Figure 1.6** — Detection of small delay fault at gate output 'e' over the longest sensitizable path (orange) through the fault site [Goel13]

The impact of certain types of faults depends on many circuit parameters, which are unknown due to process variations [Erb14], but also on the operating conditions of the circuit. Repeating the delay test under several different operation conditions can aid the detection of small delay faults. The major advantages and disadvantages of adjusting the supply voltage, the temperature and the clock frequency are summarized in the following.

Testing at very low voltage was shown to significantly increase the effect of certain defects that cause small delay faults, such as resistive shorts [Hao93, Chang96, Chang98]. However, this technique may increase the test time due to the reduced switching speed of the transistors in the scan chain.

Testing at very low temperatures (e.g. 0°C) was also shown to be effective to detect small delay faults caused by resistive salicide [Needh98, Tseng00]. However, this method presents technological challenges and might require more expensive circuit cooling solutions, especially for state-of-the-art high performance circuits.

Delay test at higher than operational clock frequencies (faster-than at speed test) reduces the slack of all paths and therefore promotes the detection of small delay faults of smaller size [Kruse04, Lee05a]. However, this approach may increases the risk of yield loss due to hazard capture and it also exacerbates the already-well-known issues of peak power and IR-drop during manufacturing test.

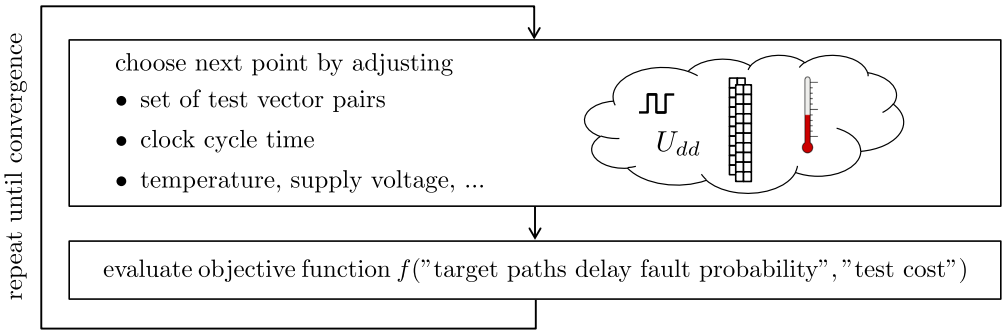
1.4.2 Invalidation and Optimization of Small Delay Fault Tests

Under the impact of process variations, a different path can be the longest path through the small delay fault site in different circuits and under different operating conditions. It is therefore widely accepted that conventional delay tests for small delay faults must be extended to account for the detrimental impact of process variations on the delay test [Ingel09, Becke10, Polia11].

Even if a small delay fault is undetectable under some operating conditions, the fault could significantly reduce the timing margins and cause timing failures under different operating conditions. Testing all sensitizable paths through the small delay fault site under all admissible operating conditions would guarantee the detection of the small delay fault. However, this kind of exhaustive testing is impractical since, depending on the circuit structure, the number of paths can increase exponentially with the circuit size. Furthermore, there may be an infinite number of admissible operating conditions.

In order to find a suitable compromise between the delay test quality and the test cost, it is necessary to optimize the set of test vector-pairs [Becke10, Czutr12, Sauer14] and all delay test parameters [Needh98, Fonse10]. The set of *delay test parameters* includes the supply voltage [Chang98], the temperature [Tseng00], the clock cycle time [Chakr12] and possibly the masking of the combinational network outputs for faster-than-at-speed testing [Lee05a]. The choice of the burn-in time also has a significant impact on the detectability of small delay faults [Sumik12].

The delay test optimization procedure is illustrated in fig. 1.7. The process starts by adjusting the delay test parameters and the set of test vector-pairs to improve the delay test quality and the test cost. Afterwards, the resulting change of the delay test quality and the test cost is evaluated to obtain the new value of the objective function, which is to be maximized. Based on the result, further adjustments to the set of test vector-pairs and the operating conditions are made and this process continues until a suitable compromise has been found.



▲ **Figure 1.7 —** Small delay fault test optimization

It is apparent that the computational cost for solving this multidimensional optimization problem strongly depends on the objective function, which involves the probability that at least one of the target paths has a path delay fault. The stepwise optimization process involves a huge number of probability evaluations and therefore requires very sophisticated statistical timing analysis methods.

1.4.3 Computation of Target Paths Delay Fault Probability

One of the fundamental problems for the optimization of small delay fault tests is the efficient computation of the probability, that at least one of the target paths (see definition 1.2) has a path delay fault. Formally, the fundamental statistical timing analysis problem is described by the following two definitions.

Definition 1.4. Given a set of k test vector-pairs and let $X_{i,1}, \ldots, X_{i,n_i}$ denote the delays of the n_i target paths of the i-th test vector-pair. Then the *maximum delay of the target paths* Z is defined as

$$Z = \max(Y_1, \ldots, Y_k), \tag{1.1}$$

where $Y_i = \max(X_{i,1}, \ldots, X_{i,n_i})$ for all $1 \leq i \leq k$.

Above definition is not only applicable to a small delay fault of fixed size, but it can also capture the distribution of the delay fault size as part of the delay distributions of those target paths, which contain the faulty gate.

Definition 1.5 (target paths delay fault probability)**.** Let Z denote the maximum delay of the target paths of a given set of test vector-pairs. Then the *target paths delay fault probability* Ψ is defined as

$$\Psi = \mathbb{P}(Z > T_{clk}), \tag{1.2}$$

where T_{clk} denotes the clock cycle time.

The main difficulty for computing the target paths delay fault probability is caused by the relationship between the random variables. To express these relationships, the individual path delays must be grouped into a random vector, which is described by

a joint probability distribution (e.g. a multivariate normal distribution). Computing the target paths delay fault probability requires numerical integration over the joint probability distribution, which is computationally very expensive in general.

1.5 Variation-Aware Delay Fault Simulation

The goal of the variation-aware delay fault simulation is to evaluate the delay test response of a circuit with any of a list of delay faults during the application of a set of test vector-pairs [Radha11]. By comparing the responses to the specification of the circuit, the probability of a circuit passing/failing the delay test is determined. The fault simulation also provides vital information for the prediction of important test metrics such as defective parts per million (DPPM) and yield loss [Qian10].

An easy way of conducting the fault simulation is by injecting a delay fault into the circuit followed by a Monte-Carlo simulation with each test vector-pair to evaluate the delay test response under the impact of process variations. However, since there are many small delay faults and the fault simulation has to be repeated for different circuit operating conditions, a Monte-Carlo simulation based approach is inefficient or even infeasible.

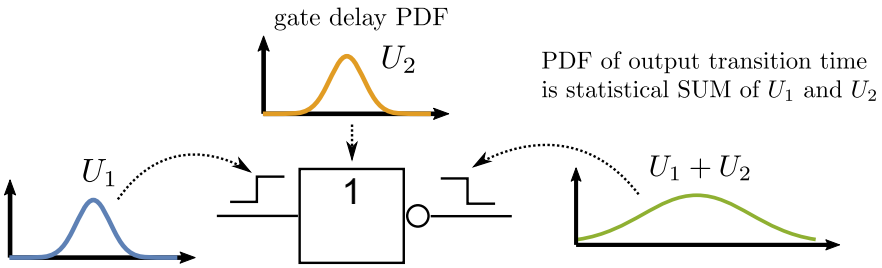
A more suitable approach is based on block-based statistical timing analysis, which is described in section 1.5.1. The problem of efficiently computing the fundamental statistical operations is explained in section 1.5.2.

1.5.1 Block-Based Statistical Timing Analysis

In block-based statistical timing analysis, the delay values of a gate and the arrival time of a transition are represented by random variables of some known probability distribution. A marginally detectable small delay fault can be modelled by increasing the mean of one of multiple delay values of a gate by a fixed fault size.

During the simulation of a test vector-pair, the transitions at the circuit inputs are propagated along the sensitized paths to the circuit outputs [Lee05b]. At each gate, the analysis computes the distribution of the arrival time of the last transition at the gate output. For a gate with one input, the gate delay is added to the arrival time of the transition at the gate input to obtain the arrival time of the transition at the gate output. This is done using the statistical SUM-operation as shown in fig. 1.8.

For a gate with two inputs, the steady state logic value of the gate output is known from the second test vector. If the output is switching because a gate input is switching to the non-controlling value, then the maximum of the sum of arrival times and pin-to-pin delays of all gate inputs that switch to the non-controlling value must be computed. For example, let U_1 and U_2 denote the arrival times of two rising transitions at a NAND gate, as shown in fig. 1.9. The arrival time of the transition at the gate output is $\max(U_1 + D_1, U_2 + D_2)$, where D_1 and D_2 denote the gate delays for propagating the transition at the first and the second gate input, respectively, to the gate output.

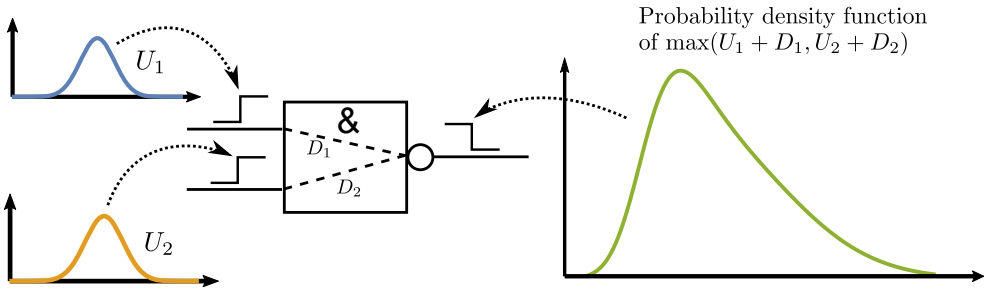


▲ **Figure 1.8 —** Statistical SUM-operation

On the other hand, if the gate output is switching because any gate input is switching to a controlling value, then the earliest arrival time of those gate input transitions at the gate output must be computed. For example, if both rising transitions at the gate inputs in fig. 1.9 are replaced by falling transitions, then the arrival time of the transition at the gate output is $\min(U_1 + D_1, U_2 + D_2)$. The problem of computing the minimum of two random variables X_1 and X_2 can always be transformed into the problem of computing the maximum using the identity

$$\min(X_1, X_2) = -\max(-X_1, -X_2). \tag{1.3}$$

It is apparent that the accuracy and the computational cost of block-based statistical timing analysis is determined by the efficiency of the statistical SUM and MAX-operations.



▲ **Figure 1.9 —** Statistical MAX-operation

1.5.2 Efficient Computation of Statistical SUM and MAX-Operations

The main problem of block-based statistical timing analysis is the limited accuracy of the SUM and the inaccuracy of the MAX-operation.

The SUM-operation is usually easy to compute efficiently. Many families of multivariate distributions, such as the family of multivariate normal distributions, are closed under affine transformations. Therefore, the exact distribution of the sum of random variables can usually be computed using closed form formulas.

The efficient computation of the MAX-operation is, on the other hand, one of the most challenging problems of statistical timing analysis. Only very few exotic families of distributions are closed under maximum [Balas91] and these are of little practical value for statistical timing analysis. In other words, for the vast majority of probability distributions, the distribution of the maximum $\max(X_1, X_2)$ is not a member of the same family of distributions as the distributions of X_1 and X_2. For example, if X_1 and X_2 have a normal distribution then it is well known that the distribution of $\max(X_1, X_2)$ belongs to a much more complicated family of distributions [Nadar08]. To preserve the efficiency of block-based statistical timing analysis, it is necessary to approximate the distribution of the maximum $\max(X_1, X_2)$ with a member of the same family of distributions to which the distributions of the operands X_1 and X_2 belong.

Any approach which can efficiently compute the statistical SUM and the statistical MAX-operation can immediately be applied to many other statistical timing analysis problems. For example, the SUM-operation can be used to compute the delay distribution of the sensitized paths and the MAX-operation can be applied to compute the distribution of the maximum delay of the target paths, as defined in definition 1.4.

1.6 Organisation and Contributions of this Work

This work targets key statistical timing analysis problems affecting delay test applications in innovative technology nodes.

Chapter 2 introduces the reader to the sources and classification of variability as well as to the formal foundation on which subsequent chapters build upon. This includes definitions of essential terms and concepts in probability theory and circuit modelling.

Chapter 3 provides a concise summary of the state of the art in statistical timing analysis for the evaluation of path and small delay fault tests. Recent advances in the computation of the statistical MAX-operation are reviewed. The scalability and limitations of existing methods are discussed in the context of variation-aware delay fault testing.

Chapter 4 presents a novel probabilistic sensitization analysis, which detects for a given test vector-pair the most likely path delay fault test invalidation mechanisms and identifies the paths that are responsible for the invalidation. The information provided by the analysis allows the delay test generation to specifically target the location and mechanism of path delay fault test invalidation. The proposed method is applied to all critical paths (definition 2.17) that are sensitizable by a given test vector-pair under the impact of delay variations. The result of the analysis is a subcircuit that gives the same delay test response to the application of the test vector-pair as the complete circuit with almost 100% probability, while requiring only a fraction of the computational cost for a Monte-Carlo simulation.

Chapter 5 presents novel algorithms, which compute and incrementally update an approximation of the target paths delay fault probability after delay test parameter

modifications. In particular, the insertion/removal of a test vector-pair, the masking/unmasking of primary outputs and changes of test clock cycle time T_{clk} are addressed. The accuracy of the results does not depend on the type or the order in which these delay test parameter are modified. Furthermore, the new algorithms only require that the joint delay distribution of sufficiently long target paths can be accurately approximated by a multivariate normal distribution. In other words, the algorithms are not restricted to normally distributed gate and interconnect delays.

Chapter 6 introduces the novel skew-normal distribution based SUM and MAX-operations, which significantly reduce the approximation error compared to the normal distribution based SUM and MAX-operations. Several optimizations are presented, which make the new operations applicable to a large number of random variables.

Chapter 7 presents the experimental results for several large benchmark circuits.

Chapter 8 summarizes the findings of this work and starting points for further research are pointed out.

CHAPTER 2

Fundamentals of Statistical Timing Analysis

Essential facts from linear algebra are reviewed in section 2.1. The major sources of variations are introduced in section 2.2. In order to create a compact gate delay model considering many sources of variability, it is beneficial to distinguish between different classes of process variations, which are explained in section 2.3. The variations in gate and interconnect delays are formally described in section 2.4 by random variables with known probability distributions. This section also reviews important concepts of probability theory, which is an integral component of all statistical timing analysis algorithms. The circuit model, which will be used throughout this work, is defined in section 2.5.

2.1 Review of Linear Algebra

This section briefly reviews important facts from linear algebra, which are essential for the understanding of this thesis. Unless explicitly cited otherwise, the following definitions and facts can be found in [Golub13].

The *indicator function* is

$$\mathbb{1}_A(x) := \begin{cases} 1 & \text{if } x \in A, \\ 0 & \text{if } x \notin A. \end{cases} \tag{2.1}$$

The *eigenvalues* of a real valued matrix $A \in \mathbb{R}^{n\times n}$ are the roots of the characteristic polynomial

$$p(x) = \det(A - xI_n), \tag{2.2}$$

where $I_n \in \mathbb{R}^{n\times n}$ is the identity matrix. Hence, A has exactly n not necessarily distinct eigenvalues. If A is symmetric, then all its eigenvalues are real.

If $H \in \mathbb{R}^{n\times n}$ is non-singular and $B = H^{-1}AH$, then A and B are called *similar*. If two matrices are similar, they have exactly the same eigenvalues.

If ψ is an eigenvalue of A, then there exists a nonzero vector $\boldsymbol{x}$ so that $A\boldsymbol{x} = \psi\boldsymbol{x}$. In this case, $\boldsymbol{x}$ is said to be an *eigenvector* of A associated with the eigenvalue ψ. Throughout this work, it is assumed that the eigenvectors are normalized to unit magnitude.

The greatest eigenvalue of A is called *dominant eigenvalue* and the corresponding eigenvector is called *dominant eigenvector*.

Closed form formulas for the computation of the eigenvalues exist for $2 \leq n \leq 4$ and can be used to efficiently compute the eigenvectors [Kopp08]. However, several numerical issues due to floating point round-off errors must be solved to obtain a numerically stable algorithm. One of the main problems is the exact computation of the rank of $A - \psi I_n$. For higher dimensions, the eigenvalues and the eigenvectors can be computed using iterative methods.

Definition 2.1. A matrix $A \in \mathbb{R}^{n\times n}$ is positive definite if $\boldsymbol{x}^T A\boldsymbol{x} > 0$ for all non-zero $\boldsymbol{x} \in \mathbb{R}^n$ and positive semidefinite if $\boldsymbol{x}^T A\boldsymbol{x} \geq 0$ for all $\boldsymbol{x} \in \mathbb{R}^n$.

For example, the matrix

$$B = \begin{pmatrix} 1 & -2 & 3 \\ -2 & 13 & -3 \\ 3 & -3 & 19 \end{pmatrix} \tag{2.3}$$

is positive definite, because for any vector $\boldsymbol{x} = (x_1, x_2, x_3)^T \in \mathbb{R}^3$

$$\begin{aligned} \boldsymbol{x}^T B\boldsymbol{x} &= x_1(x_1 - 2x_2 + 3x_3) + x_2(-2x_1 + 13x_2 - 3x_3) + x_3(3x_1 - 3x_2 + 19x_3) \\ &= x_1^2 - 4x_1x_2 + 13x_2^2 + 6x_1x_3 - 6x_2x_3 + 19x_3^2 \\ &= 6x_2^2 + 7x_3^2 + (x_1 - 2x_2 + 3x_3)^2 + 3(x_2 + x_3)^2 \end{aligned}$$

is non-negative and zero only for $x_1 = x_2 = x_3 = 0$. Clearly, any positive definite matrix is also positive semidefinite. It is also obvious that a positive definite matrix A is non-singular. Otherwise, there would be a non-zero vector $\boldsymbol{x}$ such that $\boldsymbol{x}^T A\boldsymbol{x} = 0$.

Lemma 2.2. *If $A \in \mathbb{R}^{n\times n}$ is positive semidefinite, then all its eigenvalues are non-negative.*

Proof. Let $\boldsymbol{x} \in \mathbb{R}^n$ be an eigenvector of A corresponding to the eigenvalue ψ, such that $A\boldsymbol{x} = \psi\boldsymbol{x}$. Left multiplication with $\boldsymbol{x}^T$ gives $\boldsymbol{x}^T A\boldsymbol{x} = \psi\boldsymbol{x}^T\boldsymbol{x}$, where $\boldsymbol{x}^T\boldsymbol{x} > 0$ and $\boldsymbol{x}^T A\boldsymbol{x} \geq 0$ because A is positive semidefinite. Therefore, $\psi \geq 0$. □

Lemma 2.3. *If $A \in \mathbb{R}^{n\times n}$ is a positive definite matrix and $\boldsymbol{v} \in \mathbb{R}^n$, then $A + \boldsymbol{v}\boldsymbol{v}^T$ is positive definite.*

Proof. It must be shown that $\boldsymbol{x}^T A\boldsymbol{v} + \boldsymbol{x}^T\boldsymbol{v}\boldsymbol{v}^T\boldsymbol{x} > 0$ for all non-zero $\boldsymbol{x} \in \mathbb{R}^n$. Clearly, $\boldsymbol{x}^T\boldsymbol{v}\boldsymbol{v}^T\boldsymbol{x} = (\boldsymbol{x}^T\boldsymbol{v})(\boldsymbol{x}^T\boldsymbol{v})^T \geq 0$ and $\boldsymbol{x}^T A\boldsymbol{x} > 0$ holds because A is positive definite. □

Theorem 2.4. *If $A \in \mathbb{R}^{n\times n}$ is a positive definite matrix and the matrix $G \in \mathbb{R}^{n\times k}$ has rank k, then $B = G^T AG \in \mathbb{R}^{k\times k}$ is also positive definite.*

The proof can be found in [Golub13, Theorem 4.2.1]. One of the important implication of this theorem is that if $A \in \mathbb{R}^{n\times n}$ is positive definite, then $A^{-T}AA^{-1} = A^{-T}$ is positive definite. In particular, if A is symmetric positive definite then A^{-1} is also symmetric positive definite.

The following theorem shows that deleting the ith row and column of a symmetric positive definite matrix preserves positive definiteness, which will be used in section 6.4.

Corollary 2.5. *If A is positive definite, then all its principal submatrices are positive definite. In particular, all the diagonal entries are positive.*

The proof can be found in [Golub13, Corollary 4.2.2].

Theorem 2.6 (Cholesky Factorization)**.** *If $A \in \mathbb{R}^{n\times n}$ is symmetric positive definite, then there exists a unique lower triangular $L \in \mathbb{R}^{n\times n}$ with positive diagonal entries such that $A = LL^T$.*

The proof can be found in [Golub13, Theorem 4.2.7]. This factorization is called *Cholesky factorization* (or Cholesky decomposition) and the matrix L is called the (lower) *Cholesky factor*.

Algorithm 2.1 is one of several possible algorithms for computing the lower Cholesky factor L. The computation of the Cholesky factor using this algorithm requires $n^3/3$ floating point operations.

Algorithm 2.1: Gaxpy Cholesky [Golub13, Algorithm 4.2.1]

input: $n \times n$ symmetric positive definite matrix A
output: For all $i \geq j$, $L_{i,j}$ overwrites $A_{i,j}$

1 **for** $j = 1$ *to* n **do**
2 **if** *j>1* **then**
3 $A_{j:n,j} = A_{j:n,j} - A_{j:n,1:j-1} \cdot A^T_{j,1:j-1}$
4 $A_{j:n,j} = A_{j:n,j} / \sqrt{A_{j,j}}$

Again considering the example matrix B given in eq. (2.3), the Cholesky factorization of matrix B is

$$\begin{pmatrix} 1 & -2 & 3 \\ -2 & 13 & -3 \\ 3 & -3 & 19 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ -2 & 3 & 0 \\ 3 & 1 & 3 \end{pmatrix} \begin{pmatrix} 1 & -2 & 3 \\ 0 & 3 & 1 \\ 0 & 0 & 3 \end{pmatrix} \tag{2.4}$$

The Sherman-Morrison-Woodbury-Matrix identity states that the inverse of a rank-k correction of a matrix can be computed by a rank-k correction of the inverse matrix [Golub13, p.65]. More precisely, the Sherman-Morrison-Woodbury-Matrix identity is

$$(A + UV^T)^{-1} = A^{-1} - A^{-1}U(I_k + V^TA^{-1}U)^{-1}V^TA^{-1}, \tag{2.5}$$

where $A \in \mathbb{R}^{n\times n}$ and $U, V \in \mathbb{R}^{n\times k}$.

The Kronecker product is an operation, which can be applied to two matrices of arbitrary size and results in another matrix.

Definition 2.7. For a matrix $A \in \mathbb{R}^{m\times n}$ and a matrix $B \in \mathbb{R}^{p\times q}$, the Kronecker product $A \otimes B$ of A and B is defined as the $mp \times nq$ block matrix

$$A \otimes B = \begin{pmatrix} a_{1,1}B & \dots & a_{1,n}B \\ \vdots & \ddots & \vdots \\ a_{m,1}B & \dots & a_{m,n}B \end{pmatrix} \in \mathbb{R}^{mp\times nq}. \tag{2.6}$$

For example, the Kronecker product of two 2×2 matrices is

$$\begin{aligned} \begin{pmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \end{pmatrix} \otimes \begin{pmatrix} b_{1,1} & b_{1,2} \\ b_{2,1} & b_{2,2} \end{pmatrix} &= \begin{pmatrix} a_{1,1}\begin{pmatrix} b_{1,1} & b_{1,2} \\ b_{2,1} & b_{2,2} \end{pmatrix} & a_{1,2}\begin{pmatrix} b_{1,1} & b_{1,2} \\ b_{2,1} & b_{2,2} \end{pmatrix} \\ a_{2,1}\begin{pmatrix} b_{1,1} & b_{1,2} \\ b_{2,1} & b_{2,2} \end{pmatrix} & a_{2,2}\begin{pmatrix} b_{1,1} & b_{1,2} \\ b_{2,1} & b_{2,2} \end{pmatrix} \end{pmatrix} \\ &= \begin{pmatrix} a_{1,1}b_{1,1} & a_{1,1}b_{1,2} & a_{1,2}b_{1,1} & a_{1,2}b_{1,2} \\ a_{1,1}b_{2,1} & a_{1,1}b_{2,2} & a_{1,2}b_{2,1} & a_{1,2}b_{2,2} \\ a_{2,1}b_{1,1} & a_{2,1}b_{1,2} & a_{2,2}b_{1,1} & a_{2,2}b_{1,2} \\ a_{2,1}b_{2,1} & a_{2,1}b_{2,2} & a_{2,2}b_{2,1} & a_{2,2}b_{2,2} \end{pmatrix}. \end{aligned} \tag{2.7}$$

If u and v are vectors then

$$uv^T = u \otimes v^T = v^T \otimes u. \tag{2.8}$$

Important properties of the Kronecker product are [Golub13, p.27]

$$(B \otimes C)^T = B^T \otimes C^T \tag{2.9}$$

$$(B \otimes C)(D \otimes F) = BD \otimes CF \tag{2.10}$$

$$(B \otimes C)^{-1} = B^{-1} \otimes C^{-1} \tag{2.11}$$

$$B \otimes (C \otimes D) = (B \otimes C) \otimes D, \tag{2.12}$$

where BD and CF must be defined for eq. (2.10) to make sense and the matrices B and C must be nonsingular in eq. (2.11).

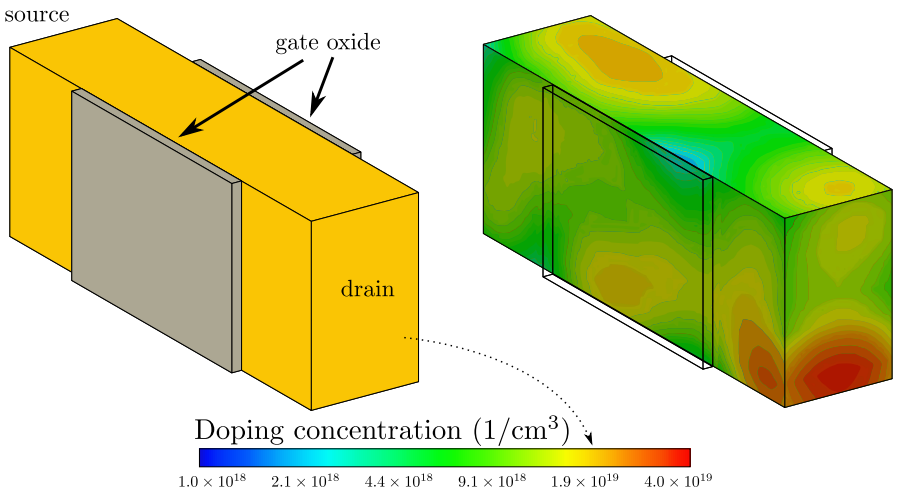
2.2 Sources of Variability

This section provides a summary of the major sources of variability, which can have a large impact on the gate and interconnect delays and ultimately on the delay test [Kenne01, Sriva05, Blaau08].

2.2.1 Process Variations

Process variations are the deviations of the manufactured circuit structure and parameters from the designed layout and expected parameters [Forza09]. In state-of-the-art

nanoscale CMOS circuits, even variations in the placement of a few atoms may significantly change the electrical properties of a transistor. Of particular importance are variations in the threshold voltage and the saturation-current of the transistors. For example, the average number of dopant atoms in the channel of a transistor in 32-nm process technology has dropped below 100 [Kuhn08] so even small variations in the number of dopants and the distribution of dopants across the channel, as shown in fig. 2.1, can have a large impact on the threshold voltage. This so-called random-dopant fluctuation is known to contribute around 60% of the total transistor threshold voltage variability in technology nodes below 90nm [Kuhn08, Aseno08, Yamag13]. Variations in the gate materials and the gate dielectric constitute another major source of threshold voltage and saturation-current variability [Li10].



▲ **Figure 2.1** — Model of a single-gate FinFET structure without random-dopant fluctuation (left) and with random-dopant fluctuation (right), adopted from [Leung12a]

In the latest technology nodes, line-edge roughness has become a major source of variability [Dietr12]. Line-edge roughness describes the noisy pattern in the line edges of each gate, which is caused by variations in the number of photons and the chemical composition of the photoresist during the lithographic exposure. In future technology nodes, line-edge roughness is expected to supplant random dopant fluctuation as the dominant source of variability [Kuhn09].

2.2.2 Environmental Variations

Environmental variations arise during the circuit operation due to variations in the environment that surrounds the chip [Sriva05]. Of particular importance for the gate delays are the temperature, the supply voltage and the switching activity. A drop in supply voltage lowers the saturation-current of the transistors, resulting in increased gate and interconnect delay. The actual voltage seen by a circuit component is the supply voltage minus the IR-drop, where the IR-drop refers to the voltage drop across the power distribution network due to the resistivity of the interconnects. High switching

activity causes large currents to flow in power and ground networks which can lead to a significant drop in the voltage seen by a transistor. Increasing the temperature has a similar negative impact on the circuit performance by increasing the interconnect resistance and the transistor switching delay.

2.2.3 Model Inadequacy and Numerical Errors

Every circuit model, no matter how detailed, is only an imperfect and idealized representation of a real physical circuit and it cannot perfectly describe the true electrical behaviour of a real circuit during delay testing. For example, the delay of a particular gate may be overestimated or underestimated by the model, depending on the operating conditions of the circuit and the transitions at the inputs of the gate.

The uncertainty introduced by model inadequacy may be reduced by using more detailed models at lower abstraction levels. But even if the model takes all important physical laws into account, it may not be possible to determine all model parameters with sufficient accuracy. Furthermore, the model may be too complex to evaluate exactly and require approximations which introduce numerical errors.

2.2.4 Other Sources of Variability

Some sources of variability change the electrical properties of gates and interconnects over time. Important examples are negative bias temperature instability and hot-carrier injection, which change the threshold voltage of transistors over time. Another important effect is electromigration, which increases the resistance of interconnects by reducing the wire width and therefore increases the transition propagation delay along the wire.

2.2.5 Impact on Important Electrical Parameters of Transistors

Physical parameter variations cause variations in the electrical parameters such as saturation current, threshold voltage and gate capacitance. The electrical parameter variations cause gate delay variations, which can then lead to delay test invalidation.

To study the impact of process variations on the electrical parameters of a FinFET transistor, the structure of a transistor can be modelled as a 3-dimensional grid and simulated with statistical technology computer-aided design tools. For example, the simulation model of a single-gate FinFET for the analysis of the effects of random dopant fluctuation is shown in Figure 2.1. Based on this approach, the impact of random dopant fluctuation and line-edge roughness on state-of-the-art FinFET transistors has been studied in [Leung12b] and [Leung12a]. The standard deviation of several electrical parameters, expressed as percentage of their nominal value, is presented in table 2.1.

▼ **Table 2.1 —** Trends in FinFET device variability caused by line-edge roughness (LER) and random-dopant fluctuation (RDF) [Leung12b, Leung12a]

	LER (1 nm amplitude)			**RDF** (10 nm fin height)		
Technology node (nm)	32 nm	21 nm	15 nm	32 nm	21 nm	15 nm
threshold voltage variability	5.3%	7.6%	10.8%	42%	57%	69%
saturation current variability	2.8%	3.8%	5.8%	19%	22%	23%

2.3 Classification of Process Variations

To analyse the impact of process variations on the delay test, it is beneficial to distinguish between different components of process variations [Sriva05, Blaau08, Forza09], which are shown in fig. 2.2. These components have a different impact on the circuit performance and will be explained in the following.

2.3.1 Systematic Variations

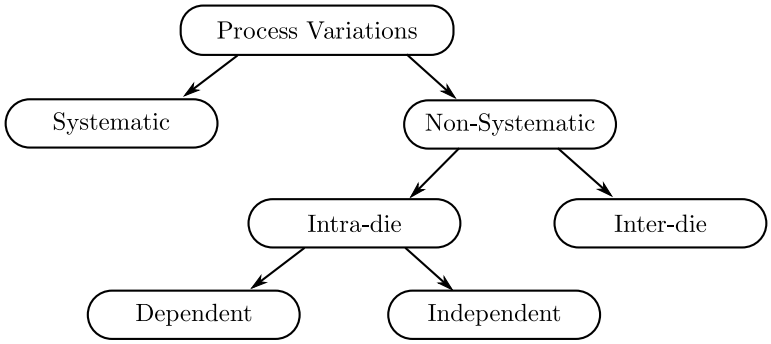
Systematic variations are presumed to be caused by a lack of knowledge or data [Kiure09]. For example, the imperfections of a photomask or the impact of optical proximity effects on the circuit layout might not be measurable with sufficient accuracy, or the circuit model used for analysis might neglect these effects altogether. Their influence on a particular circuit layout could also be predicted before circuit manufacturing based on detailed analysis of the layout and the manufacturing process. However, the necessary data for such a detailed analysis is often kept secret by a semiconductor foundry for commercial reasons.

2.3.2 Non-Systematic Variations

Non-systematic variations describe the truly unknown component of process variations, of which only statistical characteristics are available during test generation. The impact of non-systematic variations must be modelled by random variables with some known probability distributions and relationships. Major sources of non-systematic variations are for example random-dopant fluctuation and line-edge roughness. Depending on the spacial scales of process variations, non-systematic variations can be classified into inter-die and intra-die variations.

The *inter-die variations* are those non-systematic process variations, which have the same values across a die but different values from die-to-die, wafer-to-wafer and lot-to-lot. For example, variations in the lithography process like variations in the exposure time affect the gate length of all gates within the same die similarly.

Intra-die variations are non-systematic process variations that cause deviations between different parts of the same die. For example, random dopant fluctuation is widely considered to be a purely random phenomenon which occurs independently for each transistor. Therefore, each transistor in the design requires its own random variable to model intra-die variations, which hugely increases the complexity of statistical timing



▲ **Figure 2.2 —** Taxonomy of process variations [Blaau08]

analysis. Even for variations which show a systematic trend across the wafer, these trends might change over time in unpredictable ways. Therefore, this kind of systematic variation is usually also considered part of intra-die variations.

Intra-die variations are expected to dominate in future technology nodes, due to a shift to purely random and independent physical sources of process variations like random dopant fluctuation and line edge roughness [Li10, Agarw07].

2.4 Formal Modelling of Variability

Due to the limitations of the manufacturing process, each circuit has slightly different parameters. To analyse the impact of process variations on the delay test, it would be ideal to simulate the entire manufacturing process and the delay test. However, this kind of approach is prohibitively complex. Instead, a mathematical model, which can be validated with silicon measurements, can used to understand, describe, and quantify important aspects of the circuit's timing behaviour and to predict the effectiveness of delay tests considering variability.

The mathematical concepts described in this section form the basis for the modelling and the analysis of the circuits timing behaviour in presence of process variations, which is called *statistical timing analysis*. Statistical timing analysis employs probability theory and statistics to treat the variability in all critical circuit parameters in an explicit manner [Sriva05, Blaau08].

Unless cited otherwise, the following definitions and theorems are wherever possible quoted verbatim from: M. H. DeGroot and M. J. Schervish. *Probability and Statistics*. Pearson Education, fourth edition, 2012. ISBN 978-0-321-50046-5

2.4.1 Probability Space

The semiconductor manufacturing process of VLSI circuits is a *random experiment*, which is an experiment, which can produce different outcomes (circuits with different physical and electrical properties) even though it is repeated in the same manner every

time [Montg13]. The process variations introduced by the manufacturing process are described by a *probability space* $(\Theta, \mathcal{A}, \mathbb{P})$. The probability space consists of the sample space Θ, which is the set of all possible outcomes and a set of events $\mathcal{A}$, where each event $A \in \mathcal{A}$ is a set with one or more outcomes.

To describe the impact of process variations on the manufacturing of a particular VLSI circuit layout, the sample space Θ is defined as an infinite set of *circuit instance*s. Each circuit instance $\theta \in \Theta$ has unique deviations from the expected shape of the circuit layout due to process variations. The circuit instance with zero deviations is called *nominal circuit instance* and denoted by θ_0. Layout deviations which are of particular importance for the circuit timing behaviour are for example the transistor channel length and width, the oxide thickness and the channel dopant profile.

An event $A \in \mathcal{A}$ is a subset of circuit instances which, for example, all share the same physical defect. Other events of interest can be described by combinations of existing events by using basic set operations such as unions, intersections and complements.

The probability measure $\mathbb{P}$ assigns a probability to the occurrence of each event $A \in \mathcal{A}$.

Definition 2.8. A *probability measure*, or simply *probability*, on a sample space Θ is a function $\mathbb{P} : \mathcal{A} \to [0, 1]$, which satisfies the following three conditions

$$\mathbb{P}(\Theta) = 1, \tag{2.13}$$

$$\mathbb{P}(A) \geq 0 \text{ for all } A \in \mathcal{A} \tag{2.14}$$

and for every infinite sequence of disjoint events $A_1, A_2, \ldots$,

$$\mathbb{P}\left(\bigcup_{i=1}^{\infty} A_i\right) = \sum_{i=1}^{\infty} \mathbb{P}(A_i). \tag{2.15}$$

The expression $\mathbb{P}(A)$ represents how likely it is that the experiment's actual outcome is a member of A.

Definition 2.9 (Conditional Probability)**.** Suppose that we learn that an event B has occurred and that we wish to compute the probability of another event A taking into account that we know that B has occurred. The new probability of A is called the *conditional probability* of the event A given that the event B has occurred and is denoted $\mathbb{P}(A \,|\, B)$. If $\mathbb{P}(B) > 0$, we compute this probability as

$$\mathbb{P}(A \,|\, B) = \frac{\mathbb{P}(A \cap B)}{\mathbb{P}(B)}. \tag{2.16}$$

The conditional probability $\mathbb{P}(A \,|\, B)$ is undefined if $\mathbb{P}(B) = 0$.

Definition 2.10 (partition of sample space)**.** Let $(\Theta, \mathcal{A}, \mathbb{P})$ be a probability space. A *partition* of Θ is a collection of events $B_1 \in \mathcal{A}, \ldots, B_k \in \mathcal{A}$ such that $B_1, \ldots, B_k$ are disjoint and $\bigcup_{i=1}^{k} B_i = \Theta$.

Theorem 2.11 (law of total probability). *Suppose that the events $B_1, \ldots, B_k$ form a partition of the sample space Θ and $\mathbb{P}\left(B_j\right) > 0$ for $j = 1, \ldots, k$. Then, for every event A in $\mathcal{A}$,*

$$\mathbb{P}(A) = \sum_{j=1}^{k} \mathbb{P}\left(B_j\right)\mathbb{P}\left(A \,\middle|\, B_j\right). \tag{2.17}$$

The proof can be found in [DeGro12, Theorem 2.1.4].

Definition 2.12 (independence). The k events $A_1 \in \mathcal{A}, \ldots, A_k \in \mathcal{A}$ are called *independent* (or *mutually independent*) if, for every subset $A_{i_1}, \ldots, A_{i_j}$ of j of these events ($j = 2, \ldots, k$),

$$\mathbb{P}\left(A_{i_1} \cap \ldots \cap A_{i_j}\right) = \mathbb{P}\left(A_{i_1}\right) \cdots \mathbb{P}\left(A_{i_j}\right). \tag{2.18}$$

2.4.2 Random Variables

A random variable can be thought of as a quantity that can be measured in a random experiment. The quantity is determined once an experiment has been performed, in other words, an outcome $\theta \in \Theta$ (a circuit instance) has been chosen.

Formally, a random variable X is a function that assigns a real number to an outcome of an experiment

$$X : \theta \in \Theta \rightarrow \mathbb{R}. \tag{2.19}$$

For compactness of notation, θ is omitted and X is written to denote $X(\theta)$, wherever it is safe to to do.

Definition 2.13 (Continues Random Variable). We say that a random variable X has a *continuous distribution* or that X is a *continuous random variable* if there exists a non-negative function f, defined on the real line, such that for every interval of real numbers (bounded or unbounded), the probability that X takes a value in the interval is the integral of f over the interval.

This work is mostly concerned with continues random variables so that in the following, "random variable" is used to refer to a continues random variable, wherever it is safe to do so.

Random variables can be used to define events. For example, if $X(\theta)$ denotes the delay of a particular path, then

$$\{\theta \in \Theta : X(\theta) \leq T_{clk}\} \tag{2.20}$$

represents the event (subset of all circuit instances) in which the delay X of the path is less or equal the clock cycle time T_{clk}. Then the probability of this event is

$$\mathbb{P}(\{\theta \in \Theta : X(\theta) \leq T_{clk}\}). \tag{2.21}$$

Instead of the above expression, the shorthand notation $\mathbb{P}(X \leq T_{clk})$ is used throughout this work.

The distribution of the random variable X can be described by a probability density function, which assigns probabilities to intervals.

Definition 2.14 (Probability Density Function (PDF))**.** If X has a continuous distribution, then the function f described in definition 2.13 is called the *probability density function* (PDF) of X.

The distribution of the random variable X can also be described by the cumulative distribution function.

Definition 2.15 (Cumulative Distribution Function (CDF))**.** The *distribution function* or *cumulative distribution function* (CDF) F of a random variable X is the function

$$F(x) = \mathbb{P}(X \leq x) \quad \text{for } -\infty < x < \infty. \tag{2.22}$$

It can be shown that the cumulative distribution function F has the following properties

- F is monotonically increasing, right-continuous, $0 \leq F \leq 1$,
- $\lim_{x \to +\infty} F(x) = 1$ and
- $\lim_{x \to -\infty} F(x) = 0$.

Theorem 2.16. *Let X have a continuous distribution, and let $f(x)$ and $F(x)$ denote its PDF and the CDF, respectively. Then F is continuous at every x,*

$$F(x) = \int_{-\infty}^{x} f(t)dt, \tag{2.23}$$

and

$$\frac{dF(x)}{dx} = f(x) \tag{2.24}$$

at all x such that f is continuous.

The proof can be found in [DeGro12, Theorem 3.3.5].

The term critical path is redefined in a statistical setting as a path whose probability of exceeding a deterministic critical delay for the circuit is higher than a certain threshold [Sriva05]. This is formalized in the following definition.

Definition 2.17. A sensitizable logical path with delay X is called *critical path* if and only if

$$\mathbb{P}(X > T_{clk}) > p_{cri}, \tag{2.25}$$

where T_{clk} denotes the clock cycle time and p_{cri} is a user defined threshold.

Moments

Important properties of random variables can often be described by specific quantitative measures, called moments, instead of the distribution function. For example, the distribution of a random variable X is often summarized by two numbers: the *mean* as a measure of the centre and the *variance* as a measure of the dispersion or spread.

Definition 2.18. Let X denote a random variable with probability density function f. Suppose that at least one of the following integrals is finite:

$$\int_0^{+\infty} xf(x)dx, \qquad \int_{-\infty}^{0} xf(x)dx. \tag{2.26}$$

Then the *mean*, *expectation*, or *expected value* of X is said to exist and is defined to be

$$\mathbb{E}(X) = \int_{-\infty}^{+\infty} xf(x)\,dx. \tag{2.27}$$

If both of the integrals in eq. (2.26) are infinite, then $\mathbb{E}(X)$ does not exist.

Definition 2.19. [Variance/Standard Deviation] Let X be a random variable with finite mean $\mu = \mathbb{E}(X)$. The *variance* of X, denoted by $\mathrm{Var}(X)$, is defined as

$$\mathrm{Var}(X) = \mathbb{E}((X-\mu)^2). \tag{2.28}$$

If X has infinite mean or if the mean of X does not exist, it is said that $\mathrm{Var}(X)$ *does not exist*. The *standard deviation* of X is $\sqrt{\mathrm{Var}(X)}$ if the variance exists.

Higher order moments like $\mathbb{E}(X^2)$ could be computed from definition 2.18 by computing the PDF of the random variable $Y = X^2$. However, the expectation of X^2 can always be calculated directly using the following theorem.

Theorem 2.20. *Let X be a random variable, and let r be a real-valued function of a real variable. If X has a continuous distribution, then*

$$\mathbb{E}(r(X)) = \int_{-\infty}^{+\infty} r(x)f(x)\,dx, \tag{2.29}$$

if the mean exists.

The proof can be found in [DeGro12, Theorem 4.1.1].

Definition 2.21. Suppose that X is a random variable with $\mathbb{E}(X) = \mu$ well defined. For every positive integer k, the expectation $\mathbb{E}((X-\mu)^k)$ is called the kth *central moment* of X or the kth moment of X about the mean.

For example, the variance of X is the second central moment of X.

The mean and the variance do not uniquely identify a probability distribution. In fact, random variables of very different probability distributions can have the same mean and variance. This leads to a measure that can be used to quantify the lack of symmetry of the PDF.

Definition 2.22 (Skewness)**.** Let X be a random variable with mean $\mu = \mathbb{E}(X)$, standard deviation σ and finite third moment. The *skewness* of X, denoted by $\mathrm{Sk}(X)$, is defined as

$$\mathrm{Sk}(X) = \mathbb{E}((X-\mu)^3)/\sigma^3. \tag{2.30}$$

The reason for dividing by σ^3 is to make this measure independent of the variance of the distribution. In fact, $\mathrm{Var}((X-\mu)^3/\sigma^3) = 1$. The skewness is often used to quantify the deviation of a distribution of a random variable from a normal distribution.

Univariate Normal Distribution

The normal distribution is a very common continuous probability distribution.

Definition 2.23. A random variable X has the *univariate normal distribution with mean μ and variance σ^2* ($-\infty < \mu < \infty$ and $\sigma > 0$) if X has a continuous distribution with the following PDF:

$$f(x;\mu,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \quad \text{for } -\infty < x < \infty. \tag{2.31}$$

The proof that the above defined function is indeed a PDF can be found in [DeGro12, Theorem 5.6.1]. In the following, the convenient shorthand notation $X \sim \mathcal{N}(\mu,\sigma^2)$ is used to describe that X has a univariate normal distribution with parameters μ and σ.

Theorem 2.24. *The mean and variance of the distribution with PDF given by eq. (2.31) are μ and σ^2, respectively.*

The proof can be found in [DeGro12, Theorem 5.6.3].

The *univariate standard normal distribution* is a special case of the univariate normal distribution with $\mu = 0$ and $\sigma = 1$. In this work, the function ϕ with

$$\phi(x) := \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} \tag{2.32}$$

will be used to denote the probability density function and the function Φ with

$$\Phi(x) := \int_{-\infty}^{x} \phi(y)dy \tag{2.33}$$

will be used to denote the cumulative distribution function of the standard normal distribution.

Whenever a random experiment is replicated many times, the random variable that represents the average result over all replicates tends to be similar to a normal distribution. The underlying fundamental result is known as the central limit theorem, which makes the normal distribution remarkably useful.

Theorem 2.25 (Lindeberg-Feller Central Limit Theorem). *Suppose that* $\{X_i\}$, $i = 1, \ldots, n$ *is a sequence of independent random variables with finite means* μ_i *and finite positive variances* σ_i^2 *and* $\bar{X}_n = (1/n)\sum_{i=1}^{n} X_i$. *Let*

$$\bar{\mu}_n = \frac{1}{n}(\mu_1 + \mu_2 + \cdots + \mu_n) \quad \textit{and} \quad \bar{\sigma}_n^2 = \frac{1}{n}(\sigma_1^2 + \sigma_2^2 + \cdots + \sigma_n^2).$$

If no single term dominates the average variance $\bar{\sigma}_n^2$, *say*

$$\lim_{n\to\infty} \frac{1}{n\bar{\sigma}_n} \max_i(\sigma_i) = 0 \tag{2.34}$$

and if the average variance converges to a finite constant

$$\bar{\sigma}^2 = \lim_{n\to\infty} \bar{\sigma}_n^2, \tag{2.35}$$

then

$$\sqrt{n}(\bar{X}_n - \bar{\mu}_n) \xrightarrow{d} \mathcal{N}(0, \bar{\sigma}^2). \tag{2.36}$$

Theorem 2.25 is a simplified description of the central limit theorem, which is quoted from [Green12, Theorem D.19]. The complete theorem, the Lindeberg condition and an elaborate proof can for example be found in [Resni14, Theorem 9.8.1].

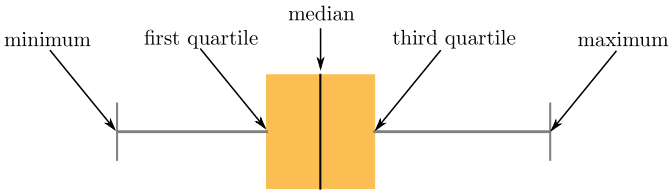
Descriptive Statistics: Quartiles and Box Plot

In some situations the distribution of a random variable is unknown and only a sample of measurements is available, for example of the oscillation frequency of a particular ring oscillator on a large number of circuits. These measured values must be summarized for further analysis [Montg13].

The *sample median* describes the central tendency of the data by dividing the data into two parts of equal size, half below and half above the median. In case the number of measurements is even, then the median is defined to be halfway between the two central measured values. The ordered set of measured values can be further divided into four equal sized parts, where the division points are called *quartiles*. The first quartile has approximately 25% of the measured values below and 75% of the measured values above it. The second quartile is the median and the third quartile has approximately 75% of the measured values below its value.

A *box plot* describes several important features of the data, such as center, spread and skewness, in a compact diagram. The box plot (box-and whisker plot) shows the minimum, maximum and the three quartiles in a rectangular box. The left/lower edge of the box is the first quartile and the right/upper edge is the third quartile. The black line inside the box marks the second quartile, which is the median of the data. The left/lower whisker is the line between the minimum and the first quartile. Likewise, the right/upper whisker is the line between the third quartile and the maximum.

A typical example is shown in fig. 2.3. It shows that the distribution is fairly symmetric because the lengths of the left and right whisker are about the same and the black line (median) roughly divides the box into two equal parts.



▲ **Figure 2.3 —** Description of a box plot

2.4.3 Canonical Delay Model

In [Zhong11], the authors show for a 65nm manufacturing technology that the abstraction of process variations by delay variations is possible and reasonable to limit the complexity of the statistical timing analysis.

Following the classification of non-systematic variability given in section 2.3.2, a process parameter Z is a random variable and can be written as [Sriva05]

$$Z = z + \check{Z} + \hat{Z}, \tag{2.37}$$

where z denotes the nominal value of the process parameter, $\check{Z}$ denotes a random variable representing the inter-die variation and $\hat{Z}$ denotes a random variable representing the intra-die variation. The random variables $\check{Z}$ and $\hat{Z}$ are usually assumed to have a normal distribution with zero mean.

In statistical timing analysis, each delay value of a gate is a function of random variables and therefore itself a random variable. According to [Viswe04], a delay value U can be expressed as a linear function

$$U = u + \sum_{i=1}^{k} a_i Z_i + a_{k+1}\tilde{Z} \tag{2.38}$$

of k process parameters $Z_1, \ldots, Z_k$, where u denotes the mean delay, the coefficients $a_1, \ldots, a_k$ represent the sensitivity of the delay to the process parameters $Z_1, \ldots, Z_k$ and a_{k+1} is the sensitivity of the delay to the residual variability $\tilde{Z}$, which is also a random variable with univariate normal distribution and mean zero.

More details about the circuit and gate modelling can be found in sections 2.5 and 7.2.

2.4.4 Random Vectors

The delays of n paths in a circuit, described by n random variables, are usually not independent due to structural and spacial correlations. To describe the relationship between the path delays, the random variables must be grouped into an n-dimensional *random vector* $\boldsymbol{X}$, which is a column vector of n random variables on the same probability space. For any positive integers $1 \leq i_1 < i_2 < \ldots i_r \leq n$ with $1 \leq r \leq n$ the random vector

$$(X_{i_1}, X_{i_2}, \ldots, X_{i_r})^T \tag{2.39}$$

is called an r-dimensional *subvector* of the random vector $\boldsymbol{X}$.

The distribution of a random vector is described by extending the concept of a probability density function for a single random variable to n-dimensional spaces.

Definition 2.26 (Joint CDF)**.** The *joint cumulative distribution function (CDF)* of n random variables $X_1, \ldots, X_n$ is the function F whose value at every point $(x_1, \ldots, x_n)^T \in \mathbb{R}^n$ is specified by the relation

$$F(x_1, \ldots, x_n) = \mathbb{P}(X_1 \leq x_1, \ldots, X_n \leq x_n). \tag{2.40}$$

Definition 2.27 (Joint PDF)**.** It is said that n random variables $X_1, \ldots, X_n$ have a continuous joint distribution if there is a nonnegative function f defined on $\mathbb{R}^n$ such that for every subset $C \subset \mathbb{R}^n$,

$$\mathbb{P}((X_1, \ldots, X_n) \in C) = \int_C f(x_1, \ldots, x_n)\, d\boldsymbol{x} \tag{2.41}$$

with $\boldsymbol{x} = (x_1, \ldots, x_n)$ if the integral exists. The function f is then called the *joint probability density function (PDF)* of $X_1, \ldots, X_n$.

For the given random vector $\boldsymbol{X} = (X_1, X_2)^T$ of two path delays X_1 and X_2 with joint PDF $f_{\boldsymbol{X}}$, the probability that the observed path delays are within a particular region R of a two-dimensional space is given by the double integral

$$\mathbb{P}((X_1, X_2) \in R) = \iint_R f_{\boldsymbol{X}}(x_1, x_2)\, dx_1 dx_2 \tag{2.42}$$

of $f_{\boldsymbol{X}}$ over the region R. The above expression can be interpreted as the volume under $f_{\boldsymbol{X}}$ in the region R. For example, the probability that none of the paths has a path delay fault is

$$\mathbb{P}(X_1 \leq T_{clk}, X_2 \leq T_{clk}) = \int_{-\infty}^{T_{clk}} \int_{-\infty}^{T_{clk}} f_{\boldsymbol{X}}(x_1, x_2)\, dx_1 dx_2, \tag{2.43}$$

where T_{clk} denotes the clock cycle time.

Theorem 2.28. *Let X and Y have a joint CDF F. The CDF F_1 of just the single random variable X can be derived from the joint CDF F as $F_1(x) = lim_{y\to\infty} F(x, y)$. Similarly, the CDF F_2 of Y equals $F_2(y) = lim_{x\to\infty} F(x, y)$, for $0 < y < \infty$.*

The proof can be found in [DeGro12, Theorem 3.4.5].

Definition 2.29 (Marginal CDF/PDF)**.** Suppose that X and Y have a joint distribution. The CDF of X derived by theorem 2.28 is called the *marginal CDF* of X. Similarly, the PDF of X associated with the marginal CDF of X is called *marginal PDF* of X.

Theorem 2.30. *If X and Y have a continuous joint distribution with joint PDF f, then the marginal PDF f_1 of X is*

$$f_1(x) = \int_{-\infty}^{+\infty} f(x,y)\,dy \qquad \text{for } -\infty < x < \infty. \tag{2.44}$$

Similarly, the marginal PDF f_2 of Y is

$$f_2(y) = \int_{-\infty}^{+\infty} f(x,y)\,dx \qquad \text{for } -\infty < y < \infty. \tag{2.45}$$

The proof can be found in [DeGro12, Theorem 3.5.2].

Definition 2.31 (Conditional PDF)**.** Let X and Y have a continuous joint distribution with joint PDF f and respective marginals f_1 and f_2. Let y be a value such that $f_2(y) > 0$. Then the conditional PDF g_1 of X given that $Y = y$ is defined as follows:

$$g_1(x|y) = \frac{f(x,y)}{f_2(y)} \text{ for } -\infty < x < +\infty. \tag{2.46}$$

Moments

The concept of moments of a single random variable can be extended to random vectors. The mean (expectation) of the n-dimensional random vector $\boldsymbol{X} = (X_1, \ldots, X_n)^T$ is the vector of the expected values of all random variables

$$\mathbb{E}(\boldsymbol{X}) = (\mathbb{E}(X_1), \ldots, \mathbb{E}(X_n))^T. \tag{2.47}$$

A common measure for the relationship between two random variables is the covariance. To formally define the covariance, it is necessary to introduce the expected value of a function of the two random variables.

Theorem 2.32. *Suppose that $X_1, \ldots, X_n$ are random variables with the joint PDF f. Let r be a real-valued function of n real variables, and suppose that $Y = r(X_1, \ldots, X_n)$. Then $\mathbb{E}(Y)$ can be determined directly from the relation*

$$\mathbb{E}(Y) = \int_{\mathbb{R}^n} r(x_1, \ldots, x_n) f(x_1, \ldots, x_n)\,d\boldsymbol{x} \tag{2.48}$$

with $\boldsymbol{x} = (x_1, \ldots, x_n)^T$, if the mean exists.

The proof can be found in [DeGro12, Theorem 4.1.2].

Definition 2.33 (Covariance)**.** Let X and Y be random variables with finite means μ_X and μ_Y, respectively. The covariance of X and Y, which is denoted by $\mathrm{Cov}(X,Y)$, is defined as

$$\mathrm{Cov}(X,Y) = \mathbb{E}((X - \mu_X)(Y - \mu_Y)), \tag{2.49}$$

if the expectation in eq. (2.49) exists.

The sign of the covariance shows the tendency of linear relationship between X_1 and X_2. However, a covariance of zero does not imply independence because a non-linear relationship might exist between the random variables. A well-known example is the relationship between X and $Y = X^2$, which are clearly not independent. However, if $\mathbb{E}(X) = 0$ and $\mathbb{E}(X^3) = 0$, then $\text{Cov}(X, Y) = \mathbb{E}(XY) - \mathbb{E}(X)\mathbb{E}(Y) = \mathbb{E}(X^3) = 0$.

Theorem 2.34. *For all random variables X and Y such that $Var(X) < \infty$ and $Var(Y) < \infty$,*

$$Cov(X, Y) = \mathbb{E}(XY) - \mathbb{E}(X)\mathbb{E}(Y). \tag{2.50}$$

The proof can be found in [DeGro12, Theorem 4.6.1].

Interpreting the magnitude of the covariance $\text{Cov}(X, Y)$ is difficult because it depends on the variances of X and Y. A normalized version of the covariance is the *correlation*, which is defined as follows.

Definition 2.35 (Correlation)**.** Let X and Y be random variables with finite variances σ_X^2 and σ_Y^2, respectively. Then the correlation of X and Y, which is denoted by $\rho(X, Y)$, is defined as follows:

$$\rho(X, Y) = \frac{\text{Cov}(X, Y)}{\sigma_X \sigma_Y}. \tag{2.51}$$

It can be shown that $-1 \leq \rho(X, Y) \leq +1$ [DeGro12, Theorem 4.6.3].

Definition 2.36 (Mean Vector/Covariance Matrix)**.** If $\boldsymbol{Y}$ is a n-dimensional random vector, then the vector $\mathbb{E}(\boldsymbol{Y}) = (\mathbb{E}(Y_1), \ldots, \mathbb{E}(Y_n))^T$ is called the *mean vector* of $\boldsymbol{Y}$. The *covariance matrix* $\text{Var}(\boldsymbol{Y})$ of $\boldsymbol{Y}$ is defined to be the $n \times n$ matrix such that, for $i = 1, \ldots, n$ and $j = 1, \ldots, n$, the element in the ith row and jth column is $\text{Cov}(Y_i, Y_j)$.

For any affine transformation of the random vector of the form $\boldsymbol{Y} = A\boldsymbol{X} + \boldsymbol{b}$ with $A \in \mathbb{R}^{k \times n}$ of rank k and $\boldsymbol{b} \in \mathbb{R}^k$, the mean vectors and the covariance matrices of $\boldsymbol{X}$ and $\boldsymbol{Y}$ are related as follows.

Theorem 2.37. *Let $\boldsymbol{X}$ denote a n-dimensional random vector with mean vector $\boldsymbol{\mu_X} = \mathbb{E}(\boldsymbol{X})$ and covariance matrix $\Sigma_{\boldsymbol{X}} = Var(\boldsymbol{X})$. If $A \in \mathbb{R}^{k \times n}$ has rank k and $\boldsymbol{b} \in \mathbb{R}^k$, then the k-dimensional random vector*

$$\boldsymbol{Y} = A\boldsymbol{X} + \boldsymbol{b} \tag{2.52}$$

has a mean vector $\mu_{\boldsymbol{Y}} = \mathbb{E}(\boldsymbol{Y})$ and covariance matrix $\Sigma_{\boldsymbol{Y}} = Var(\boldsymbol{Y})$ given by

$$\boldsymbol{\mu_Y} = A\boldsymbol{\mu_X} + \boldsymbol{b} \tag{2.53}$$

$$\Sigma_{\boldsymbol{Y}} = A\Sigma_{\boldsymbol{X}}A^T. \tag{2.54}$$

The proof can be found in [DeGro12, Theorem 11.5.2]. The above theorem implies that $\text{Var}(\boldsymbol{X} - \boldsymbol{c}) = \text{Var}(\boldsymbol{X})$ for any real-valued vector $\boldsymbol{c} \in \mathbb{R}^n$.

Theorem 2.38. *Correlation matrices and covariance matrices are symmetric and positive definite.*

The proof can be found in [Mille12, Theorem 6.1].

Using theorem 2.32, the concept of covariance can be generalized to higher order joint moments, which will be frequently used in chapter 6.

Definition 2.39. Let X_1 and X_2 be random variables with finite means μ_{X_1} and μ_{X_2}, respectively. Then $\mathbb{E}(X_1^m X_2^n)$ is called the (m,n)th *joint moment* and

$$\mathbb{E}((X_1 - \mu_{X_1})^m (X_2 - \mu_{X_2})^n) \tag{2.55}$$

is called the (m,n)th *joint central moment* of X_1 and X_2.

Let $f_{\boldsymbol{X}}$ denote the joint PDF of X_1 and X_2, then from theorem 2.32 it follows that

$$\mathbb{E}(X_1^m X_2^n) = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} x_1^m x_2^n f_{\boldsymbol{X}}(x_1, x_2)\, dx_1 dx_2 \tag{2.56}$$

and

$$\mathbb{E}((X_1 - \mu_{X_1})^m (X_2 - \mu_{X_2})^n) = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} (x_1 - \mu_{X_1})^m (x_2 - \mu_{X_2})^n f_{\boldsymbol{X}}(x_1, x_2)\, dx_1 dx_2. \tag{2.57}$$

Definition 2.40. The *third multivariate cumulant*, denoted by κ_3, of a random vector $\boldsymbol{X}$ is

$$\kappa_3(\boldsymbol{X}) = \mathbb{E}((\boldsymbol{X} - \mathbb{E}(\boldsymbol{X})) \otimes (\boldsymbol{X} - \mathbb{E}(\boldsymbol{X})) \otimes (\boldsymbol{X} - \mathbb{E}(\boldsymbol{X}))^T), \tag{2.58}$$

where $\otimes$ denotes the Kronecker product.

Important properties of $\kappa_3(\boldsymbol{X})$ can be found in chapter 6 and in [Franc10, Luca15].

Definition 2.41 (Conditional Expectation)**.** Let X and Y be random variables such that the mean of Y exists and is finite. If Y has a continuous conditional distribution given $X = x$ with conditional PDF $g_2(y|x)$, then

$$\mathbb{E}(Y\,|x) = \int_{-\infty}^{+\infty} y g_2(y|x) dy. \tag{2.59}$$

is called the *conditional expectation* (or *conditional mean*) of Y given $X = x$.

Definition 2.42 (Conditional Means of Random Variables)**.** Let $h(x) = \mathbb{E}(Y\,|x)$ denote the conditional mean of Y given $X = x$, then the conditional mean $\mathbb{E}(Y\,|X)$ of Y given X is $\mathbb{E}(Y\,|X) = h(X)$.

Clearly, $\mathbb{E}(Y\,|X)$ is a function of the random variable X and therefore itself a random variable whose value is $\mathbb{E}(Y\,|x)$ when $X = x$. An important special case is given by the following theorem.

Theorem 2.43. *Let $(\Theta, \mathcal{A}, \mathbb{P})$ be a probability space and $B \in \mathcal{A}$ be an event with $\mathbb{P}(B) > 0$. Furthermore, let $X(\theta) := \mathbb{1}_B(\theta)$ and $\mathbb{E}(Y(\theta)\,|B) := \mathbb{E}(Y(\theta)\,|X(\theta) = 1)$, then*

$$\mathbb{E}(Y(\theta)\,|B) = \frac{\mathbb{E}(Y(\theta)\mathbb{1}_B(\theta))}{\mathbb{P}(B)}, \tag{2.60}$$

where $\mathbb{1}$ denotes the indicator function.

The proof can be found in [Ash00, p.212].

Bivariate Normal Distribution

Theorem 2.44. *Suppose that Z_1 and Z_2 are independent random variables, each of which has the standard normal distribution. Let $\mu_1, \mu_2, \sigma_1, \sigma_2$, and ρ be constants such that $-\infty < \mu_i < \infty$, $\sigma_i > 0$ for $i = 1, 2$ and $-1 < \rho < 1$. If the random variables X_1 and X_2 are defined as*

$$X_1 = \sigma_1 Z_1 + \mu_1 \tag{2.61}$$

$$X_2 = \sigma_2 \left(\rho Z_1 + (1-\rho^2)^{1/2} Z_2\right) + \mu_2, \tag{2.62}$$

then the joint PDF of X_1 and X_2 is

$$f(x_1, x_2) = \frac{1}{2\pi(1-\rho^2)^{1/2}\sigma_1\sigma_2} \exp\left(-\frac{1}{2(1-\rho^2)}\left(\left(\frac{x_1-\mu_1}{\sigma_1}\right)^2 - 2\rho\left(\frac{x_1-\mu_1}{\sigma_1}\right)\left(\frac{x_2-\mu_2}{\sigma_2}\right) + \left(\frac{x_2-\mu_2}{\sigma_2}\right)^2\right)\right). \tag{2.63}$$

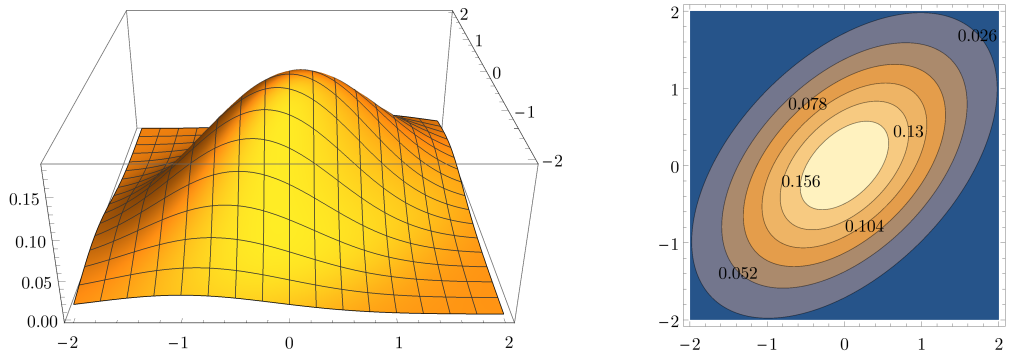
The proof can be found in [DeGro12, Theorem 5.10.1].

Definition 2.45 (Bivariate Normal Distributions)**.** If the joint PDF of two random variables X_1 and X_2 is of the form in eq. (2.63), it is said that the random vector $\boldsymbol{X} = (X_1, X_2)^T$ has a *bivariate normal distribution* with means μ_1 and μ_2, variances σ_1^2 and σ_2^2, and correlation ρ.

The probability density function of the bivariate normal distribution with $\boldsymbol{\mu} = (0,0)^T$ and

$$\Sigma = \begin{pmatrix} 1 & 0.5 \\ 0.5 & 1 \end{pmatrix}$$

is shown in fig. 2.4. A more general and flexible distribution will be introduced in section 6.1, whose PDF is shown in fig. 6.2 for the same mean vector $\boldsymbol{\mu}$ and covariance matrix Σ.



▲ **Figure 2.4 —** Probability density function of the bivariate normal distribution with $\boldsymbol{\mu} = (0,0)^T$ and $\Sigma = \left(\begin{smallmatrix} 1 & 0.5 \\ 0.5 & 1 \end{smallmatrix}\right)$

The *bivariate standard normal distribution* is a special case of the bivariate normal distribution with $\mu_1 = \mu_2 = 0$ and $\sigma_1 = \sigma_2 = 1$. In this work, the function ϕ_2 with

$$\phi_2(x_1, x_2; \rho) := \frac{1}{2\pi(1-\rho^2)^{1/2}} \exp\left(-\frac{1}{2(1-\rho^2)}\left(x_1^2 - 2\rho x_1 x_2 + x_2^2\right)\right) \tag{2.64}$$

will be used to denote the probability density function and the function Φ_2 with

$$\Phi_2(x_1, x_2; \rho) := \int_{-\infty}^{x_1}\int_{-\infty}^{x_2} \phi_2(y_1, y_2; \rho)\, dy_1 dy_2 \tag{2.65}$$

will be used to denote the cumulative distribution function of the standard bivariate normal distribution.

Multivariate Normal Distribution

The multivariate normal distribution is a generalization of the bivariate normal distribution to random vectors of arbitrary dimension [Czado11].

Definition 2.46. A n-dimensional random vector $\boldsymbol{X}$ is said to have a *n-dimensional normal distribution*, if there exits an $\boldsymbol{\mu} \in \mathbb{R}^n$ and an $L \in \mathbb{R}^{n\times n}$ with rank $(L) = n$, such that

$$\boldsymbol{X} = L\boldsymbol{Z} + \boldsymbol{\mu}, \tag{2.66}$$

where $\boldsymbol{Z} = (Z_1, ..., Z_n)^T$ is a random vector of independent random variables $Z_1, \ldots, Z_n$ and $Z_i \sim \mathcal{N}(0,1)$ for all $i \in \mathbb{N}$ with $1 \leq i \leq n$.

In this case, $\boldsymbol{X}$ is called *normal random vector* and it is written $\boldsymbol{X} \sim \mathcal{N}_n(\boldsymbol{\mu}, \Sigma)$, where $\Sigma = LL^T$.

From eq. (2.66) and theorems 2.24 and 2.37 it follows that $\mathbb{E}(\boldsymbol{X}) = \boldsymbol{\mu}$, $\mathrm{Var}(\boldsymbol{X}) = \Sigma$ and that the family of multivariate normal distributions is closed under affine transformations. This is formalized in the following theorem.

Theorem 2.47. *If* $\boldsymbol{X} \sim \mathcal{N}_n(\boldsymbol{\mu_X}, \Sigma_{\boldsymbol{X}})$, $\boldsymbol{b} \in \mathbb{R}^k$ *and* $A \in \mathbb{R}^{k \times n}$ *has rank* k*, then the random vector*

$$\boldsymbol{Y} = A\boldsymbol{X} + \boldsymbol{b} \tag{2.67}$$

has a k*-dimensional normal distribution* $\mathcal{N}_k(\boldsymbol{\mu_Y}, \Sigma_{\boldsymbol{Y}})$ *with parameters*

$$\boldsymbol{\mu_Y} = A\boldsymbol{\mu_X} + \boldsymbol{b} \tag{2.68}$$

$$\Sigma_{\boldsymbol{Y}} = A\Sigma_{\boldsymbol{X}}A^T. \tag{2.69}$$

The proof can be found in [Gut09, chapter 5, Theorem 3.1].

Theorem 2.48. *Let* $\det(\Sigma)$ *denote the determinant of the matrix* Σ*. If rank* $(\Sigma) = n$ *and the* n*-dimensional random vector* $\boldsymbol{X} \sim \mathcal{N}_n(\boldsymbol{\mu}, \Sigma)$ *has a normal distribution with parameters* $\boldsymbol{\mu}$ *and* Σ*, then the probability density function of* $\boldsymbol{X}$ *is*

$$\phi_n(\boldsymbol{x}; \boldsymbol{\mu}, \Sigma) = \frac{1}{\sqrt{(2\pi)^n \det(\Sigma)}} \exp\left(-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu})^T \Sigma^{-1} (\boldsymbol{x} - \boldsymbol{\mu})\right). \tag{2.70}$$

The proof can be found in [Gut09, chapter 5, Theorem 5.2].

If $\boldsymbol{X} \sim \mathcal{N}_n(\boldsymbol{\mu}, \Sigma)$ and $\mathrm{Cov}(X_i, X_j) = 0$ for any $1 \leq i < j \leq n$, then X_i and X_j are independent.

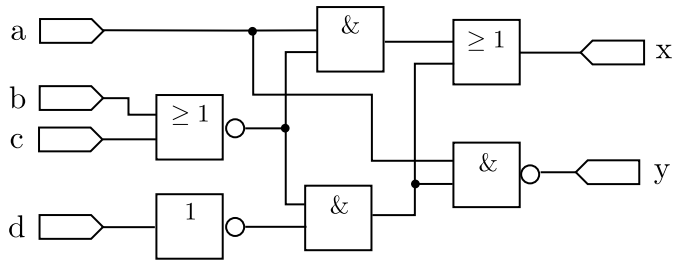
2.5 Circuit Modelling

The physical, structural and behaviour aspects of a VLSI circuit can be modelled at different abstraction levels [Gajsk83]. The algorithms presented in this work are based on the structural representation at gate level, which provides a suitable trade-off between the accuracy and the computational cost. The following subsections describe the structure and the behaviour of any randomly chosen circuit instance $\theta \in \Theta$ (see section 2.4.1).

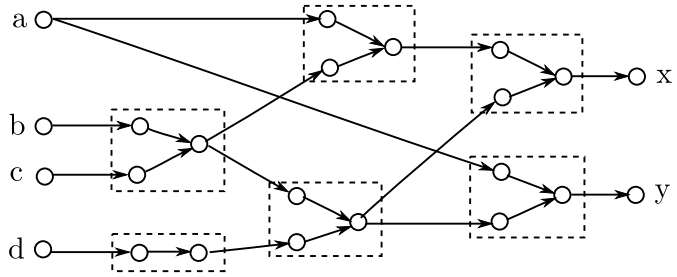
2.5.1 Structural Circuit Instance Modelling

The circuit structure at gate level is a representation of the circuit components (gates, standard cells, ports) and their interconnections. In the following and throughout this work, only the primitive logic gates BUF, INV, AND, OR, NAND, NOR and the complex cells XOR, XNOR with at most two inputs are considered.

To formally define the circuit instance model, the circuit netlist is transformed into a directed graph $G = (V, E)$ with a set of vertices V and a set of edges $E \subseteq V \times V$. Each vertex represents an input/output pin of a gate/complex cell or an input/output port of the netlist. An edge either corresponds to a pin-to-pin delay arc within a gate or represents an interconnection between gates/complex cells or input/output ports of the netlist. To account for asymmetric rising/falling and conditional path delays, a tuple of real valued delays is associated with each edge. The construction of the circuit instance graph is illustrated by the example in fig. 2.5.



(a) Representation as netlist



(b) Representation as graph

▲ **Figure 2.5 —** Structural representation of a circuit instance at gate level

A *structural path* is a sequence of gates $g_1, \ldots, g_n$ such that an input of g_1 is connected to a circuit input, an output of g_n is connected to an observable circuit output and the output of g_i is connected to an input of g_{i+1} for all $1 \leq i < n$. The input of g_{i+1}, which is connected to the output of g_i, is called *on-path input* and any other input of g_i is called *off-path input*.

2.5.2 Behavioural Modelling of Circuit Instance

The circuit instance graph defines the structure of the circuit instance at gate level. This subsection describes the propagation of transitions through the circuit instance during the application of a test vector-pair. Each node of the circuit instance is associated with a waveform, which consists of one or multiple transitions [Yalci97].

Definition 2.49 (transition)**.** A *transition* is an ordered pair $\tau = (v, t)$, where $v \in \{0, 1\}$ denotes the logic value after the transition and $t \in \mathbb{R}$ denotes the time at which the transition occurs.

Definition 2.50 (waveform)**.** A *waveform* w is a non-empty list of transitions

$$w = ((v_0, t_0), ..., (v_m, t_m)), \tag{2.71}$$

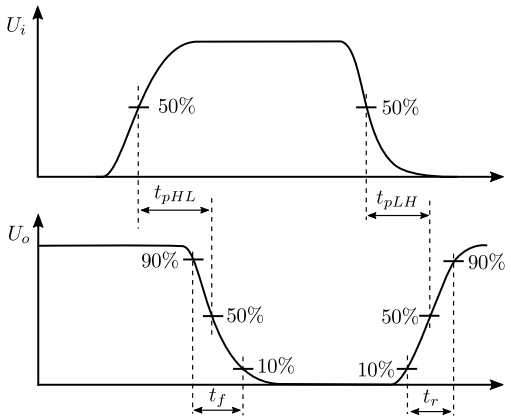
such that $t_{k-1} < t_k$ and $v_{k-1} \neq v_k$ for all $k \in \mathbb{N}$ with $0 < k \leq m$ and w contains at least the special transition (v_0, t_0) with $t_0 = -\infty$, which defines the initial value v_0 of the waveform.

The *initial value* of a waveform is defined as the logic value set by the first transition in the waveform. Likewise, the *final value* of a waveform is defined by the last transition in the waveform.

The above definition of a waveform is an abstraction of the voltage waveforms obtained by SPICE simulations at electrical level. This abstraction provides a suitable trade-off between the accuracy and the complexity of statistical timing analysis.

The definition of the gate model at this higher abstraction level requires a formal foundation, which will be motivated by the example given in fig. 1.4a. The figure shows that a transition at a gate input causes a transition at the gate output only after some delay. It is also visible, that the gate output voltage changes more rapidly in response to the rising transition at input 'b' than in response to the falling transition at 'a'.

This behaviour is described by this model at a higher abstraction level as follows. The *arrival time* t of a transition (v, t) at a circuit node describes the time at which the voltage crosses $U_{dd}/2$, where U_{dd} denotes the supply voltage. If a transition (v_1, t_1) at the input of a gate causes a transition (v_2, t_2) at the output of the gate, then the arrival time difference $t_2 - t_1$ is called *propagation delay*. This is illustrated in fig. 2.6 for an inverter (INV) with input 'i' and output 'o'. The figure shows that the propagation delay is measured for a falling and a rising transition at the gate output and that these delays are different in general. It also defines the rise and fall time of a transition as the time between neighbouring $0.1U_{dd}$ and $0.9U_{dd}$ voltage crossing times.



▲ **Figure 2.6 —** Definition of propagation delay and rise and fall times [Rabae03]

For primitive gates with two inputs, the gate delay also depends on which gate input is switching. This is due to the cell design but also because the slope of the gate input transitions depends on the circuit structure. For example, one of the gate inputs might be connected to a *fanout-gate*, that is, a gate whose output is connected to the inputs of multiple gate. As a consequence, the voltage at the gate input might change only very slowly, leading to higher propagation delay. Hence, in this model, each primitive gate with two inputs is associated with four propagation delay values, depending on which gate input is switching and if the transition at the gate output is a rising or falling transition.

For the complex cells XOR/XNOR, the propagation delay also depends on the logic values at the other gate inputs at the arrival time of the input transition. This is again due to the cell design and the circuit structure. Hence, each XOR/XNOR cell with two inputs is associated with eight delay values. From among these delay values, the appropriate value is selected for each input transition depending on the input at which the transition occurs, the state of the other gate input and the direction of the output transition created by the input transition.

A physical gate can only propagate a subset of the transitions at the gate inputs to the gate output. A transition (v, t) at one of the input nodes of a gate is propagated to the gate output node if and only if (v, t) satisfies the *dynamic sensitization condition* and the *inertial delay condition*. The conjunction of dynamic sensitization and inertial delay condition is called *propagation condition*.

The *dynamic sensitization condition* is always satisfied for a gate with only a single input and for XOR/XNOR complex cells. For AND/NAND and OR/NOR gates with two inputs, the opposite gate input must have the non-controlling value at the time t. A gate input is said to have a *controlling value* if it determines the gate output logic value irrespective of the other input values. The logic complement of the controlling value is called *non-controlling* value. For example, for an OR and NOR gate the controlling value is '1' and the non-controlling value is '0'.

The *inertial delay condition* captures the physical limitation that a real gate cannot produce arbitrary short glitches at the output, as shown in section 1.3.1. Without loss of generality, the following description assumes that the gate output has stabilized to logic '0' or logic '1' by the time the transition (v, t) occurs at one of the gate inputs. Furthermore, it is assumed that (v, t) satisfies the dynamic sensitization condition so that the gate can propagate this transition to the gate output after a particular propagation delay δ. This delay represents the time the gate requires to charge or discharge the capacitive load connected to the gate output. The transition can only appear at the gate output if no other transition (v', t'), which also satisfies the dynamic sensitization condition, occurs at the gate inputs during this time. More precisely, if there exists an input transition (v', t') that satisfies the dynamic sensitization condition of the gate and $t < t' < t + \delta$, then the inertial delay condition is violated by (v, t) and (v', t'). Otherwise, (v, t) satisfies the inertial delay condition of the gate and is propagated to the gate output after delay δ.

CHAPTER 3

State of the Art

This chapter discusses the state of the art in fundamental statistical timing analysis problems that arise in delay test applications under the impact of delay variability. Recent methods that can be used for the variation-aware evaluation of path delay fault tests are discussed in section 3.1. The latest advances in the evaluation of the small delay fault tests and their limitation are reviewed in section 3.2. The normal distribution based SUM and MAX-operations are presented in section 3.3.1. Finally, recent approaches to improve the efficiency of the statistical SUM and MAX-operations are summarized in section 3.3.3.

3.1 Evaluation of Path Delay Fault Tests

This section summarizes previous methods, which can be used to analyse the sensitization of a path by a set of test vector-pairs under the impact of delay variations.

3.1.1 Proof of Path Sensitization by Test Vector-Pair

A notable special case are methods, which treat gate delays as unknown quantities but don't explicitly consider the probability distributions of the gate delays. Although strictly speaking not statistical timing analysis, these methods are nevertheless important for path delay fault testing to proof that, under the timing-model assumptions, a particular path (1) *can be sensitized* or (2) *is always sensitized* by a test vector-pair under the impact of delay variations.

To formally describe path sensitization considering unknown gate delays, the authors of [Ishiu89] propose time-symbolic simulation which treats time as an algebraic expression. Each circuit node is associated with a set of waveforms, where a particular waveform

occurs depending on the conditions satisfied by the gate delays of the circuit. For a given test vector-pair, the condition that a path is sensitized is the disjunction of the conditions associated with those waveforms in which a transition was propagated along the path. A solution to any of those conditions yields a subset of the set of all circuit instances Θ in which the path is sensitized. Although this method works well for small subcircuits, it is infeasible to evaluate and represent all possible waveforms at each circuit node in large circuits.

An important special case, where the gate delays are unknown but bounded by some known upper and lower bounds, has been addressed in [Ishiu90]. The authors propose to use discrete gate delay values and to enumerate all possible delay values between the lower and upper bound for each gate. The path sensitization by a given test vector-pair is evaluated for all possible combinations of delay values. Although the runtime complexity of this approach is exponential in the number of gates, several optimizations such as representing all possible waveforms at the circuit nodes with shared binary decision diagrams, can be applied to efficiently analyse small subcircuits. However, this approach cannot be directly applied to large circuits due to its high runtime complexity.

3.1.2 Probability that Path is Sensitized by Test Vector-Pair

In [Bose07, Jayar13], the authors use the bounded delay model to evaluate the probability of path delay fault test invalidation. However, the bounded delay model implicitly assumes that the gate delays have a uniform distribution and that all gate and path delays are independent. These assumptions can lead to incorrect results. For example, a target path might only be sensitized if and only if all gates along the target path have highest possible delay and the gates of another path all have their lowest possible delays. In practice, this event is extremely unlikely or even impossible in case of reconvergence where at least two gates must lie on both paths. The identification of such impossible cases requires time consuming reconvergent fanout analysis [Chakr99], which must consider a large number of gate delay value combinations.

The authors of [Jung12] represent a waveform as a sequence of random variables, where each random variable denotes the arrival time of a rising or falling transition, respectively. To compute the waveform at the output of the gate, the normal distribution based SUM and MAX-operations are applied to the waveforms at the gate inputs using the corresponding gate delay values, which are also represented by random variables. The main advantage of this method is that it can approximate the delay distribution of the paths that are sensitized by a given test vector-pair while considering the impact of the delay variability on the sensitization of the paths. However, this method doesn't consider the inertial delay property of the gates and a non-trivial extension would be required to compute the probability that a particular path is sensitized by a given test vector-pair under the impact of delay variations.

The most general and flexible approach to analyse the sensitization of a path is a Monte Carlo simulation of the given test vector-pair considering delay variations [Xie09].

However, it is well known that this approach has a very high computational cost and might even necessitate the use of (possibly specialized) parallel hardware architectures for practical purposes [Schne15].

In [Tang14a], the authors propose a statistical simplified transistor model which can be simulated by solving random differential equations. A voltage waveform is represented by a piecewise linear function of nominal voltages, where each voltage point is associated with a sensitivity coefficient, which relates the variation of the voltage to the variation of the process parameters. Although this method could in theory be applied to compute the probability that a particular target path is sensitized by a given test vector-pair, a lot of simplifying assumptions are made in the algorithm and the experimental setup. In particular, intra-die variations are ignored during the experimental evaluation and only small circuits are considered.

3.2 Evaluation of Small Delay Fault Tests

This section summarizes previous methods, which can be used to analyse the detection of a given small delay fault by a given set of test vector-pairs under the impact of delay variations.

The statistical timing analysis of large circuits can be drastically simplified by neglecting all structural and spatial correlations [Yilma08, Wang09]. This is because the probability that several events occur simultaneously is equal to the product of the probabilities of the individual events under the assumption that all events are independent. However, this assumption is unrealistic and results in large errors, except for tiny delay variations in classical and mature technology nodes.

Some authors went even further and proposed measures for the "quality" or "effectiveness" of a test vector-pair, which are misleading even under the assumption of independent path delays. For example in [Peng10, Peng13], the authors compute the probability distributions of the path delays by a Monte-Carlo simulation. For each path that is sensitized by a test vector-pair, the probability that the delay of the path is longer than a predefined threshold, is computed. But then the authors compute the sum of these probabilities and use it as a measure of how likely a small delay fault will be detected by the test vector-pair. It can easily be seen that this is an unrealistic measure because it ignores structural and spacial correlations and the result is no longer a probability. For example, a test vector-pair which sensitizes two paths that have all but one gate in common might get a weight of e.g. $0.5 + 0.6 = 1.1$, while a test vector-pair which sensitizes ten slightly shorter but mutually independent paths might get a weight of $5 \cdot 0.02 + 5 \cdot 0.01 = 0.15$. Clearly, the latter test vector-pair is more likely to detect a randomly chosen small delay fault. Even if the path delays were mutually independent, the probability that the delay of at least one of these paths exceeds the predefined threshold would be computed by multiplication and not addition.

In the pioneering work [Ingel11], the authors introduce the "test robustness" metric for a test vector-pair, considering resistive bridging faults with uniformly distributed

resistance. Given a resistive bridging fault of random resistance that can be detected by at least one test vector-pair, the authors define the "test robustness" of a test vector-pair as the probability $\mathbb{P}(A)$ of the event $A \in \mathcal{A}$ in which the resistive bridging fault is detected by the test vector-pair under the impact of process variations. The authors propose to approximate this probability by computing the conditional probability $\mathbb{P}(A\,|\Theta_i)$ of detecting the resistive bridging fault in a subset of circuit instances $\Theta_i \subset \Theta$, for a large number of pairwise disjoint subsets $\Theta_1 \subset \Theta, \ldots, \Theta_n \subset \Theta$ of circuit instance. The authors then approximate the probability of detecting the fault with the test vector-pair ("test robustness") by

$$\mathbb{P}(A) \approx \frac{\sum_{i=1}^{n} \mathbb{P}(\Theta_i)\mathbb{P}(A\,|\Theta_i)}{\sum_{i=1}^{n} \mathbb{P}(\Theta_i)}, \tag{3.1}$$

where $\mathbb{P}(\Theta_i)$ denotes the probability that a randomly chosen circuit instance lies in Θ_i. The expected error of this approximation decreases with the size of the denominator and becomes negligible if the denominator is close to 1, which can be seen as follows. Suppose that the subsets $\Theta_1, \ldots, \Theta_n$ are pairwise disjoint and $\Theta_1 \cup \ldots \cup \Theta_n = \Theta$ holds, then $\Theta_1, \ldots, \Theta_n$ is a partition of Θ. From eq. (2.13) it follows that the denominator in eq. (3.1) becomes 1 and theorem 2.11 implies the equality of both sides.

Except for mentioning a SPICE simulation, the authors don't explicitly describe the computation of the probabilities involved in eq. (3.1) and no information about the runtime of this approach is given in the paper. Furthermore, it is questionable if the assumption of a uniformly distributed resistive bridging fault resistance is realistic.

A Monte Carlo simulation based approach can naturally consider structural and spatial correlations even in very complex circuit models. However, it is well known that this approach is computationally very expensive and even requires the use of parallel hardware architectures for practical applications. For example, the authors of [Czutr12, Sauer14] compute the probability of detecting a small delay fault of fixed size with a given set of test vector-pairs under the impact of process variations by a Monte-Carlo simulation. The simulation time is reduced by simulating multiple circuit instances in parallel on a GPGPU. To avoid the large overhead of repeating the Monte-Carlo simulation of the entire test set after ever insertion or removal of a test vector-pair, the authors store the random delay values of all circuit instances together with the delay test results for all vector-pairs. However, this requires a large amount of memory. An alternative approach was taken in [Aftab09], where the authors reduce the computational cost of the Monte-Carlo simulation through compiled code simulation. However, the compile time increases rapidly with the circuit size.

After a more careful analysis of the statistical timing analysis problem, it becomes clear that a Monte-Carlo simulation of the entire circuit is unnecessary for the computation of the probability of detecting a given small delay fault of fixed size with a given set of test vector-pairs. Instead, only the critical paths (see definition 2.17), which are also likely sensitized by the test vector, can have a significant impact on the delay test result.

In [Lee05b], the authors propose to combine a sensitization analysis with a block based statistical timing analysis approach. At first, a single event driven timing simulation using only nominal delay values is performed with the given test vector-pair. This is done to identify the sensitized cone of the circuit on which the subsequent statistical timing analysis is applied. Afterwards, the distribution of the arrival time of the last transition at each gate output is computed using block-based statistical timing analysis.

However, this approach uses the normal distribution based SUM and MAX-operation and therefore requires, that all gate delays have a normal distribution. This requirement is not satisfied in low power applications, where the gate delays are more accurately approximated by a log-normal distribution [Hanso05]. Furthermore, the distribution of the size of small delay faults is widely believed to be similar to an exponential distribution [Nigh00, Hopsc10].

3.3 Statistical SUM and MAX-Operations

The following sections 3.3.1 and 3.3.2 introduce the normal distribution based SUM and MAX-operations. The main problem of block-based statistical timing analysis is the limited accuracy of the SUM and the inaccuracy of the MAX-operation in the recent technology nodes [Ul Ha11], which spurred research into a number of ways to improve the accuracy. A summary of the most promising approaches is presented in section 3.3.3.

3.3.1 Normal Distribution based SUM-operation

In the following, the SUM-operation is defined in a more general setting in which m sums and their linear relationships are computed and each sum involves an arbitrary subset of n random variables.

Definition 3.1. Let $\boldsymbol{X} = (X_1, \ldots, X_n)^T$ denote a n-dimensional normal random vector. Let $B \in \{0,1\}^{k \times n}$ be a binary matrix of rank k and let $b_{i,j} \in \{0,1\}$ denote the element in the ith row and jth column of B with $1 \leq i \leq k$ and $1 \leq j \leq n$. Then the *normal distribution based SUM-operation* computes a k-dimensional random vector $\boldsymbol{Y} = (Y_1, \ldots, Y_k)^T$, defined as

$$Y_i = \sum_{j=1}^{n} b_{i,j} X_j \quad \text{for all } 1 \leq i \leq k,$$

where $b_{i,j} = 1$ if and only if X_j is part of the ith sum Y_i, otherwise $b_{i,j} = 0$.

In a notable special case, the matrix B is a row vector of zeros and ones so that the result of the SUM-operation is a random variable Y with univariate normal distribution. The normal distribution based SUM-operation can always be accurately computed by $\boldsymbol{Y} = B\boldsymbol{X}$ using theorem 2.47 with $\boldsymbol{b} = \boldsymbol{0}$, because the family of multivariate normal distributions is closed under affine transformations.

3.3.2 Normal Distribution based MAX-operation

Definition 3.2. Given a random vector $\boldsymbol{X} = (X_1, \ldots, X_n) \sim \mathcal{N}_n(\boldsymbol{\mu}, \Sigma)$ of n-dimensional normal distribution, where $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_n)^T$ denotes the mean vector and $\Sigma \in \mathbb{R}^{n \times n}$ denotes the covariance matrix. Then the accurate approximation of the $(n-1)$-dimensional random vector

$$\boldsymbol{Y} = (Y_1, \ldots, Y_{n-1})^T = (X_1, \ldots, X_{n-2}, \max(X_{n-1}, X_n))^T. \tag{3.2}$$

with a normal distribution $\mathcal{N}_{n-1}(\hat{\boldsymbol{\mu}}, \hat{\Sigma})$ is called *normal distribution based MAX-operation*.

A very fast algorithm for the normal distribution based MAX-operation was presented by Clark [Clark61]. At first, the algorithm approximates the distribution of the maximum $Y_{n-1} = \max(X_{n-1}, X_n)$ by a normal distribution. Afterwards, the non-linear relationship between Y_{n-1} and each remaining random variable Y_i with $1 \leq i \leq n-2$ is approximated by the covariance $\mathrm{Cov}(Y_i, Y_{n-1})$, which only describes the degree of linear relationship between Y_i and Y_{n-1}.

Let $\sigma_{i,j}$ and $\hat{\sigma}_{i,j}$ denote the element in the ith row and jth column of Σ and $\hat{\Sigma}$, respectively. Following the definitions

$$a := \sqrt{\sigma_{n-1,n-1} - 2\sigma_{n-1,n} + \sigma_{n,n}} \tag{3.3}$$

$$\alpha := \frac{\mu_{n-1} - \mu_n}{a}, \tag{3.4}$$

and using eqs. (2.32) and (2.33), the mean and the variance of $Y_{n-1} = \max(X_{n-1}, X_n)$ is

$$\hat{\mu}_{n-1} = \mu_{n-1}\Phi(\alpha) + \mu_n\Phi(-\alpha) + a\phi(\alpha) \tag{3.5}$$

$$\begin{aligned}\hat{\sigma}_{n-1,n-1} = {} & (\sigma_{n-1,n-1} + \mu_{n-1}^2)\Phi(\alpha) + (\sigma_{n,n} + \mu_n^2)\Phi(-\alpha) \\ & + (\mu_{n-1} + \mu_n)a\phi(\alpha) - \hat{\mu}_{n-1}^2.\end{aligned} \tag{3.6}$$

The mean vector $\hat{\boldsymbol{\mu}}$ of $\boldsymbol{Y}$ is obtained by removing the last and replacing the second last component of $\boldsymbol{\mu}$, such that the last component of $\hat{\boldsymbol{\mu}}$ is $\hat{\mu}_{n-1}$.

Likewise, the covariance matrix $\hat{\Sigma}$ of $\boldsymbol{Y}$ is obtained from Σ by removing the last and replacing the second last row and column. The variance $\hat{\sigma}_{n-1,n-1}$ of Y_{n-1} is located in the bottom right corner of $\hat{\Sigma}$. Each of the remaining elements of the last row and column of $\hat{\Sigma}$ is replaced with the covariance between $\max(X_{n-1}, X_n)$ and X_i, which is computed by

$$\hat{\sigma}_{n-1,i} = \hat{\sigma}_{i,n-1} = \sigma_{i,n-1}\Phi(\alpha) + \sigma_{i,n}\Phi(-\alpha) \quad \text{for all } 1 \leq i \leq n-2. \tag{3.7}$$

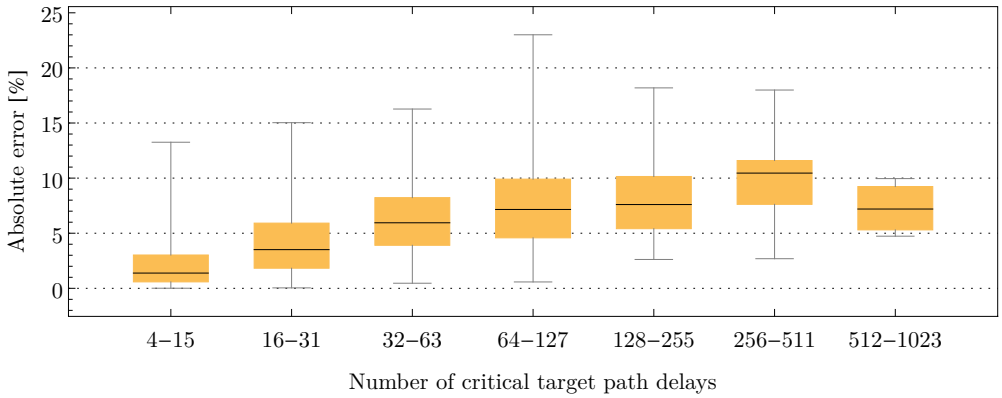
Although the computational cost of this algorithm is very small, it can be further reduced by separate handling of boundary cases like $\Phi(\alpha) \approx 1$ for $\alpha > 4$ [Kuruv13].

Suppose $\boldsymbol{X} = (X_1, \ldots, X_n)^T$ is a random vector of critical target path delays and the normal distribution based MAX-operation is repeatedly applied to approximate

$\max(X_1, \ldots, X_n)$ with a random variable Z of univariate normal distribution. A critical target path is a target path of a test vector-pair that also satisfies eq. (2.25) of definition 2.17. Let the absolute error ϵ of this approximation be defined as the maximum absolute difference

$$\epsilon = \max_{t \in \mathbb{R}} \left(\left| \mathbb{P}(Z \leq t) - \mathbb{P}(\max(X_1, \ldots, X_n) \leq t) \right| \right). \tag{3.8}$$

A box plot (see section 2.4.2) of the absolute error $|\epsilon|$ is shown in fig. 3.1 in percent for a large number of critical target path sets of various sizes. It is apparent that using the normal distribution based MAX-operation to compute the statistical maximum can result in a large absolute error of more than 20%. Suppose the number n of critical target path delays is between 64 and 127, then the absolute error caused by approximating the distribution of $\max(X_1, \ldots, X_n)$ by repeated application of the normal distribution based MAX-operation was found to be between 5% and 10% in about 50% of the conducted experiments.



▲ **Figure 3.1 —** Box plot of the absolute error $|\epsilon|$ of the approximation of the distribution of $\max(X_1, \ldots, X_n)$ by using normal distribution based MAX-operation, where $X_1, \ldots, X_n$ are critical target path delays

3.3.3 Latest Approaches to Improve Accuracy of MAX-operation

In [Liou01, Devga03], the authors model gate delays as discrete random variables, described by a joint probability mass function. However, the computational cost of the MAX-operation increases exponentially in the number of possible delay values due to structural and spacial correlations.

In [Chopr06], the authors define the MAX-operation based on the two piece normal distribution, which has to be distinguished from the skew-normal distribution proposed by Azzalini [Azzal13]. This approach is fundamentally limited by the unfortunate choice of the bivariate two-piece normal distribution as the underlying joint distribution of the operands of the MAX-operation, because this distribution cannot accurately

represent the linear relationship between the random variables. Also, the relationship between $\max(X_{n-1}, X_n)$ and the remaining random variables $X_1, \ldots, X_{n-2}$ is ignored.

The authors of [Cheng08] use the identity

$$\max(X_i, X_j) = X_i + \max(0, X_j - X_i) \tag{3.9}$$

and estimate the coefficients of a second order polynomial to approximate the distribution of $\max(0, X_j - X_i)$. However, this significantly complicates the approximation of $\max(X_i, X_j)$ due to the singularity of the distribution of $\max(0, X_j - X_i)$ at 0. Even more importantly, this identity creates a non-linear relationship between X_i and $\max(0, X_j - X_i)$, which clearly depends on the condition $X_j > X_i$. In fact, this approach transforms the problem of accurately computing the distribution of the maximum of correlated random variables into a new an equally hard problem of computing the sum of X_i and $\max(0, X_j - X_i)$.

A gaussian mixture model based approach to represent the time and slope/slew of transitions was studied in [Tsuki11]. The computation utilizes the algorithm for the normal distribution based MAX-operation. However, accurately fitting mixture models is known to be very difficult due to the higher number of parameters. The benefit of this MAX-operation in terms of accuracy could only be demonstrated for certain extreme cases. This approach also ignores the relationship between $\max(X_{n-1}, X_n)$ and the remaining random variables $X_1, \ldots, X_{n-2}$.

In [Vijay14], the authors investigate the use of the skew-normal distribution based MAX-operation. However, the presented algorithm only considers the last two random variables X_{n-1} and X_n and ignores the relationship between $\max(X_{n-1}, X_n)$ and the remaining random variables $X_1, \ldots, X_{n-2}$.

CHAPTER 4

Probabilistic Sensitization Analysis

This chapter presents the novel probabilistic sensitization analysis, which efficiently detects and analyses path delay fault test invalidation for a given test vector-pair. Some of the techniques described here are used in the next chapter for the computation of the target paths delay fault probability (see definition 1.5).

Section 4.1 formally defines the probabilistic sensitization analysis problem. A method to detect the invalidation of a path delay fault test is introduced in section 4.2. Section 4.3 presents an efficient analysis algorithm, which identifies the paths that control the sensitization of a target path during the application of a test vector-pair under the impact of delay variations. To evaluate the impact of path delay fault test invalidation on the delay test with a given test vector-pair, a representative subcircuit is constructed in section 4.4, which enables the efficient Monte-Carlo simulation of the test vector-pair.

The experimental results are presented in section 7.3.

4.1 Probabilistic Sensitization Analysis Problem

The problem of path delay fault test invalidation is addressed by solving the underlying fundamental statistical timing analysis problem, which is stated as follows.

Definition 4.1 (Representative Subcircuit)**.** Given a test vector-pair v that can sensitize any path from a set of critical paths Π with non-negligible probability. Let $\theta \in \Theta$ denote a randomly chosen circuit instance and let $\mathcal{S}$ be a subcircuit of θ of well defined fixed structure, such that $\mathcal{S}$ contains at least the paths Π and all floating inputs of $\mathcal{S}$ are set to either constant '0' or '1'. The subcircuit $\mathcal{S}$ is called *representative subcircuit for* v, if the probability that Π contains a critical path that is sensitized by v in only either $\mathcal{S}$ or θ, is below a given threshold.

It is important to emphasize that the structure of the subcircuit $\mathcal{S}$ is fixed but $\mathcal{S}$ always "inherits" the propagation delay values from the circuit instance it is compared to. Suppose $\theta_1, \theta_2 \in \Theta$ with $\theta_1 \neq \theta_2$ are two unique circuit instances, then the propagation delays of the fixed set of gates in $\mathcal{S}$ are equal to those of θ_1 or equal to those of θ_2, depending on if $\mathcal{S}$ is compared to θ_1 or to θ_2.

Definition 4.2. Finding the minimal representative subcircuit for a given test vector-pair is called the *probabilistic sensitization analysis problem*.

A simple analytical method for the computation of the representative subcircuit is explained by the example circuit shown in fig. 4.1, which has inputs 'a','b','c' and 'd' and output 'l'. A test vector-pair $((1,1,0,1)^T, (0,0,1,0)^T)$ is applied to the circuit inputs, which is supposed to sensitize the path d-h-k-l. The sensitization of the path depends on the waveforms at its off-path inputs 'c' and 'j'. The transition at the off-path 'c' is known to occur at time 0. If X_3 denotes the delay of the inverter u_3, then $X_3 > 0$ must hold for the rising transition at 'h' to be propagated to the gate output 'k', which is always satisfied.

The analysis of the off-path input 'j' is more complex. At first, it is determined if a glitch can occur at 'i'. Let X_2 and X_5 denote the delay of u_2 and u_5, respectively. Furthermore, let X_6 denote the delay of u_6 to propagate the falling transition from 'g' to 'i'. Then

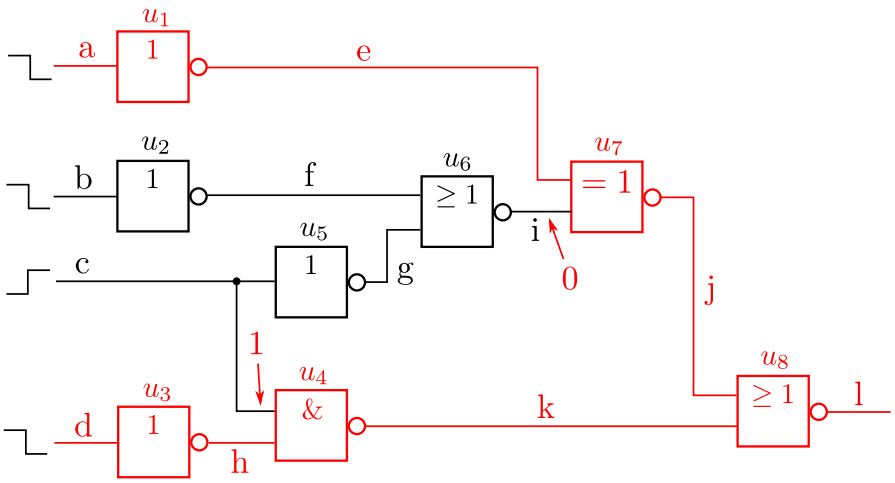
$$X_5 + X_6 < X_2 \tag{4.1}$$

is necessary for the glitch to occur at 'i'. Given the probability distribution of the random vector $(X_2, X_5, X_6)^T$, the probability of the event defined by eq. (4.1) can be computed. In this example, this probability is extremely small so that 'i' can be assumed constant logic value '0'. If X_1 denotes the delay of u_1 and X_7 denotes the delay of u_7 to propagate the transition from 'e' to 'j', then a falling transition will occur at 'j' at time $X_1 + X_7$. Similarly, a falling transition will occur at 'k' at time $X_3 + X_4$, where X_4 denotes the delay of u_4 to propagate the transition from 'h' to 'k'. Then

$$X_1 + X_7 < X_3 + X_4 \tag{4.2}$$

is necessary for the path d-h-k-l to be sensitized, that is, for the falling transition at 'k' to be propagated to the circuit output 'l'. Given the probability distribution of the random vector $(X_1, X_3, X_4, X_7)^T$, the probability of the event defined by eq. (4.2) can be computed. Suppose $\mathbb{P}(X_1 + X_7 < X_3 + X_4) \approx 0.5$, then the sensitization of the path d-h-k-l clearly depends on the path a-e-j, which must therefore be part of the representative subcircuit. In this case, the path a-e-j is said to be a *controlling path* of the path d-h-k-l.

Definition 4.3 (Controlling Path)**.** A path p' in the input cone of a path p is said to be a *controlling path* of p, if in a randomly chosen circuit instance $\theta \in \Theta$ a random variation of the delays of the gates on p' changes the sensitization of p (that is p becomes sensitized/not sensitized) with non-negligible probability.



▲ **Figure 4.1 —** Circuit and representative subcircuit (red) for a given test vector-pair

Any test invalidation mechanism is immediately apparent from the structure of the representative subcircuit. In fact, the delay test generation process is not only made aware of precisely those off-path inputs of the sensitized critical paths where test invalidation is likely to occur, but the invalidation is also explained by the responsible controlling paths. Beyond applying more stringent sensitization conditions directly to the identified off-path inputs, the delay test generation can also prevent the sensitization of the controlling paths that are responsible for the test invalidation.

For any critical path p, the probability that p is sensitized by the test vector-pair in a randomly chosen circuit instance can be efficiently computed with a Monte-Carlo simulation of the representative subcircuit. It will be shown that the construction and Monte-Carlo simulation of the subcircuit is much faster than a Monte-Carlo simulation of the complete circuit.

The analytical approach shown here works well for small circuits, but becomes very complex for larger circuits. A more efficient method, which scales well to large circuits, will be developed in the following sections of this chapter.

4.2 Identification and Comparison of Sensitized Paths

For a given circuit instance and a test vector-pair, this section describes the efficient identification of the sensitized paths, along which transitions propagate from the circuit inputs to the circuit outputs.

At first, the given circuit instance is simulated with the given test vector-pair. During the simulation, each transition at the output of a gate is assigned a reference to its predecessor transition at one of the inputs of the gate. These references are stored such that after the simulation, any sensitized path can be efficiently identified by following the reference of a transition to its predecessor transition until a circuit input is reached.

Alternatively, if the trace starts at a circuit input, then the transition is traced to the first gate on the path. To identify the output transition corresponding to the input transition, it is checked that a transition τ' at the gate output has a reference to the transition τ at the input of the gate. If no such output transition τ' exists, then τ violates the propagation condition of the gate and the path terminates at the gate input. This process is repeated until the end of the path has been reached.

To efficiently identify and compare the sensitized critical paths in different circuit instances, a hash from the structural information of each sensitized critical path is computed and used for identification.

4.3 Analysis of Target Path Sensitization by Test Vector-Pair

This section details the probabilistic sensitization analysis of a single target path, which is sensitized by a given test vector-pair in the nominal circuit instance θ_0. The analysis is performed on n randomly selected circuit instances $\theta_1 \in \Theta, \ldots, \theta_n \in \Theta$. This approach is extended in section 4.4 to all critical paths that are sensitizable by the given test vector-pair considering delay variations. The result of the analysis is a subcircuit consisting of at least the target path. If the target path is not sensitized in at least one circuit instance, then this subcircuit is extended by the controlling paths of the target path.

For any $i \in \mathbb{N}$ with $1 \leq i \leq n$, the basic principle of the analysis is to compare and analyse the path sensitization in a circuit instance $\theta_i \in \Theta$ with the path sensitization in the subcircuit $\mathcal{S}$, which initially contains only the target path.

Definition 4.4. A path is said to be *consistently controlled* by a test vector-pair in a circuit instance θ and the subcircuit $\mathcal{S}$, if the path is sensitized in either both the circuit instance θ and the subcircuit $\mathcal{S}$ or in none of them. Otherwise, the path is said to be *inconsistently controlled* in θ and $\mathcal{S}$.

If the target path is inconsistently controlled in a circuit instance θ_i and the subcircuit $\mathcal{S}$, then the sensitization of the target path depends on a structural path that is missing in $\mathcal{S}$. The proposed analysis explains an inconsistently controlled target path by tracing waveform inconsistencies inside the input cones of the gate, which blocks the propagation of the transition along the path in either the simulation of θ_i or the simulation of $\mathcal{S}$.

Definition 4.5. If τ denotes a transition which is propagated along an inconsistently controlled path, then the waveform at the terminal node of the path is said to have the *waveform inconsistency* τ.

An example for a waveform inconsistency is a transition, which is part of a glitch that exists in only either the simulation of θ_i or in the simulation of $\mathcal{S}$, but not in both.

Once a waveform inconsistency $\tilde{\tau}$ has been identified, the analysis is applied to the inconsistently controlled path along which $\tilde{\tau}$ was propagated. In each step, the proposed

analysis moves closer to the circuit inputs until a sensitized path, which is missing in $\mathcal{S}$, is found. Afterwards, the subcircuit $\mathcal{S}$ is extended with the corresponding structural path by adding all gates and interconnects of that path. All unspecified off-path inputs are set to the logic values, which they had during the simulation of the circuit instance θ_i at the time the transition occurred at the on-path input of the gate.

In some cases, multiple paths jointly determine the sensitization of the target path in the given circuit instance. Furthermore, the extension of the subcircuit made by the analysis for one circuit instance might necessitate further extensions of the subcircuit for other circuit instances. Therefore, the simulation of the subcircuit and the proposed analysis is repeatedly applied to all circuit instances until the target path is consistently controlled in the subcircuit $\mathcal{S}$ and all circuit instances $\theta_1, \ldots, \theta_n$. The number of circuit instances n is chosen such that the probability, that the target path is inconsistently controlled in any circuit instance $\theta \in \Theta$ and the subcircuit $\mathcal{S}$, is sufficiently small.

The following subsections present the details of the proposed analysis. For simplicity, it is assumed that the inconsistently controlled path under analysis is sensitized only in the circuit instance θ_i but not in the subcircuit S. However, the same description also applies to the opposite case by swapping θ_i with $\mathcal{S}$. The tracing of an inconsistently controlled path is explained in section 4.3.1. The analysis of the propagation condition of a gate is detailed in section 4.3.2.

4.3.1 Tracing of Inconsistently Controlled Path

To explain an inconsistently controlled target path, the propagation of the transition is traced backwards starting at an observable circuit output node. In each step, the transition at the input of the gate is identified by following the reference from the output transition. Suppose the path is sensitized by the given test vector-pair in θ_i but not in $\mathcal{S}$, then the backtracing continues as long as the following two conditions are satisfied:

(i) The gate input node is part of the subcircuit $\mathcal{S}$.

(ii) The transition at the gate input occurs in θ_i but is missing in the subcircuit $\mathcal{S}$.

If the first condition (i) is violated, then at least one gate of the currently traced path must be missing in the subcircuit $\mathcal{S}$. If the second condition (ii) is violated, then the gate input transition τ is only propagated through this gate in the simulation of θ_i, but not in the simulation of $\mathcal{S}$. By definition, the circuit instance θ_i and the subcircuit $\mathcal{S}$ always share the same gate and interconnect delays. This implies that a waveform inconsistency $\tilde{\tau}$ must exist at one of the inputs of the gate, which causes τ to violate the propagation condition during the simulation of $\mathcal{S}$. After the responsible waveform inconsistency $\tilde{\tau}$ has been identified, the analysis continues by backwards tracing $\tilde{\tau}$ in the same manner. The identification of the responsible waveform inconsistency is explained in the next subsection.

4.3.2 Analysis of Transition Propagation Condition

This subsection details the analysis of the propagation condition of the gate, which blocks the propagation of a transition along the currently traced path. Let $\tau = (v, t)$ be a transition at an input node of the gate, which was propagated along a consistently controlled path. However, τ satisfies the propagation condition of the gate, without loss of generality, only in the simulation of the circuit instance θ_i, causing a gate output transition $\tau' = (v', t')$, but not in the simulation of the subcircuit $\mathcal{S}$. Then the proposed analysis must identify a responsible waveform inconsistency $\tilde{\tau}$ at one of the inputs of the gate.

Analysis of Dynamic Sensitization Condition

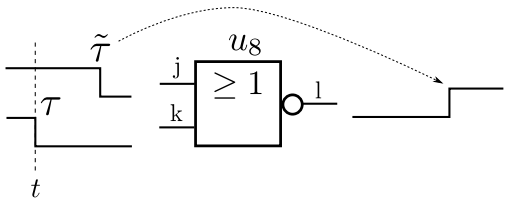
At first, the waveform at the off-path input node of the gate is analysed to check if τ violates the dynamic sensitization condition in the simulation of the subcircuit $\mathcal{S}$. If that is the case, then at least one waveform inconsistency must exist at the off-path input and the analysis proceeds by tracing the waveform inconsistency $\tilde{\tau}$, which occurs closest to t at the off-path input. Otherwise, τ must violate the inertial delay condition of the gate in the simulation of $\mathcal{S}$.

Let $\tau = (v, t)$ denote a transition at the on-path input node of the gate at time t. For the dynamic sensitization condition to be satisfied, the off-path input node of the gate must have the non-controlling value at the time t. More generally to consider also XOR/NXOR gates, the off-path input node must have the same logic value during the simulation of θ_i and $\mathcal{S}$ at time t. If this condition is not satisfied, then at least one waveform inconsistency must exist at the off-path input node and the analysis proceeds by tracing the waveform inconsistency $\tilde{\tau}$ at the off-path input node. If multiple waveform inconsistencies exist at the off-path input, then the one closest to time t is selected. Needless to say, this condition is always satisfied for logic gates with only a single input.

A simple example is given by the last NOR gate (u_8) in fig. 4.1 with a falling transition at each gate input. Figure 4.2 shows the input and output waveforms of the gate during the simulation of the circuit instance θ_i, where the transition $\tilde{\tau}$ at the off-path input 'j' of the target path occurs after the transition τ at the on-path input 'k'. As a consequence, τ violates the dynamic sensitization condition of the gate so that the target path is not sensitized. Suppose the path a-e-j does not exist in the subcircuit, then $\tilde{\tau}$ occurs in the simulation of θ_i but not in the simulation of $\mathcal{S}$ and is therefore a waveform inconsistency. Consequently, the analysis proceeds by backwards tracing $\tilde{\tau}$ and then finally adds the path a-e-j to the subcircuit.

Analysis of Inertial Delay Condition

If τ satisfies the dynamic sensitization condition but τ is not propagated to the gate output, then τ must violate the inertial delay condition. For an inverter or a buffer, this implies that the waveform at the gate input node must contain at least one waveform



▲ **Figure 4.2 —** Transition τ violates and $\tilde{\tau}$ satisfies dynamic sensitization condition of NOR gate

inconsistency. Then the proposed analysis identifies and starts tracing the last waveform inconsistency $\tilde{\tau}$ before the transition $\tau' = (v', t')$ occurs at the gate output in θ_i.

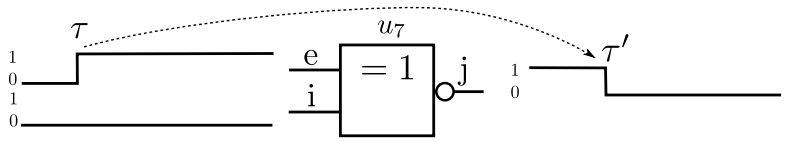
The analysis of the inertial delay condition for a logic gate with two inputs is more complex. Clearly, only those input transitions which satisfy the dynamic sensitization condition can have an impact on the gate output waveform. From among those transitions, only those which occur before or directly after τ but no later than t' are *relevant* for the propagation of τ. Let L_{θ_i} and $L_{\mathcal{S}}$ denote the set of relevant transitions at the inputs of the gate in the simulation of $\mathcal{S}$ and θ_i, respectively. Then a waveform inconsistency must exist in $L_{\theta_i} \cup L_{\mathcal{S}}$, which is responsible for τ satisfying the inertial delay condition of the gate in either the simulation of θ_i or the simulation of $\mathcal{S}$, but not in both.

A simple example is provided by the XNOR gate u_7 in fig. 4.1, under the assumption that the subcircuit $\mathcal{S}$ also contains the path c-g-i-j-l with 'f' set to logic '0'. In other words, the subcircuit $\mathcal{S}$ is obtained from the complete circuit, depicted in fig. 4.1, by removing the path b-f and setting 'f' to logic '0'. The waveforms at the inputs and output of gate u_7 are shown in fig. 4.3 for random gate delays. The problem is that the glitch at the output 'i' of u_6 does not appear in the simulation of the circuit instance θ_i, so that 'i' remains constant zero. However, in the simulation of the subcircuit $\mathcal{S}$, the falling transition at 'g' is propagated through u_6, because 'f' is defined as constant '0'. As a consequence, a rising transition $\tilde{\tau}$ appears at 'i' shortly after the rising transition τ at 'e'. In this example it is assumed that the time between τ and $\tilde{\tau}$ is too short for a glitch to appear at the output 'j' so that τ violates the inertial delay condition.

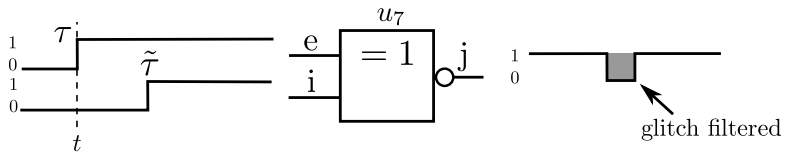
For the example given above, the sets of relevant transitions are

$$L_{\theta_i} = \{\tau\}$$
$$L_{\mathcal{S}} = \{\tau, \tilde{\tau}\}.$$

The proposed analysis then explains the violation of the inertial delay condition by τ with the first waveform inconsistency in $L_{\theta_i} \cup L_{\mathcal{S}}$ after t. If no such waveform inconsistency exists, then the last waveform inconsistency in $L_{\theta_i} \cup L_{\mathcal{S}}$ before t is selected instead. In this example, the only waveform inconsistency in $L_{\theta_i} \cup L_{\mathcal{S}}$ is $\tilde{\tau}$, which occurs directly after τ.



(a) Transition τ is propagated in circuit instance θ_i



(b) Transition τ violates inertial delay condition in subcircuit $\mathcal{S}$

▲ **Figure 4.3 —** Input and output waveforms of XNOR gate u_7 in the circuit instance and its subcircuit after the simulation of a test vector-pair

4.4 Construction of Representative Subcircuit

Under the impact of large delay variations, all possible delay tests of a target path may be affected by path delay fault test invalidation. In this case, it is necessary to identify those delay tests that have a sufficiently low probability of test invalidation or to apply a suitable combination of path delay fault tests to the target path. All necessary probabilities can be efficiently obtained from a Monte-Carlo simulation of a small representative subcircuit, which will be constructed in this section.

The idea is to apply the proposed analysis in section 4.3 to all critical paths that can be sensitized by the given test vector-pair with non-negligible probability. It will be shown that the construction and Monte-Carlo simulation of the resulting subcircuit is much faster than the Monte-Carlo simulation of the complete circuit. One important observation is that the result of a delay test with a given test vector-pair is uniquely determined by only the critical paths that are sensitized by the test vector-pair in the circuit instance. A formal definition of a critical path is given by definition 2.17.

Given a circuit instance θ_i and a test vector-pair, the analysis starts by identifying all sensitized critical paths in θ_i and in the subcircuit $\mathcal{S}$. A critical path might be sensitized in θ_i but not in $\mathcal{S}$ for two possible reasons. The first reason is that the corresponding structural path is missing in $\mathcal{S}$ and must be added to $\mathcal{S}$. Otherwise, the path is inconsistently controlled and the sensitization analysis described in section 4.3 is applied to that path.

It is also possible for a critical path to be sensitized only in the subcircuit $\mathcal{S}$ but not in the circuit instance θ_i. In that case, it can only be an inconsistently controlled path and the sensitization analysis described in section 4.3 is applied to that path.

To repeat the simulation of the subcircuit $\mathcal{S}$ after adding one or more paths, the subcircuit is reconstructed from all gates and interconnects which lie on at least one of the paths of the subcircuit. All floating gate input nodes are set to their respective non-controlling values. For the floating off-path input node of a XOR/XNOR gate, the

logic value the off-path input had at the time of the on-path transition is used. Finally, all floating circuit outputs are set to the value observed at the clock cycle time.

The procedure outlined above is repeatedly applied to all circuit instances $\theta_0, \ldots, \theta_n$ until every critical path, which is sensitized in at least one circuit instance, is consistently controlled in all circuit instances $\theta_0, \ldots, \theta_n$. The number of circuit instances is chosen such that the probability that a critical path is sensitized by the test vector-pair in only either the randomly chosen circuit instance θ_i or the subcircuit $\mathcal{S}$, is sufficiently small.

For the given test vector-pair, the sensitization probabilities of the critical paths can now be efficiently computed with a Monte-Carlo simulation of the obtained representative subcircuit $\mathcal{S}$. In each iteration, the sensitization of a particular critical path can be determined by tracing the transition from the beginning to the end of the path, as described in section 4.2.

4.5 Simplified Probabilistic Sensitization Analysis

This section presents a simplified version of the sensitization analysis, which merely detects likely path delay fault test invalidation but doesn't explain the invalidation by the responsible paths. Instead, this analysis provides the location of the gate which blocks the propagation of the transition along the target path and other useful details for the path delay fault test generation process.

A given test vector-pair is simulated with the nominal circuit instance θ_0 and a randomly chosen circuit instance $\theta_i \in \Theta$ with $1 \leq i \leq n$. Afterwards, the simulation results are compared and any critical path that is sensitized in the nominal circuit instance, but not in the randomly chosen circuit instance, is subjected to this analysis.

By tracing the propagation of the transition along the path as explained in section 4.3.1, the gate which blocks the propagation is identified. Afterwards, it is determined if the propagation condition for the gate has changed because of a waveform inconsistency at one of the gate inputs. In this context, the term "waveform inconsistency" is redefined and now describes for a given test vector-pair a transition, which is propagated along a logical path in only either the nominal or the randomly chosen circuit instance, but not in both. For example, due to delay variations, a glitch may have appeared or disappeared or any of the gate input transitions may have been propagated along different paths.

If a waveform inconsistency exists at one of the gate inputs, then it is possible that this inconsistency directly causes the invalidation of the path delay fault test. In other words, the gate might not cause but merely propagate the effects of a waveform inconsistency from inside the input cones of the gate inputs. Alternatively, the gate itself might directly cause the waveform inconsistency at the gate output due to a variation of the arrival time of the input transitions, regardless of any waveform inconsistency at the gate inputs.

To distinguish these two cases, the proposed analysis first corrects all waveform inconsistencies at the gate inputs. This is done by tracing the gate input transitions in

the nominal circuit instance θ_0 and computing the propagation delay of these paths in the randomly chosen circuit instance θ_i. Afterwards, the simulation of the gate is repeated and it is checked if the target transition is successfully propagated. This analysis can be extended to arbitrary modifications of the input waveforms.

4.6 Summary

The invalidation of path delay fault tests is a serious problem which can cause a large number of test escapes. Probabilistic sensitization analysis is essential to efficiently guide the delay test generation process into generating path delay fault tests that are more tolerant towards delay variations.

This chapter introduced a novel probabilistic sensitization analysis, which examines likely path delay fault test invalidation mechanisms. The analysis not only provides the location of the gate which blocks the propagation of the transition along a target path, but it also identifies the controlling paths that control the sensitization of the target path.

The probability of path delay fault test invalidation with a given test vector-pair is evaluated by constructing a representative subcircuit, which allows the efficient Monte-Carlo simulation of the test vector-pair. The results of the analysis are essential to efficiently compare the effectiveness of alternative test vector-pair and combinations of test vector-pairs in order to minimize the risk of path delay fault test invalidation without increasing test cost.

CHAPTER 5

Computation of Target Paths Delay Fault Probability

This chapter presents an incremental and a non-incremental algorithm for the computation of the *target paths delay fault probability*, which is defined by definition 1.5, assuming a single small delay fault of fixed size is present in the circuit. The analysis considers only those test vector-pairs, which are applicable for the detection of the small delay fault.

Definition 5.1. The term *test subset* refers to the subset of all test vector-pairs in the test set, which target at least one path through the site of the particular small delay fault under consideration.

The set of target paths of a test vector-pair is defined by definition 1.2.

The computation of the joint delay distribution of the critical target paths, presented in section 5.1, is a fundamental part of both algorithms. The non-incremental algorithm is described in section 5.2. In section 5.3, the faster but slightly less accurate incremental algorithm is presented. The incremental algorithm requires an extension of the normal distribution based MAX-operation, which is introduced in section 5.4.

In section 7.4, both algorithms are applied to approximate the delay fault detection probability of a given test vector-pair, which is defined by definition 7.1.

5.1 Computation of Critical Target Paths Delay Distribution

For a given small delay fault of fixed size and a suitable test subset, this section present the computation of the joint delay distribution of a subset of all target paths. This

subset contains only those target paths that have a significant impact on the target paths delay fault probability. The precise meaning of the term "target path" is given by definition 1.2.

5.1.1 Identification of Target Paths

For a given test vector-pair, the target paths are identified using the method described in section 4.2. At first, the nominal circuit instance is simulated with the given test vector-pair. Afterwards, the sensitized paths are identified by tracing the transitions from the circuit outputs to the circuit inputs. This procedure is repeated for each test vector-pair.

In general, a large number of paths might be targeted by the test vector-pairs in the test subset. However, only a small subset of those target paths might have a path delay fault with non-negligible probability. For the computation of the target paths delay fault probability, it is sufficient to consider only those target paths that have a path delay fault with non-vanishing probability. Therefore, the analysis in this chapter can be restricted to the set of *critical target paths*, which is defined as the intersection of the set of target paths (definition 1.2) and the set of critical paths (definition 2.17).

5.1.2 Computation of Delay Distribution of a Target Path

To distinguish between critical and non-critical target paths, the delay distribution of every target path must be computed. Let n denote the number of gates along a particular target path and let the random variables $U_1, \ldots, U_n$ denote the corresponding propagation delays of the gates along the path.

For independent gate delays $U_1, \ldots, U_n$, the convolution of all probability density functions of $U_1, \ldots, U_n$ is the probability density function of their sum, that is, the probability density function of the target path delay. Furthermore, the central limit theorem implies that under certain mild conditions, the delay of a path with a sufficiently large number of gates has approximately a normal distribution, regardless of the gate delay distributions. This statement can be proven by theorem 2.25, which is also called the *Lindeberg-Feller central limit theorem*. By using the notation of theorem 2.25 with U_i and $\bar{U}$ instead of X_i and $\bar{X}$, respectively,

$$\sqrt{n}(\bar{U}_n - \bar{\mu}_n) = \frac{1}{\sqrt{n}} \left(\sum_{i=1}^{n} U_i - n\bar{\mu}_n \right), \tag{5.1}$$

so that after multiplication with $\sqrt{n}$ and subtraction of $n\bar{\mu}_n$, eq. (2.36) becomes

$$\sum_{i=1}^{n} U_i \xrightarrow{d} \mathcal{N}(n\bar{\mu}_n, n\bar{\sigma}^2). \tag{5.2}$$

In other words, under the conditions of theorem 2.25, the sum of independent random variables $\sum_{i=1}^{n} U_i$ converges in distribution (denoted by $\xrightarrow{d}$) to a normal distribution with mean $n\bar{\mu}_n$ and variance $n\bar{\sigma}^2$.

Generalizations of the central limit theorem exist for dependent random variables [Louhi02], which supports the assumption that the critical target path delay can be accurately approximated by a normal distribution even if the gate delays $U_1, \ldots, U_n$ are not normally distributed.

If the gate delays are not independent, all gate delays are grouped into a random vector

$$\boldsymbol{U} = (U_1, \ldots, U_n)^T. \tag{5.3}$$

The computation of the target path delay distribution simplifies if $\boldsymbol{U}$ has approximately a normal distribution $\mathcal{N}_n(\boldsymbol{\mu_U}, \Sigma_{\boldsymbol{U}})$ with mean vector $\boldsymbol{\mu_U}$ and covariance matrix $\Sigma_{\boldsymbol{U}}$. Then theorem 2.47 with

$$A = (1, \ldots, 1) \in \mathbb{R}^{1 \times n}$$

implies that the delay X of the target path also has a normal distribution with mean

$$\mathbb{E}(X) = A\boldsymbol{\mu_U} \tag{5.4}$$

and variance

$$\mathrm{Var}(X) = A\Sigma_{\boldsymbol{U}}A^T. \tag{5.5}$$

The above equation also considers any spacial correlations between the gate delays, which are represented by the off-diagonal elements of the covariance matrix $\Sigma_{\boldsymbol{U}}$.

5.1.3 Computation of Joint Delay Distribution

Let n be redefined as the number of critical target paths, sensitized by a test vector-pair. The delays of n critical target paths are grouped into a n-dimensional normal random vector $\boldsymbol{X} = (X_1, \ldots, X_n)^T$. The approach described in this chapter requires, that the joint delay distribution of $\boldsymbol{X}$ can be accurately approximated by a normal distribution. According to theorem 2.37, this is satisfied if all gate delays are jointly normally distributed. In this case, the joint path delay distribution can be efficiently computed using the normal distribution based SUM-operation, presented in section 3.3.1. In any other case, the joint delay distribution of $\boldsymbol{X}$ can be approximated by a normal distribution by computing the mean vector and the covariance matrix of $\boldsymbol{X}$. An example of such an approach is presented in the next chapter in section 6.3.1.

In the following, it is assumed that the random vector of critical target path delays $\boldsymbol{X} \sim \mathcal{N}_n(\boldsymbol{\mu}, \Sigma)$ has a multivariate normal distribution with mean vector $\boldsymbol{\mu} \in \mathbb{R}^n$ and covariance matrix $\Sigma \in \mathbb{R}^{n \times n}$. The i-th component μ_i of the mean vector $\boldsymbol{\mu}$ represents the mean of the i-th path delay. Likewise, the i-th diagonal element of the covariance matrix is the variance of the i-th path delay.

The off-diagonal elements of the covariance matrix describe the degree of linear relationship between the path delays, originating from structural and spacial correlation. If only structural correlations are considered, then the covariance between two path delays is equal to the sum of the variances of all gate delays, which are shared by both paths. If also spacial correlations are considered, then the covariance between the

delays of two target paths is computed as follows. Let $U_1 + \ldots + U_{k_1}$ denote the sum of k_1 gate delays along path A and let $V_1 + \ldots + V_{k_2}$ denote the sum of k_2 gate delays along path B. Then the covariance between the delay of path A and the delay of path B is

$$\mathrm{Cov}(U_1 + \ldots + U_{k_1}, V_1 + \ldots + V_{k_2}) = \sum_{i=1}^{k_1} \sum_{j=1}^{k_2} \mathrm{Cov}(U_i, V_j), \tag{5.6}$$

which follows directly from theorem 2.37 with $\boldsymbol{X} = (U_1, \ldots, U_{k_1}, V_1, \ldots, V_{k_2})^T$,

$$A = \begin{pmatrix} \overbrace{1 & \cdots & 1}^{k_1} & \overbrace{0 & \cdots & 0}^{k_2} \\ 0 & \cdots & 0 & 1 & \cdots & 1 \end{pmatrix} \in \mathbb{R}^{2 \times (k_1 + k_2)} \tag{5.7}$$

and $\boldsymbol{b} = \boldsymbol{0}$.

5.2 Non-Incremental Computation

As stated by definition 1.5, the target paths delay fault probability is the probability, that at least one of the target paths has a path delay fault. That is, the delay of at least one target path is larger than the clock cycle time T_{clk}. Instead of enumerating all possible cases in which at least one path has a path delay fault, it is more efficient to compute the probability of the complementary event

$$\overline{\Psi} = \mathbb{P}(\{\theta \in \Theta : \text{"none of the target paths has a path delay fault"}\}). \tag{5.8}$$

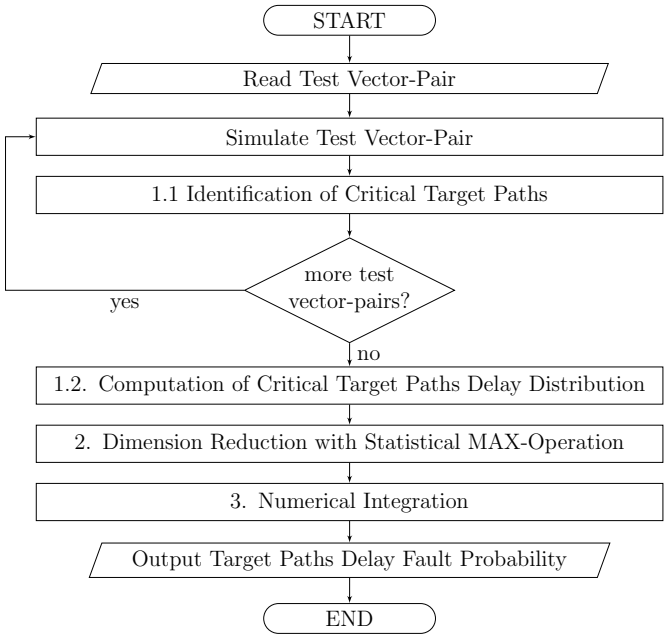
Once $\overline{\Psi}$ has been computed, the target paths delay fault probability is

$$\Psi = 1 - \overline{\Psi}. \tag{5.9}$$

The flowchart of the proposed non-incremental algorithm for the computation of the target paths delay fault probability is presented in fig. 5.1. The algorithm approximates the probability in three steps, which are described below.

5.2.1 Computation of Critical Target Paths Delay Distribution

In the first step, the joint delay distribution of the critical target paths, which terminate at an unmasked observable circuit output node, is computed as described in section 5.1. If the probability that a particular target path has a path delay fault is very large (e.g. above 0.98), then the target paths delay fault probability will be close to one and the remaining steps of the algorithm can be omitted. It is important to note that a path might be targeted by multiple test vector-pairs in the test subset. To identify and remove any duplicates among the critical target paths, a unique hash signature is computed from the input transition and structural information (e.g. unique gate and interconnect names) of each critical target path.



▲ **Figure 5.1 —** Flowchart of the non-incremental algorithm

Let n be redefined as the number of critical target paths, that are sensitized by all test vector-pairs in a given test subset. Then let the random vector $\boldsymbol{X}$ be defined as

$$\boldsymbol{X} = (X_1, \ldots, X_n)^T, \tag{5.10}$$

where $X_1, \ldots, X_n$ denote the delays of the critical target paths. By assumption $\boldsymbol{X}$ has a multivariate normal distribution.

5.2.2 Dimension Reduction with Statistical MAX-Operation

In the second step of the algorithm, the dimension of $\boldsymbol{X}$ is reduced by the repeated application of the normal distribution based MAX-operation, which was introduced in section 3.3.2. By the first application of the normal distribution based MAX-operation, the distribution of the random vector

$$(X_1, \ldots, X_{n-2}, \max(X_{n-1}, X_n))^T \tag{5.11}$$

is approximated by a $(n-1)$-dimensional normal random vector $(X_1, \ldots, X_{n-2}, Y)^T$. By the second application of the normal distribution based MAX-operation, the random vector

$$(X_1, \ldots, X_{n-3}, \max(X_{n-2}, Y))^T \tag{5.12}$$

is approximated by a $(n-2)$-dimensional normal random vector $(X_1, \ldots, X_{n-3}, \tilde{Y})^T$. This process continues until the number of random variables has dropped below a user defined threshold. In this approach, the normal distribution based MAX-operation is applied as long as the number of random variables is greater than 1000.

5.2.3 Numerical Integration

The target paths delay fault probability is computed in the last step with an efficient numerical integration algorithm. Let $m \leq n$ denote the number of the remaining random variables after the previous step and let

$$(X_1, \ldots, X_{m-1}, \tilde{X}_m)^T \sim \mathcal{N}_m(\boldsymbol{\mu}, \Sigma) \tag{5.13}$$

denote the m-dimensional normal random vector consisting of the normal distribution approximation of the maximum delay $\tilde{X}_m$ of a subset of the critical target paths and the delays $X_1, \ldots, X_{m-1}$ of the remaining critical target paths. Then the probability that none of the target paths has a path delay fault is approximately

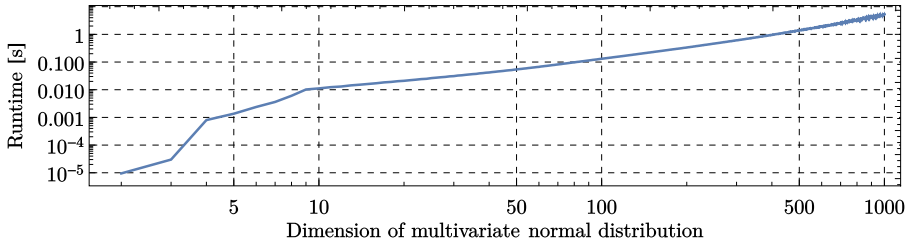
$$\overline{\Psi} = \mathbb{P}\big(X_1 \leq T_{clk}, \ldots, X_{m-1} \leq T_{clk}, \tilde{X}_m \leq T_{clk}\big). \tag{5.14}$$

According to eq. (2.40), the RHS of the above equation is merely an integral over the probability density function of the m-dimensional normal distribution, specifically

$$\overline{\Psi} = \int_{-\infty}^{T_{clk}} \cdots \int_{-\infty}^{T_{clk}} \phi_m(\boldsymbol{x}; \boldsymbol{\mu}, \Sigma) dx_1 ... dx_m. \tag{5.15}$$

The above integral can be efficiently approximated by specialized numerical integration algorithms for small [Genz04] and large dimensions [Genz92], which quickly converge to the necessary accuracy. However, care must be taken because any linear dependency (multicollinearity) between the path delays $X_1, \ldots, X_n$ implies that $\mathrm{Var}(\boldsymbol{X})$ is not positive definite and therefore not a covariance matrix. Various techniques are available to transform such a matrix into a positive definite matrix [Wothk93]. In this approach, all diagonal elements of $\mathrm{Var}(\boldsymbol{X})$ are multiplied by a very small constant greater than unity. Given $\overline{\Psi}$, the target paths delay fault probability is finally computed with eq. (5.9).

The advantage of the non-incremental method is its relatively high accuracy. However, the generation of multivariate normal random numbers for the Monte-Carlo simulation in [Genz92] requires the computation of the Cholesky factorization of the covariance matrix $\Sigma \in \mathbb{R}^{m \times m}$, which has $O(m^3)$ worst-case runtime complexity [Golub13]. For example, fig. 5.2 shows the average runtime of the algorithms [Genz04, Genz92] on a workstation with Intel(R) Core(TM) i7-2600 CPU at 3.40GiHz for a large number of randomly chosen multivariate normal distributions, which describe the delays of the critical target paths for a large number of test vector-pairs.



▲ **Figure 5.2 —** Runtime for approximating the integral in eq. (5.15)

5.3 Incremental Computation

This section presents an efficient incremental algorithm for the computation of the target paths delay fault probability. The idea is to describe each test vector-pair by the maximum delay of its sensitized critical target paths.

Definition 5.2 (Delay of Test Vector-Pair)**.** If the random vector $(X_{i,1}, \ldots, X_{i,n_i})^T$ consists of the delays of all n_i critical target paths, which are sensitized by the i-th test vector-pair in the nominal circuit instance θ_0, then the random variable

$$Y_i = \max(X_{i,1}, \ldots, X_{i,n_i}) \tag{5.16}$$

is called the *delay* of the i-th test vector-pair.

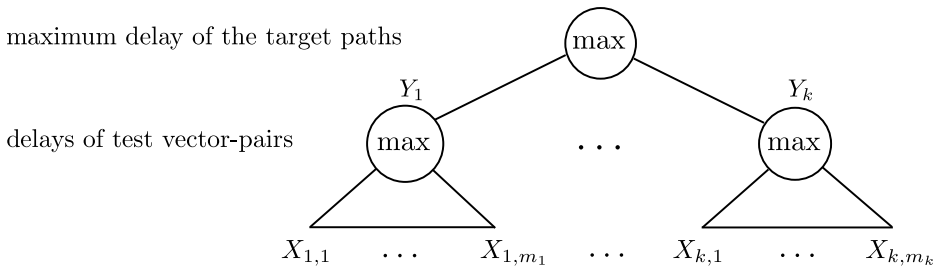
The delays of k test vector-pairs are then grouped into a k-dimensional random vector

$$\boldsymbol{Y} = (Y_1, \ldots, Y_k)^T, \tag{5.17}$$

which is of central importance in this approach. This is because the maximum delay of the target paths (see definition 1.4) can be expressed by the maximum $\max(Y_1, \ldots, Y_k)$ of the delays of the test vector-pairs, as illustrated in fig. 5.3.

For greater efficiency, the distribution of $\boldsymbol{Y}$ is approximated by a multivariate normal distribution. The incremental computation is then based on efficient modifications of the parameters of this multivariate normal distribution. For example, the random vector $\boldsymbol{Y}$ can easily be extended by the delay of a new test vector-pair without the need to recompute the distribution of $(Y_1, \ldots, Y_k)^T$ itself.

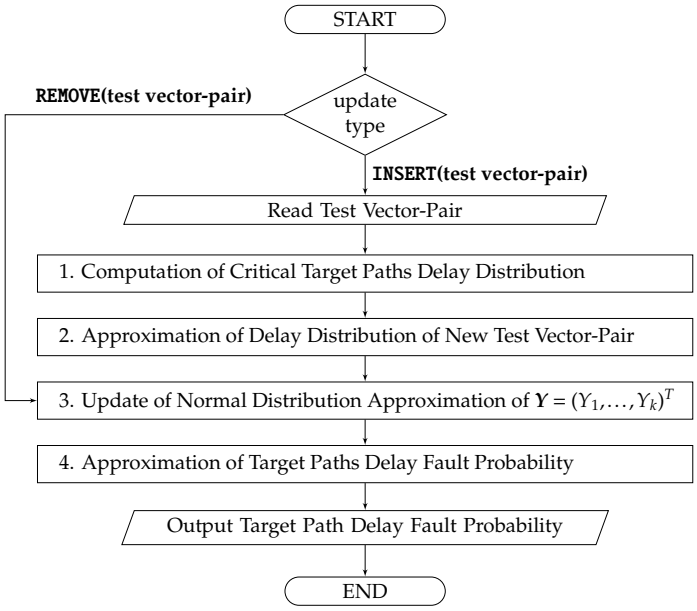
The proposed approach exploits that delay test parameter updates tend to imply only small changes to the distribution of $\boldsymbol{Y}$. For example, adding or removing a test vector-pair will only extend or reduce the distribution of $\boldsymbol{Y}$ by one dimension. Hence, the runtime for approximating the change of the target paths delay fault probability after a delay test parameter update can be substantially reduced. This approach also benefits from the typically small dimension of $\boldsymbol{Y}$. However, the disadvantage of this approach is the slightly reduced accuracy due to the approximation of the distribution of $\boldsymbol{Y}$ by a multivariate normal distribution.



▲ **Figure 5.3 —** Computation of the maximum delay of the target paths in two steps

The following description focuses on the insertion and the removal of a test vector-pair from the test subset. The update of other delay test parameters is described in section 5.3.4. The flowchart in fig. 5.4 shows the four major steps of the proposed algorithm. Following the insertion of a new test vector-pair, the joint delay distribution of the critical target paths, which are sensitized by the new test vector-pair, is determined in the first step as described in section 5.1. In the second step, the delay of the new test vector-pair is approximated by a normal distribution. Afterwards, the multivariate normal distribution, which approximates the distribution of $\boldsymbol{Y}$, is extended by one dimension in step 3. Finally, the target paths delay fault probability is computed in step 4 using an efficient numerical integration method.

Following the removal of a test vector-pair from the test subset, the corresponding entries in the mean vector and the covariance matrix of the multivariate normal distribution approximation of $\boldsymbol{Y}$, are deleted in step 3. Finally, the target paths delay fault probability is computed in the last step.



▲ **Figure 5.4 —** Flowchart of the incremental algorithm

5.3.1 Approximation of Delay Distribution of New Test Vector-Pair

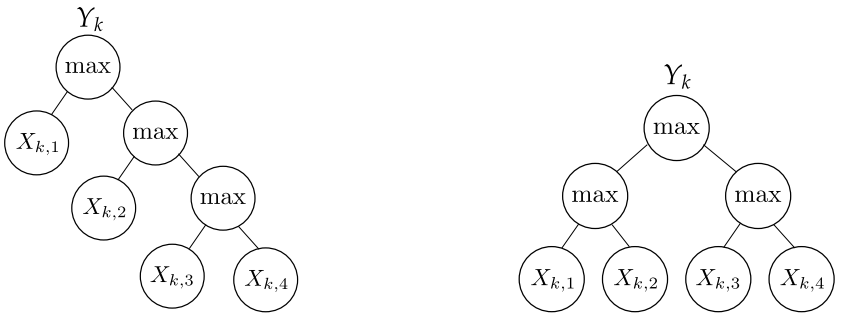
The proposed algorithm requires that the delay of any test vector-pair is described by a normal distribution. However, the distribution of the maximum of multiple normally distributed random variables is not normally distributed [Nadar08]. Then eq. (5.16) implies that the delay of a test vector-pair, which sensitizes at least two critical target paths, is not normally distributed. To satisfy the condition of the algorithm, this subsection explains how the delay of a test vector-pair can be approximated by a normal distribution. It is assumed, that a given test vector-pair extends the current test subset from $k-1$ to k test vector-pairs.

Let the random vector $(X_{k,1}, \ldots, X_{k,n_k})^T$ consist of the delays of all critical target paths that are sensitized by the new test vector-pair. Then according to eq. (5.16), the delay of the new test vector-pair is

$$Y_k = \max(X_{k,1}, \ldots, X_{k,n_k}). \tag{5.18}$$

The distribution of the delay Y_k is approximated by a normal distribution using the extension of the normal distribution based MAX-operation that is presented in section 5.4. The main advantage of using this extension is the additional flexibility, which allows the formation of balanced binary tree like dataflow graphs, as shown in fig. 5.5b. If changes to the distribution of the critical target paths are made (e.g. by masking or unmasking of observable circuit outputs), only those MAX-operations which lie on the path to Y_k must be recomputed.

Without this extension, the normal distribution based MAX-operation forms a list of MAX-operations, as shown in fig. 5.5a. As a consequence, on average half of the MAX-operation results depend on the distribution of a single critical target path delay.



(a) classical (b) proposed extension

▲ **Figure 5.5 —** Approximation of $\max(X_{k,1}, X_{k,2}, X_{k,3}, X_{k,4})$ using normal distribution based MAX-operation

5.3.2 Update of Test Vector-Pairs Delay Distribution

By approximating the delay distribution of each test vector-pair by a normal distribution, the distribution of the random vector $\boldsymbol{Y} = (Y_1, \ldots, Y_k)^T$ is approximated by a multivariate normal distribution with mean vector $\boldsymbol{\mu_Y}$ and covariance matrix $\Sigma_{\boldsymbol{Y}}$.

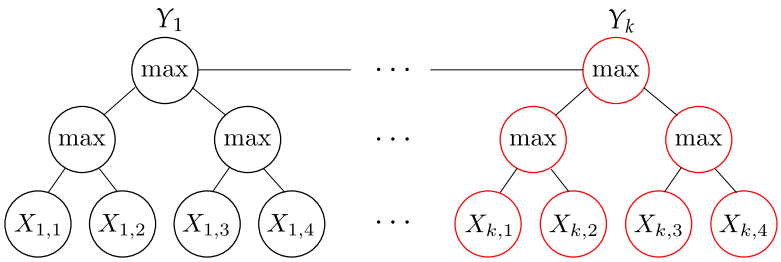
Following the insertion of a new test vector-pair into the test subset, the dimension of the random vector $\boldsymbol{Y}$ increases by one. Consequently, the length of the mean vector and the dimension of the covariance matrix increases from $k-1$ to k, as shown in fig. 5.6. The new last entry of the mean vector equals $\mu_k = \mathbb{E}(Y_k)$ and the new diagonal element in the bottom right corner of the covariance matrix equals $\sigma_{k,k} = \mathrm{Var}(Y_k)$, where Y_k denotes the delay of the new test vector-pair that was approximated in section 5.3.1. The remaining entries in the last row and last column of the covariance matrix are the covariances between the delay Y_k of the new test vector-pair and the delays $Y_1, \ldots, Y_{k-1}$ of the remaining test vector-pairs. These entries still need to be computed.

$$\boldsymbol{\mu}_{\boldsymbol{Y}} = \begin{pmatrix} \mu_1 \\ \vdots \\ \mu_{k-1} \\ \mu_k \end{pmatrix} \qquad \Sigma_{\boldsymbol{Y}} = \begin{pmatrix} \sigma_{1,1} & \cdots & \sigma_{1,k-1} & \sigma_{1,k} \\ \vdots & \ddots & \vdots & \vdots \\ \sigma_{k-1,1} & \cdots & \sigma_{k-1,k-1} & \sigma_{k-1,k} \\ \sigma_{k,1} & \cdots & \sigma_{k,k-1} & \sigma_{k,k} \end{pmatrix}$$

▲ **Figure 5.6 —** Extension of the mean vector $\boldsymbol{\mu}_{\boldsymbol{Y}}$ and the covariance matrix $\Sigma_{\boldsymbol{Y}}$ of the normal distribution approximation of $\boldsymbol{Y}$, after insertion of a test vector-pair

The normal distribution based MAX-operation doesn't provide a function to compute the covariance $\sigma_{k,i} = \mathrm{Cov}(Y_k, Y_i)$ between two maxima of normally distributed random variables. This function will be introduced in section 5.4 as the proposed extension of the normal distribution based MAX-operation.

The computation of the covariances proceeds as shown in fig. 5.7. The nodes at the leaf level represent the critical target path delays and the root nodes represent the delays of the test vector-pairs. The internal nodes symbolize the application of the normal distribution based MAX-operation. To compute the covariances between the root nodes, the computation starts at the leaf level and proceeds level by level until the root nodes have been reached. The proposed covariance computation function in section 5.4 is applied to all pairs of nodes in one level, where one node is in the tree corresponding to Y_k and the other node is in any of the other trees at the same level. After completing one level, the computation proceeds with the level above. It is not necessary to compute the covariances within the new tree for Y_k itself, since those have already been computed during the approximation of the distribution of Y_k.



▲ **Figure 5.7 —** Example for the forest data structure of the incremental algorithm, where each tree represents the computation of the delay of a test vector-pair.

Following the removal of a test vector-pair. The dimension of the distribution of $\boldsymbol{Y}$ and the corresponding multivariate normal distribution approximation decreases from k to $k-1$. Consequently, the entries in the mean vector $\boldsymbol{\mu}_{\boldsymbol{Y}}$ and the covariance matrix $\Sigma_{\boldsymbol{Y}}$, corresponding to the removed test vector-pair, have to be deleted.

The computational cost for extending the multivariate normal distribution increases linearly with the test subset size k under the assumption, that the number of target paths, sensitized by any test vector-pair in the nominal circuit instance θ_0, is bounded by some constant upper bound.

5.3.3 Approximation of Target Paths Delay Fault Probability

After the test subset size has been increased or reduced to k test vector-pairs, the probability $\overline{\Psi}$ that none of the target paths have a path delay fault is

$$\overline{\Psi} = \mathbb{P}(Y_1 \leq T_{clk}, \ldots, Y_k \leq T_{clk}). \tag{5.19}$$

The proposed approach approximates the joint distribution of the test vector-pair delays $\boldsymbol{Y} = (Y_1, \ldots, Y_k)^T$ by a k-dimensional normal distribution $\mathcal{N}_k(\boldsymbol{\mu_Y}, \Sigma_{\boldsymbol{Y}})$ with mean vector $\boldsymbol{\mu_Y}$ and covariance matrix $\Sigma_{\boldsymbol{Y}}$. Then using eq. (5.9) and theorem 2.48, the approximation $\hat{\Psi}$ of the target paths delay fault probability is

$$\hat{\Psi} = 1 - \int_{-\infty}^{T_{clk}} \cdots \int_{-\infty}^{T_{clk}} \phi_k(\boldsymbol{y}; \boldsymbol{\mu_Y}, \Sigma_{\boldsymbol{Y}}) dy_1 ... dy_k, \tag{5.20}$$

where $\boldsymbol{y} = (y_1, \ldots, y_k)$ and $\phi_k(\boldsymbol{y}; \boldsymbol{\mu}, \Sigma)$ denotes the probability density function of the k-dimensional normal distribution $\mathcal{N}_k(\boldsymbol{\mu_Y}, \Sigma_{\boldsymbol{Y}})$. This integral can be approximated by efficient numerical integration algorithms for small [Genz04] and large dimensions [Genz92] of the random vector $\boldsymbol{Y}$.

5.3.4 Changing other Delay Test Parameters

The proposed incremental computation approach can be applied to other delay test parameter updates. For example if changes to the clock cycle time or the masking of observable circuit outputs are made, the set of target paths can be stored and reused for each test vector-pair. However, some of the target paths may become critical target paths if the clock cycle time is reduced or observable circuit outputs are unmasked. Likewise, some critical target paths may need to be removed if observable circuit outputs are masked.

If the set of critical target paths of a test vector-pair changes, the algorithm proceeds as if this test vector-pair had been added to the test set. However, instead of extending the multivariate normal distribution approximation of the distribution of $\boldsymbol{Y}$, only the operations affected by the modification of the set of critical target paths must be repeated.

5.4 Extension of Normal Distribution based MAX-operation

Let $(X_1, \ldots, X_4)^T$ denote a 4-dimensional normal random vector. This section introduces a function for the normal distribution based MAX-operation, which computes the covariance between the two maxima $U = \max(X_1, X_2)$ and $V = \max(X_3, X_4)$. The accurate computation of this covariance is of critical importance for the proposed incremental algorithm.

The normal distribution based MAX-operation [Clark61] computes the covariances $\mathrm{Cov}(X_1, V)$ and $\mathrm{Cov}(X_2, V)$ using eq. (3.7), which can be written as

$$\mathrm{Cov}(X_k, \max(X_i, X_j)) = \mathbb{P}\Big(X_i > X_j\Big)\mathrm{Cov}(X_k, X_i) + \mathbb{P}\Big(X_i \leq X_j\Big)\mathrm{Cov}(X_k, X_j). \quad (5.21)$$

However, the resulting random vector $(X_1, X_2, V)^T$ doesn't have a multivariate normal distribution. Therefore, the above formula cannot be applied a second time to accurately compute the covariance $\mathrm{Cov}(\max(X_1, X_2), V)$.

The accurate covariance $\mathrm{Cov}(U, V)$ is computed from theorem 2.34, which states that

$$\mathrm{Cov}(U, V) = \mathbb{E}(UV) - \mathbb{E}(U)\mathbb{E}(V), \quad (5.22)$$

where the formula for the mean values $\mathbb{E}(U)$ and $\mathbb{E}(V)$ of the maxima is already provided by eq. (3.5) as part of the normal distribution based MAX-operation. The cross-moment $\mathbb{E}(UV)$ is expanded by applying the definition of the maximum over the real numbers $x_1, \ldots, x_4$, which gives

$$uv = \begin{cases} x_1 x_3 & \text{if } x_1 > x_2, x_3 > x_4 \\ x_1 x_4 & \text{if } x_1 > x_2, x_3 \leq x_4 \\ x_2 x_3 & \text{if } x_1 \leq x_2, x_3 > x_4 \\ x_2 x_4 & \text{if } x_1 \leq x_2, x_3 \leq x_4. \end{cases}$$

By replacing u and v with the random variables U and V, the above conditions imply a partition of the sample space into four partitions

$$\begin{aligned} A_1 &= \{\theta \in \Theta : X_1(\theta) > X_2(\theta), X_3(\theta) > X_4(\theta)\} \\ A_2 &= \{\theta \in \Theta : X_1(\theta) > X_2(\theta), X_3(\theta) \leq X_4(\theta)\} \\ A_3 &= \{\theta \in \Theta : X_1(\theta) \leq X_2(\theta), X_3(\theta) > X_4(\theta)\} \\ A_4 &= \{\theta \in \Theta : X_1(\theta) \leq X_2(\theta), X_3(\theta) \leq X_4(\theta)\}, \end{aligned}$$

where for clarity, $X_i(\theta)$ is written instead of the short-hand notation X_i for all $i = 1, \ldots, 4$ to emphasize the fact that a random variable is a function that depends on the outcome θ of the random experiment (see eq. (2.19)). Using this notation and eq. (2.1), $\mathbb{E}(UV)$ can formally be written as

$$\begin{aligned} \mathbb{E}(U(\theta)V(\theta)) = &\mathbb{E}(X_1(\theta)X_3(\theta)\mathbb{1}_{A_1}(\theta)) \\ &+ \mathbb{E}(X_1(\theta)X_4(\theta)\mathbb{1}_{A_2}(\theta)) \\ &+ \mathbb{E}(X_2(\theta)X_3(\theta)\mathbb{1}_{A_3}(\theta)) \\ &+ \mathbb{E}(X_2(\theta)X_4(\theta)\mathbb{1}_{A_4}(\theta)). \end{aligned} \quad (5.23)$$

Again using the shorthand notation, theorem 2.43 implies that

$$\begin{aligned} \mathbb{E}(UV) = &\mathbb{E}(X_1 X_3 \,|A_1)\mathbb{P}(A_1) \\ &+ \mathbb{E}(X_1 X_4 \,|A_2)\mathbb{P}(A_2) \\ &+ \mathbb{E}(X_2 X_3 \,|A_3)\mathbb{P}(A_3) \\ &+ \mathbb{E}(X_2 X_4 \,|A_4)\mathbb{P}(A_4). \end{aligned} \quad (5.24)$$

For simplicity, the following presentation will focus on the random vector $\boldsymbol{W}$, which is a linear transformation of $\boldsymbol{X}$, defined as

$$(W_1, W_2, W_3, W_4)^T = (X_1, X_1 - X_2, X_3, X_3 - X_4)^T,$$

According to theorem 2.47, the transformed random vector $\boldsymbol{W}$ also has a 4-dimensional normal distribution with mean vector

$$\bar{\boldsymbol{\mu}} = (\mu_1, \mu_1 - \mu_2, \mu_3, \mu_3 - \mu_4)^T$$

and covariance matrix

$$\bar{\Sigma} = \begin{pmatrix} \sigma_{1,1} & \sigma_{1,1} - \sigma_{1,2} & \sigma_{1,3} & \sigma_{1,3} - \sigma_{1,4} \\ & \sigma_{1,1} + \sigma_{2,2} - 2\sigma_{1,2} & \sigma_{1,3} - \sigma_{2,3} & \sigma_{1,3} - \sigma_{1,4} - \sigma_{2,3} + \sigma_{2,4} \\ & & \sigma_{3,3} & \sigma_{3,3} - \sigma_{3,4} \\ & & & \sigma_{3,3} + \sigma_{4,4} - 2\sigma_{3,4} \end{pmatrix},$$

where μ_i denotes the mean of X_i and $\sigma_{i,j}$ denotes the covariance $\mathrm{Cov}(X_i, X_j)$ with $i, j \in \{1, \ldots, 4\}$. Due to spacial limitations, only the upper triangular part of the covariance matrix $\bar{\Sigma} = \{a_{i,j}\}$ is shown here. The lower triangular part can easily be obtained due to the symmetry $\bar{\Sigma} = \bar{\Sigma}^T$ of the covariance matrix.

Let $\bar{\mu}_i$ denote the mean and $a_i = \sqrt{a_{i,i}}$ denote the standard deviation of W_i. Then $\rho_{i,j} = a_{i,j}/(a_i a_j)$ is called *Pearson correlation coefficient*, which describes the correlation between W_i and W_j. After introducing the following notations

$$\alpha_i = -\bar{\mu}_i / a_i \tag{5.25}$$

$$\beta_2 = (\alpha_4 - \rho_{2,4}\alpha_2) / \sqrt{1 - \rho_{2,4}^2} \tag{5.26}$$

$$\beta_4 = (\alpha_2 - \rho_{2,4}\alpha_4) / \sqrt{1 - \rho_{2,4}^2} \tag{5.27}$$

and using the relationship

$$\mathbb{E}(W_i W_j) = a_{i,j} + \bar{\mu}_i \bar{\mu}_j, \tag{5.28}$$

the formulas for the truncated multivariate normal distribution, presented in [Birnb51] and [Talli61], are simplified to

$$\begin{aligned} \mathbb{E}(UV) = {} & \mathbb{E}(W_1 W_3) \\ & - \mathbb{E}(W_1 W_4)\Phi(\alpha_4) + (\bar{\mu}_1 - \bar{\mu}_2 \Phi(\beta_4))\, a_4 \phi(\alpha_4) \\ & - \mathbb{E}(W_2 W_3)\Phi(\alpha_2) + (\bar{\mu}_3 - \bar{\mu}_4 \Phi(\beta_2))\, a_2 \phi(\alpha_2) \\ & + \mathbb{E}(W_2 W_4)\Phi_2(\alpha_2, \alpha_4; \rho_{2,4}) + (1 - \rho_{2,4}^2) a_2 a_4 \phi_2(\alpha_2, \alpha_4; \rho_{2,4}) \end{aligned} \tag{5.29}$$

with

$$\mathbb{E}(U) = \bar{\mu}_1 - \bar{\mu}_2 \Phi(\alpha_2) + a_2 \phi(\alpha_2) \tag{5.30}$$

$$\mathbb{E}(V) = \bar{\mu}_3 - \bar{\mu}_4 \Phi(\alpha_4) + a_4 \phi(\alpha_4), \tag{5.31}$$

where ϕ and Φ are defined by eqs. (2.32) and (2.33) and ϕ_2 and Φ_2 are defined by eqs. (2.64) and (2.65), respectively. The cumulative distribution function Φ_2 of the

standard bivariate normal distribution can be efficiently approximated using a fast double precision algorithm [Genz04].

In one particular special case where $\rho_{2,4} = 0$, the computation can be simplified to

$$\mathrm{Cov}(U, V) = a_{1,3} - a_{1,4}\Phi(\alpha_4) - a_{2,3}\Phi(\alpha_2). \tag{5.32}$$

Another noticeable special case is when either U or V has a normal distribution, in which case the computation reduces to the special case considered by eq. (5.21).

5.5 Conclusion

Large delay variations severely degrade the quality and reliability of small delay fault tests, which can result in a large number of test escapes. In order to find a suitable compromise between the delay test quality and the test cost, recent variation aware delay test generation methods are guided by the probability that at least one target path has a path delay fault. This probability can change significantly even by small modifications of the delay test parameters or the set of test vector-pairs.

This chapter presented two efficient algorithms for the computation of the target paths delay fault probability. The non-incremental algorithm provides high accuracy but may become inefficient if the delay test parameters are frequently modified. To address this shortcoming, an efficient incremental algorithm has been presented to minimize the computational cost after delay test parameter modifications. This algorithm is well suited to evaluate the impact of small changes to the test subset and delay test parameters on the target paths delay fault probability. The high efficiency and small approximation error of this algorithm makes it suitable for the inner loop of automatic test pattern generation methods.

In section 7.4, both algorithms have been compared to the computation of the delay fault detection probability (definition 7.1) using extensive Monte-Carlo simulations. The results show a very large speedup with only a small loss of accuracy, mainly caused by path delay fault test invalidation.

The main limitation of the path based approach taken in this chapter is that the number of critical target paths might get extremely large. While a Monte-Carlo simulation can be used in those easily identifiable cases, a block based approach may be much more efficient. The fundamental operations of a block based approach are the focus of the next chapter.

CHAPTER 6

SUM and MAX-Operations based on Skew-Normal Distribution

This chapter introduces the skew-normal distribution based SUM and MAX-operation. The new MAX-operation serves as a plug-in replacement for the normal distribution based MAX-operation and significantly reduces the approximation error.

This chapter consists of six parts. The skew-normal distribution is introduced in section 6.1. A detailed description of the SUM operation and its role in statistical timing analysis is given in section 6.2. Section 6.3 presents the basic algorithm for the skew-normal distribution based MAX-operation. The following two sections 6.4 and 6.5 introduce optimizations which minimize the runtime complexity of this algorithm without sacrificing the accuracy. The last section 6.6 explains the approximation of the distribution of $\max(X_1, \ldots, X_n)$ using the skew-normal distribution based MAX-operation. The experimental results are presented in section 7.5.

6.1 The Skew-Normal Distribution

The skew-normal distribution is a natural extension of the normal distribution. The definition of the skew-normal distribution in the original parameterization is presented in section 6.1.1. To simplify the presentation of the statistical SUM and MAX-operations, an alternative parameterization is adopted in section 6.1.2. It is subsequently shown in section 6.1.3 that the two definitions are indeed equivalent. For compactness of notation let $b = \sqrt{2/\pi}$.

6.1.1 Definition with Azzalini-Parametrization

This section presents the definitions and a few key properties of the univariate and the multivariate skew-normal distribution as defined in [Azzal13, chapter 2]. If a random variable X has probability density function

$$f_1(x) = \frac{2}{\omega}\phi\left(\frac{x-\xi}{\omega}\right)\Phi\left(\alpha\frac{x-\xi}{\omega}\right) \quad (-\infty < x < \infty) \tag{6.1}$$

where ϕ and Φ are defined by eqs. (2.32) and (2.33), then X is a skew-normal random variable with location parameter ξ, scale parameter ω and shape parameter α. The moment generating function of the univariate skew-normal distribution is

$$M(t) = 2\exp\left(\xi t + \frac{1}{2}\omega^2 t^2\right)\Phi(\delta\omega t) \tag{6.2}$$

Differentiating $M(t)$ yields the formulas for the moments of a random variable X, which are

$$\begin{aligned}
\mathbb{E}(X^1) &= \xi + b\omega\delta \\
\mathbb{E}(X^2) &= \xi^2 + 2b\xi\omega\delta + \omega^2 \\
\mathbb{E}(X^3) &= \xi^3 + 3b\xi^2\omega\delta + 3\xi\omega^2 + 3b\omega^3\delta - b\omega^3\delta^3
\end{aligned}$$

with $\delta = \alpha/\sqrt{1+\alpha^2}$.

A n-dimensional random vector $\boldsymbol{Z}$ is said to have a multivariate skew-normal distribution if it is continuous with density function

$$2\phi_n(z; \mathbf{0}, \bar{\Omega})\Phi\left(\boldsymbol{\alpha}^T z\right), \tag{6.3}$$

where $\phi_n(z; \mathbf{0}, \bar{\Omega})$ is the n-dimensional normal probability density function with zero mean vector and correlation matrix $\bar{\Omega}$, $\Phi(\cdot)$ is given by eq. (2.33), and $\boldsymbol{\alpha} \in \mathbb{R}^n$ is a n-dimensional vector [Azzal13, chapter 5]. For simplicity, $\bar{\Omega}$ is assumed to be of full rank.

In eq. (6.3), the location and scale parameters were omitted. To introduce them, let

$$\boldsymbol{X} := \boldsymbol{\xi} + \omega\boldsymbol{Z}, \tag{6.4}$$

where

$$\boldsymbol{\xi} = (\xi_1, \ldots, \xi_k)^T \tag{6.5}$$

is the location parameter vector and

$$\omega = \operatorname{diag}(\omega_1, \ldots, \omega_k) \tag{6.6}$$

is the scale parameter matrix, which is a square diagonal matrix with the positive elements $\omega_1, \ldots, \omega_k$ on the main diagonal. Then the n-dimensional random vector $\boldsymbol{X}$ has the probability density function

$$f_n(\boldsymbol{x}) = 2\phi_n(\boldsymbol{x}; \boldsymbol{\xi}, \Omega)\Phi\left(\boldsymbol{\alpha}^T\omega^{-1}(\boldsymbol{x}-\boldsymbol{\xi})\right) \tag{6.7}$$

where $\Omega = \omega\bar{\Omega}\omega$ is a covariance matrix.

Let $\boldsymbol{t} \in \mathbb{R}^n$, then the moment generating function for the multivariate skew-normal distribution is

$$M(\boldsymbol{t}) = 2\exp\left(\boldsymbol{t}^T\xi + \frac{1}{2}\boldsymbol{t}^T\Omega\boldsymbol{t}\right)\Phi\left(\delta^T\omega\boldsymbol{t}\right) \tag{6.8}$$

Differentiating $M(\boldsymbol{t})$ with respect to t_i, t_j and t_k with $1 \leq i,j,k \leq n$ and setting $\boldsymbol{t} = \boldsymbol{0}$ yields

$$\begin{aligned}
\mathbb{E}(X_i) &= \xi_i + b\delta_i\omega_i && (6.9)\\
\mathbb{E}(X_iX_j) &= b\delta_i\xi_j\omega_i + \omega_{i,j} + \xi_i(\xi_j + b\delta_j\omega_j) && (6.10)\\
\mathbb{E}(X_i^2X_j) &= 2\xi_i(b\delta_i\xi_j\omega_i + \omega_{i,j}) + \xi_i^2(\xi_j + b\delta_j\omega_j)\\
&\quad + \omega_i^2(\xi_j + b\delta_j\omega_j) + b\delta_i\omega_i(2\omega_{i,j} - \delta_i\delta_j\omega_i\omega_j) && (6.11)\\
\mathbb{E}(X_iX_j^2) &= 2\xi_j\omega_{i,j} + \xi_i(\xi_j^2 + 2b\delta_j\xi_j\omega_j + \omega_j^2)\\
&\quad + b(2\delta_j\omega_{i,j}\omega_j + \delta_i\omega_i(\xi_j^2 - \delta_j^2\omega_j^2 + \omega_j^2)) && (6.12)\\
\mathbb{E}(X_iX_jX_k) &= \xi_k\omega_{i,j} + \xi_j\omega_{i,k} + \xi_i(b\delta_j\xi_k\omega_j + \omega_{j,k} + \xi_j(\xi_k + b\delta_k\omega_k))\\
&\quad + b(\delta_j\omega_{i,k}\omega_j + \delta_k\omega_{i,j}\omega_k + \delta_i\omega_i(\xi_j\xi_k + \omega_{j,k} - \delta_j\omega_j\delta_k\omega_k)) && (6.13)
\end{aligned}$$

All first and second order moment can be written as follows [Azzal13, eq. (5.31-32)]

$$\mathbb{E}(\boldsymbol{X}) = \boldsymbol{\xi} + b\omega\boldsymbol{\delta} \tag{6.14}$$

$$\mathrm{Var}(\boldsymbol{X}) = \Omega - b^2\omega\boldsymbol{\delta}\boldsymbol{\delta}^T\omega \tag{6.15}$$

6.1.2 Alternative Parametrization Adopted in this Work

To simplify the presentation of the statistical SUM and MAX-operations, an alternative parameterization is introduced, which will be used throughout this chapter.

Univariate Skew-Normal Distribution

Definition 6.1. A random variable X is called skew-normal random variable with mean μ, variance σ^2 and shape parameter λ if

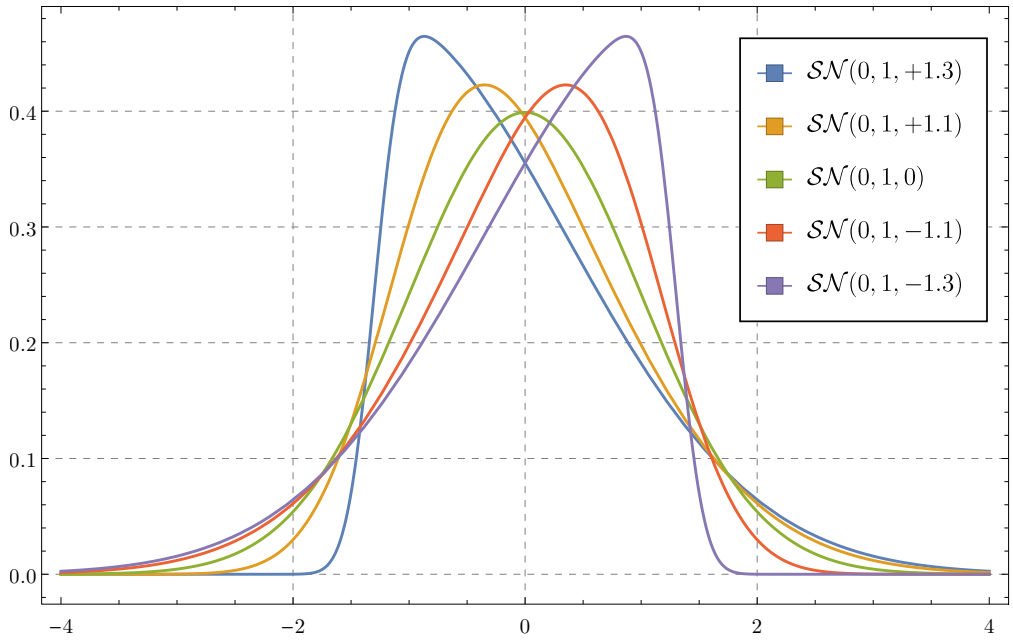
$$|\lambda| < \sqrt{2/(\pi - 2)}\sigma \tag{6.16}$$

and the probability density function of X is

$$f_1(x) = \frac{2}{\sqrt{\lambda^2 + \sigma^2}}\phi\left(\frac{x + \lambda - \mu}{\sqrt{\lambda^2 + \sigma^2}}\right)\Phi\left(\frac{\lambda(x + \lambda - \mu)}{b(\lambda^2 + \sigma^2)\sqrt{1 - \lambda^2/(b^2(\lambda^2 + \sigma^2))}}\right), \tag{6.17}$$

where $b = \sqrt{2/\pi}$. The distribution of X is then denoted by $\mathcal{SN}(\mu, \sigma^2, \lambda)$.

The probability density function of the univariate skew-normal distribution for different choices of parameters is shown in fig. 6.1.



▲ **Figure 6.1 —** Probability density function of univariate skew-normal distribution

The moment generating function $M(t) = \mathbb{E}(\exp(tX))$ of the skew-normal random variable $X \sim \mathcal{SN}(\mu, \sigma^2, \lambda)$ is

$$M(t) = 2\exp\left((\mu - \lambda)t + \frac{1}{2}(\sigma^2 + \lambda^2)t^2\right)\Phi(\lambda t/b). \tag{6.18}$$

Differentiating $M(t)$ with respect to t and setting $t = 0$ yields the formulas for the first three moments of a random variable X, which are

$$\mathbb{E}(X^1) = \mu \tag{6.19}$$
$$\mathbb{E}(X^2) = \mu^2 + \sigma^2 \tag{6.20}$$
$$\mathbb{E}(X^3) = \mu^3 + 3\mu\sigma^2 + \frac{1}{2}(4 - \pi)\lambda^3 \tag{6.21}$$

Using equations eqs. (2.27), (2.28) and (2.30), the variance and the skewness of X is

$$\mathrm{Var}(X) = \sigma^2 \tag{6.22}$$
$$\mathrm{Sk}(X) = \frac{(4 - \pi)\lambda^3}{2\sigma^3}. \tag{6.23}$$

If $\mathbb{E}(X)$, $\mathrm{Var}(X)$ and $\mathrm{Sk}(X)$ of X are known, then the above equations can be solved for σ and λ, which is called the *method of moments* (*moment matching*) approach. The result

is

$$\mu = \mathbb{E}(X) \tag{6.24}$$

$$\sigma = \sqrt{\mathrm{Var}(X)} \tag{6.25}$$

$$\lambda = \mathrm{sign}(\mathrm{Sk}(X)) \sqrt[3]{\frac{2|\mathrm{Sk}(X)|}{4-\pi}} \sigma, \tag{6.26}$$

where

$$\mathrm{sign}(x) = \begin{cases} -1 & \text{if } x < 0, \\ 0 & \text{if } x = 0, \\ 1 & \text{if } x > 0. \end{cases} \tag{6.27}$$

denotes the sign function (signum function).

Multivariate Skew-Normal Distribution

The multivariate skew-normal distribution extends the univariate skew-normal distribution to more than one dimension.

Definition 6.2. A n-dimensional random vector $\boldsymbol{X}$ is called *skew-normal random vector* with mean vector $\boldsymbol{\mu}$, covariance matrix Σ and shape vector $\boldsymbol{\lambda}$, if

$$\boldsymbol{\lambda}^T \Sigma^{-1} \boldsymbol{\lambda} < \frac{2}{\pi - 2} \tag{6.28}$$

and the probability density function of $\boldsymbol{X}$ is

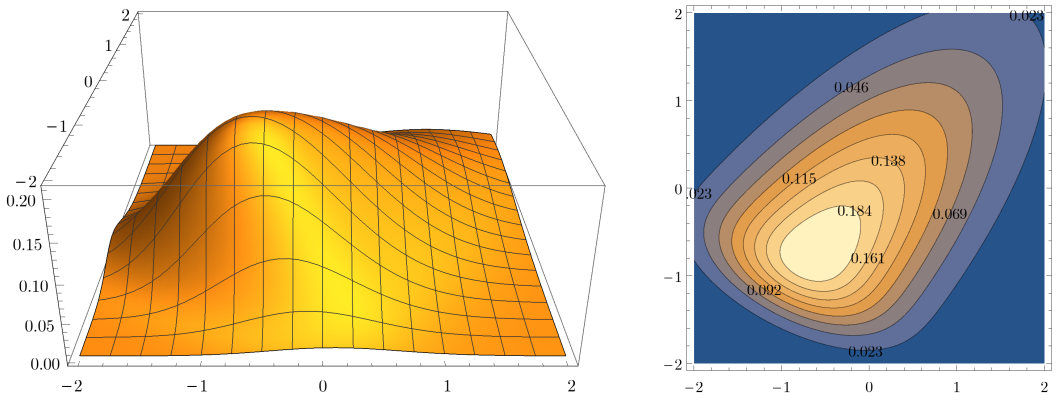
$$f_n(\boldsymbol{x}) = 2\phi_n\left(\boldsymbol{x}; \boldsymbol{\mu} - \boldsymbol{\lambda}, \Sigma + \boldsymbol{\lambda}\boldsymbol{\lambda}^T\right) \Phi\left(\frac{\boldsymbol{\lambda}^T \Sigma^{-1} (\boldsymbol{x} - \boldsymbol{\mu} + \boldsymbol{\lambda})}{\sqrt{(1+c)\left((2/\pi)(1+c) - c\right)}}\right), \tag{6.29}$$

where $b = \sqrt{2/\pi}$ and $c = \boldsymbol{\lambda}^T \Sigma^{-1} \boldsymbol{\lambda}$. The distribution of $\boldsymbol{X}$ is then denoted by $\mathcal{SN}_n(\boldsymbol{\mu}, \Sigma, \boldsymbol{\lambda})$.

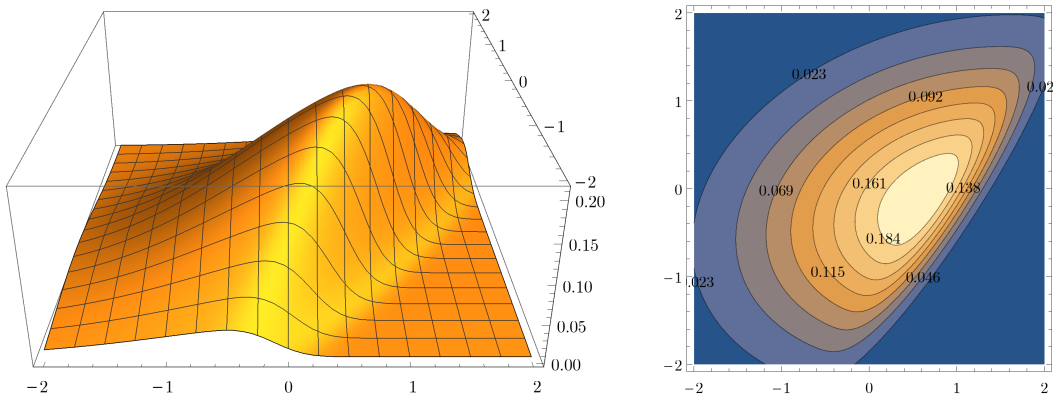
The 3D surface plot and contour plot of the probability density function of the bivariate skew-normal distribution with mean vector $\boldsymbol{\mu} = (0, 0)^T$, covariance matrix

$$\Sigma = \begin{pmatrix} 1 & 0.5 \\ 0.5 & 1 \end{pmatrix} \tag{6.30}$$

and different shape vectors are shown in fig. 6.2. The reader can compare the shape of the PDF to the shape of the PDF of a bivariate normal distribution in fig. 2.4, which has the same mean vector and the same covariance matrix.



(a) shape vector $\boldsymbol{\lambda} = (1.0, 1.2)^T$



(b) shape vector $\boldsymbol{\lambda} = (-1.0, 0.2)^T$

▲ **Figure 6.2 —** Probability density function of the bivariate skew-normal distribution

Let $\boldsymbol{t} \in \mathbb{R}^n$, then the moment generating function $M(\boldsymbol{t}) = \mathbb{E}(\exp(\boldsymbol{t}^T \boldsymbol{X}))$ of the skew-normal random vector $\boldsymbol{X} \sim \mathcal{SN}_n(\boldsymbol{\mu}, \Sigma, \boldsymbol{\lambda})$ is [Azzal13]

$$M(\boldsymbol{t}) = 2\exp\left(\boldsymbol{t}^T(\boldsymbol{\mu} - \boldsymbol{\lambda}) + \frac{1}{2}\boldsymbol{t}^T(\Sigma + \boldsymbol{\lambda}\boldsymbol{\lambda}^T)\boldsymbol{t}\right)\Phi\left(\boldsymbol{\lambda}^T\boldsymbol{t}/b\right). \tag{6.31}$$

Differentiating $M(\boldsymbol{t})$ with respect to t_i, t_j and t_k with $1 \leq i, j, k \leq n$ and setting $\boldsymbol{t} = \boldsymbol{0}$ yields

$$\mathbb{E}(X_i) = \mu_i \tag{6.32}$$
$$\mathbb{E}(X_i X_j) = \sigma_{i,j} + \mu_i \mu_j \tag{6.33}$$
$$\mathbb{E}(X_i^2 X_j) = 2\mu_i \sigma_{i,j} + \mu_j(\mu_i^2 + \sigma_i^2) + (4 - \pi)\lambda_i^2 \lambda_j / 2 \tag{6.34}$$
$$\mathbb{E}(X_i X_j^2) = \mu_i(\mu_j^2 + \sigma_j^2) + 2\mu_j \sigma_{i,j} + (4 - \pi)\lambda_i \lambda_j^2 / 2 \tag{6.35}$$
$$\mathbb{E}(X_i X_j X_k) = \mu_i(\mu_j \mu_k + \sigma_{j,k}) + \mu_j \sigma_{i,k} + \mu_k \sigma_{i,j} + (4 - \pi)\lambda_i \lambda_j \lambda_k / 2. \tag{6.36}$$

The reader should note the simplicity introduced by the alternative parameterization by comparing the above equations to the equivalent eqs. (6.9) to (6.13).

The third order moments (eqs. (6.34) to (6.36)) can also be expressed using matrix notation as

$$\mathbb{E}(\boldsymbol{X} \otimes \boldsymbol{X} \otimes \boldsymbol{X}^T) = \boldsymbol{\mu} \otimes \Sigma + \Sigma^V \otimes \boldsymbol{\mu}^T + \Sigma \otimes \boldsymbol{\mu} + \boldsymbol{\mu} \otimes \boldsymbol{\mu} \otimes \boldsymbol{\mu}^T + \left(2 - \frac{\pi}{2}\right) \boldsymbol{\lambda} \otimes \boldsymbol{\lambda}^T \otimes \boldsymbol{\lambda}, \tag{6.37}$$

where Σ^V denotes the vector obtained by stacking the columns of the matrix Σ on top of each other [Franc10]. Theorem 6.3 implies that $(\boldsymbol{X} - \mathbb{E}(\boldsymbol{X})) \sim \mathcal{SN}_n(\boldsymbol{0}, \Sigma, \boldsymbol{\lambda})$, so that the third multivariate cumulant $\kappa_3(\boldsymbol{X})$ of $\boldsymbol{X}$ (see definition 2.40) can be obtained from eq. (6.37) with $\boldsymbol{\mu} := \boldsymbol{0}$, which yields

$$\kappa_3(\boldsymbol{X}) = \left(2 - \frac{\pi}{2}\right) \boldsymbol{\lambda} \otimes \boldsymbol{\lambda}^T \otimes \boldsymbol{\lambda}. \tag{6.38}$$

6.1.3 Equivalence of Parametrizations

This subsection presents the proof that the alternative parameterization of the skew-normal distribution is equivalent to the classical definition by [Azzal13].

Univariate Skew-Normal Distribution

For any valid choice for the (μ, σ, λ) parameterization used in this work, a feasible (ξ, ω, α) pair can be found using the equations

$$\xi = \mu - \lambda \tag{6.39}$$

$$\omega = \sqrt{\sigma^2 + \lambda^2} \tag{6.40}$$

$$\alpha = \delta / \sqrt{1 - \delta^2}, \tag{6.41}$$

where $\delta = \lambda / (b\omega)$. For this, it must be shown that

$$\delta^2 = \frac{\lambda^2}{b^2(\sigma^2 + \lambda^2)} < 1 \tag{6.42}$$

which is satisfied if and only if

$$\lambda^2 < \frac{b^2\sigma^2}{1 - b^2}. \tag{6.43}$$

It can be seen that the above inequality is equivalent to eq. (6.16). Furthermore, for any valid choice of (ξ, ω, α), a feasible (μ, σ, λ) pair can be found using the equations

$$\mu = \xi + \lambda \tag{6.44}$$

$$\sigma = \sqrt{\omega^2 - \lambda^2} \tag{6.45}$$

$$\lambda = b\omega\delta, \tag{6.46}$$

where $\delta = \alpha / \sqrt{1 + \alpha^2}$ [Azzal13, eq.(2.6)]. Hence, the parametrizations and the probability density functions eq. (6.1) and eq. (6.17) are indeed equivalent.

Multivariate Skew-Normal Distribution

In this section, it is shown that the $(\boldsymbol{\mu}, \Sigma, \boldsymbol{\lambda})$ parametrization adopted in this work is equivalent to the $(\boldsymbol{\xi}, \Omega, \boldsymbol{\alpha})$ parametrization of [Azzal13].

Specifically, it is shown that for any valid choice of $(\boldsymbol{\xi}, \Omega, \boldsymbol{\alpha})$, a feasible $(\boldsymbol{\mu}, \Sigma, \boldsymbol{\lambda})$ pair exits and can be computed as

$$\boldsymbol{\delta} = \frac{\bar{\Omega}\boldsymbol{\alpha}}{\sqrt{1 + \boldsymbol{\alpha}^T \bar{\Omega} \boldsymbol{\alpha}}} \tag{6.47}$$

$$\boldsymbol{\lambda} = b\omega\boldsymbol{\delta}, \tag{6.48}$$

where eq. (6.47) is taken from [Azzal13, eq. (5.11)], $\bar{\Omega} = \omega^{-1}\Omega\omega^{-1}$ is the correlation matrix corresponding to Ω and ω is defined by eq. (6.6). Clearly, $\boldsymbol{\alpha}^T\bar{\Omega}\boldsymbol{\alpha} > 0$ because $\bar{\Omega}$ is a positive definite matrix according to theorem 2.4. Plugging eq. (6.48) into eqs. (6.14) and (6.15) yields

$$\boldsymbol{\mu} = \boldsymbol{\xi} + \boldsymbol{\lambda} \tag{6.49}$$

$$\Sigma = \Omega - \boldsymbol{\lambda}\boldsymbol{\lambda}^T. \tag{6.50}$$

It remains to be shown, that eq. (6.50) is a symmetric positive definite matrix. By using theorem 2.4 and eq. (6.47), it is sufficient to show that

$$\omega^{-1}(\Omega - \boldsymbol{\lambda}\boldsymbol{\lambda}^T)\omega^{-1} = \bar{\Omega} - b^2\boldsymbol{\delta}\boldsymbol{\delta}^T \tag{6.51}$$

$$= \bar{\Omega} - b^2\bar{\Omega}\boldsymbol{\alpha}(1 + \boldsymbol{\alpha}^T\bar{\Omega}\boldsymbol{\alpha})^{-1}\boldsymbol{\alpha}^T\bar{\Omega} \tag{6.52}$$

is symmetric positive definite. This is done by using eq. (2.5) with $A^{-1} = \bar{\Omega}$ and $U = V = b\boldsymbol{\alpha}$ so that eq. (6.52) becomes $(\bar{\Omega}^{-1} + b^2\boldsymbol{\alpha}\boldsymbol{\alpha}^T)^{-1}$, which is positive definite according to lemma 2.3 and theorem 2.4. It is also symmetric because the sum of symmetric matrices is a symmetric matrix. Therefore, $\omega^{-1}(\Omega - \boldsymbol{\lambda}\boldsymbol{\lambda}^T)\omega^{-1}$ is symmetric positive definite and because ω is a non-singular matrix, theorem 2.4 implies that $\Omega - \boldsymbol{\lambda}\boldsymbol{\lambda}^T$ is also symmetric positive semidefinite.

Next, for any valid assignment to the parameters $\boldsymbol{\mu}$, Σ and $\boldsymbol{\lambda}$, the corresponding values of the $(\boldsymbol{\xi}, \Omega, \boldsymbol{\alpha})$ parametrization are

$$\boldsymbol{\xi} = \boldsymbol{\mu} - \boldsymbol{\lambda} \tag{6.53}$$

$$\Omega = \Sigma + \boldsymbol{\lambda}\boldsymbol{\lambda}^T. \tag{6.54}$$

The equation for $\boldsymbol{\alpha}$ is derived from [Azzal13, eq. (5.12)]

$$\boldsymbol{\alpha} = \frac{\bar{\Omega}^{-1}\boldsymbol{\delta}}{\sqrt{1 - \boldsymbol{\delta}^T\bar{\Omega}^{-1}\boldsymbol{\delta}}} \tag{6.55}$$

by using eq. (6.48) to obtain the intermediate result

$$\boldsymbol{\alpha} = \frac{\omega\Omega^{-1}\omega\boldsymbol{\delta}}{\sqrt{1 - \boldsymbol{\delta}^T(\omega\Omega^{-1}\omega)\boldsymbol{\delta}}} = \frac{\omega\Omega^{-1}\boldsymbol{\lambda}b^{-1}}{\sqrt{1 - b^{-2}\boldsymbol{\lambda}^T\Omega^{-1}\boldsymbol{\lambda}}} = \frac{\omega\Omega^{-1}\boldsymbol{\lambda}}{\sqrt{b^2 - \boldsymbol{\lambda}^T\Omega^{-1}\boldsymbol{\lambda}}}. \tag{6.56}$$

The application of eq. (2.5) with $U = V = \boldsymbol{\lambda}$ to eq. (6.54) gives

$$\Omega^{-1} = \Sigma^{-1} - \frac{1}{1 + \boldsymbol{\lambda}^T \Sigma^{-1} \boldsymbol{\lambda}} \Sigma^{-1} \boldsymbol{\lambda} \boldsymbol{\lambda}^T \Sigma^{-1} \tag{6.57}$$

so that

$$\Omega^{-1} \boldsymbol{\lambda} = \frac{\Sigma^{-1} \boldsymbol{\lambda} (1 + \boldsymbol{\lambda}^T \Sigma^{-1} \boldsymbol{\lambda})}{1 + \boldsymbol{\lambda}^T \Sigma^{-1} \boldsymbol{\lambda}} - \frac{\Sigma^{-1} \boldsymbol{\lambda} (\boldsymbol{\lambda}^T \Sigma^{-1} \boldsymbol{\lambda})}{1 + \boldsymbol{\lambda}^T \Sigma^{-1} \boldsymbol{\lambda}} = \frac{\Sigma^{-1} \boldsymbol{\lambda}}{1 + \boldsymbol{\lambda}^T \Sigma^{-1} \boldsymbol{\lambda}} \tag{6.58}$$

must hold. Plugging the last equation into eq. (6.56) finally yields

$$\boldsymbol{\alpha} = \frac{\omega \Sigma^{-1} \boldsymbol{\lambda}}{\sqrt{\frac{b^2(1+c)-c}{1+c}}(1+c)} = \frac{\omega \Sigma^{-1} \boldsymbol{\lambda}}{\sqrt{(1+c)(b^2(1+c)-c)}}, \tag{6.59}$$

where $c := \boldsymbol{\lambda}^T \Sigma^{-1} \boldsymbol{\lambda}$ and ω is defined by eq. (6.6). Hence, it remains to be shown that

$$(1+c)(b^2(1+c)-c) > 0 \tag{6.60}$$

for any valid choice of Σ and $\boldsymbol{\lambda}$. It can be seen that eq. (6.60) is a quadratic polynomial with exactly two roots: one at $c = -1$ and the other at $c = 2/(\pi - 2)$. Furthermore, the LHS is equal to b^2 for $c = 0$, which confirms that the above inequality is true for all $-1 < c < 2/(\pi - 2)$. It is also clear that $c > 0$ because Σ^{-1} is positive definite and by definition $c < 2/(\pi - 2)$. Hence, the parametrizations and the probability density functions eq. (6.7) and eq. (6.29) are indeed equivalent.

6.2 Statistical SUM-operation

This section describes and presents the algorithm for the skew-normal distribution based SUM-operation. The operation is described for the general case that k sums are computed given n random variables $X_1, \ldots, X_n$, e.g. to compute the propagation delays of k paths given the propagation delays of n gate. The problem of computing the arrival time of the gate output transition given the arrival time of a gate input transition and the propagation delay of the gate is considered as the special case $k = 1$ and $n = 2$.

Let $\boldsymbol{X} = (X_1, \ldots, X_n)^T$ denote a n-dimensional skew-normal random vector. Let $B \in \{0,1\}^{k \times n}$ be a binary matrix of rank k and let $b_{i,j} \in \{0,1\}$ denote the element in the ith row and jth column of B with $1 \leq i \leq k$ and $1 \leq j \leq n$. Then the *skew-normal distribution based SUM-operation* computes a k-dimensional random vector $\boldsymbol{Y} = (Y_1, \ldots, Y_k)^T$, defined as

$$Y_i = \sum_{j=1}^{n} b_{i,j} X_j \quad \text{for all } 1 \leq i \leq k,$$

where $b_{i,j} = 1$ if and only if X_j is part of the ith sum Y_i, otherwise $b_{i,j} = 0$.

Because the family of multivariate skew-normal distributions is closed under affine transformations, the skew-normal distribution based SUM-operation can be efficiently computed by $\boldsymbol{Y} = B\boldsymbol{X}$ using the following theorem.

Theorem 6.3. *If $\boldsymbol{X} \sim \mathcal{SN}_n(\boldsymbol{\mu}, \Sigma, \boldsymbol{\lambda})$, $\boldsymbol{b} \in \mathbb{R}^k$ and $A \in \mathbb{R}^{k \times n}$ has rank k, then the random vector*

$$\boldsymbol{Y} = A\boldsymbol{X} + \boldsymbol{b} \tag{6.61}$$

has a k-dimensional skew-normal distribution $\mathcal{SN}_k(\boldsymbol{\mu_Y}, \Sigma_{\boldsymbol{Y}}, \boldsymbol{\lambda_Y})$ with parameters

$$\boldsymbol{\mu_Y} = A\boldsymbol{\mu} + \boldsymbol{b} \tag{6.62}$$

$$\Sigma_{\boldsymbol{Y}} = A\Sigma A^T \tag{6.63}$$

$$\boldsymbol{\lambda_Y} = A\boldsymbol{\lambda}. \tag{6.64}$$

The proof is presented in section A.2.1. The reader should note the similarities to the affine transformation of a random vector with multivariate normal distribution in theorem 2.47.

The application of the skew-normal distribution based SUM-operation is illustrated by two examples. The first example is a NAND gate with a falling transition at each input. Let X_1 and X_2 denote the transition arrival time at input A and B, respectively. Furthermore, let X_3 and X_4 denote the propagation delay for the transition at input A and B, respectively. To compute the sums $Y_1 = X_1 + X_3$ and $Y_2 = X_2 + X_4$, the matrix B is defined as

$$B = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix}. \tag{6.65}$$

Another example is the computation of the propagation delay of a path given the propagation delays of n gates. Let

$$A = (1, \ldots, 1) \in \mathbb{R}^{1 \times n} \tag{6.66}$$

denote a n-dimensional row vector and let $\boldsymbol{b} = \boldsymbol{0}$ be the zero vector. Let $\sigma_{i,j}$ denote the element in the ith row and jth column of Σ, then theorem 6.3 implies that the path delay is a skew-normal random variable $Y \sim \mathcal{SN}(\mu_Y, \sigma_Y^2, \lambda_Y)$ with parameters

$$\mu_Y = \sum_{i=1}^{n} \mu_i \tag{6.67}$$

$$\sigma_Y^2 = \sum_{i=1}^{n} \sum_{j=1}^{n} \sigma_{i,j} \tag{6.68}$$

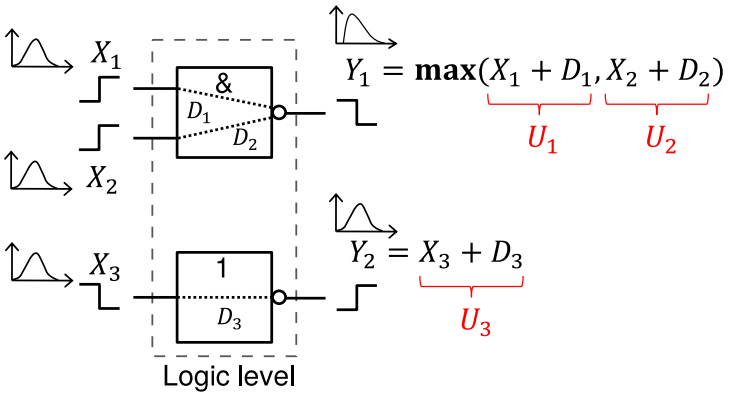
$$\lambda_Y = \sum_{i=1}^{n} \lambda_i. \tag{6.69}$$

6.3 Statistical MAX-Operation

This sections describes and presents a basic algorithm for the skew-normal distribution based MAX-operation. The application of the MAX-operation is illustrated with an

example in fig. 6.3, which shows two gates in a logic level. A NAND gate with two rising transitions at the gate inputs and an inverter with a falling transition at the gate input. The arrival times of the input transitions are denoted by X_1, X_2 and X_3. The corresponding propagation delays of the gates are denoted by D_1, D_2 and D_3. For the inverter, the arrival time of the output transition is simply the sum of the arrival time of the input transition and the corresponding propagation delay of the gate. For the NAND gate, the arrival time of the last output transition is computed by adding the corresponding propagation delays of the gate to the arrival times of the input transitions followed by the computation of the statistical maximum of both results.

For simplicity, let $U_i := X_i + D_i$ for all $1 \leq i \leq 3$. To compute the arrival time of the last transitions at the outputs of the logic level, the statistical maximum operation must compute the distribution of $\max(U_1, U_2)$, but also the relationship (covariance) between $\max(U_1, U_2)$ and U_3.



▲ **Figure 6.3 —** Example for the application of the MAX-operation in a logic level

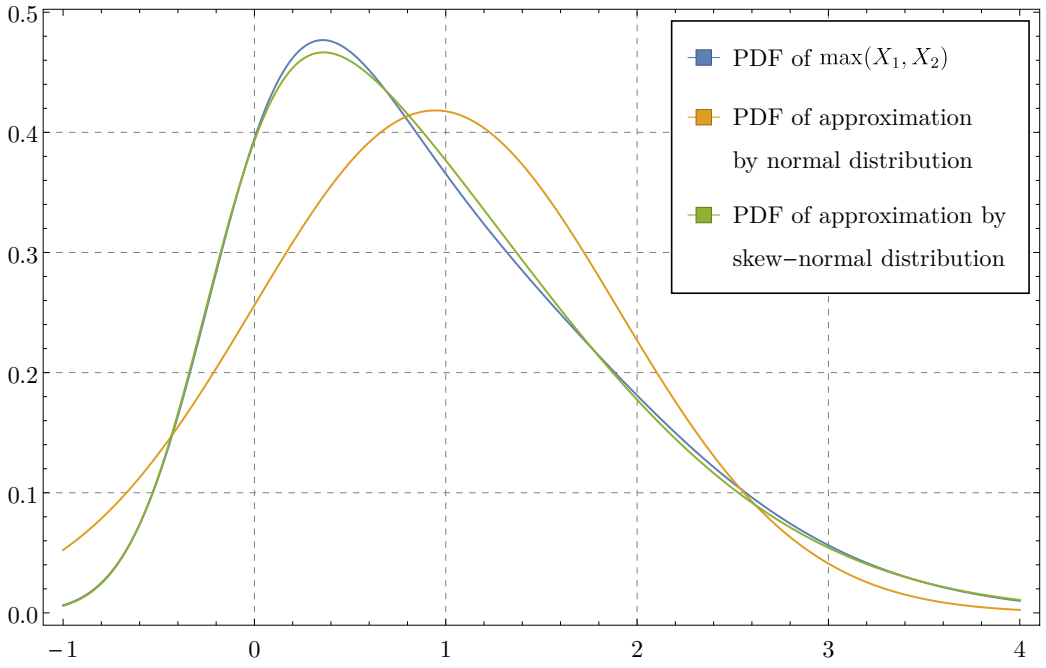
To minimize the approximation error, the MAX-operation must be defined on a flexible, general family of probability distributions which allows the accurate approximation of $\max(X_1, X_2)$ with a member of the same family of distributions. Compared to the normal distribution based MAX-operation in section 3.3.2, the maximum $\max(X_1, X_2)$ can be much more accurately approximated by a skew-normal distribution, as shown in fig. 6.4.

More generally, the skew-normal distribution based MAX-operation is described as follows.

Let $(X_1, \ldots, X_n)^T \sim \mathcal{SN}_n(\boldsymbol{\mu}, \Sigma, \boldsymbol{\lambda})$ denote a n-dimensional skew-normal random vector. The *approximation* of the $(n-1)$-dimensional distribution of the random vector

$$\boldsymbol{Y} = (Y_1, \ldots, Y_{n-1})^T := (X_1, \ldots, X_{n-2}, \max(X_{n-1}, X_n))^T \tag{6.70}$$

by a skew-normal distribution $\mathcal{SN}_{n-1}(\hat{\boldsymbol{\mu}}, \hat{\Sigma}, \hat{\boldsymbol{\lambda}})$ with mean vector $\hat{\boldsymbol{\mu}}$, covariance matrix $\hat{\Sigma}$ and shape vector $\hat{\boldsymbol{\lambda}}$ is called *skew-normal distribution based MAX-operation*.



▲ **Figure 6.4 —** Probability density functions of $\max(X_1, X_2)$ and its approximation by a normal distribution and a skew-normal distribution

The computation of the parameters $\hat{\boldsymbol{\mu}}$, $\hat{\Sigma}$ and $\hat{\boldsymbol{\lambda}}$ is explained in the following three subsections. At first, the mean vector and the covariance matrix of $\boldsymbol{Y}$ are computed. Afterwards, section 6.3.2 introduces the third multivariate cumulant $\boldsymbol{\gamma}$, which will be used to estimate the shape vector $\hat{\boldsymbol{\lambda}}$ in section 6.3.3.

6.3.1 Computation of Mean Vector $\hat{\mu}$ and Covariance Matrix $\hat{\Sigma}$

Let $\boldsymbol{Y}$ be a $(n-1)$-dimensional random vector as defined in eq. (6.70). The first two parameters of the skew-normal distribution $\mathcal{SN}_{n-1}(\hat{\boldsymbol{\mu}}, \hat{\Sigma}, \hat{\boldsymbol{\lambda}})$ are approximated by the mean vector and the covariance matrix of $\boldsymbol{Y}$. Explicit formulas for the computation of all required moments can be found in section 6.1.2 and in appendix A.1.

According to definition 2.36, the mean vector of $\boldsymbol{Y}$ is

$$\hat{\boldsymbol{\mu}} = (\mathbb{E}(Y_1), \mathbb{E}(Y_2), \ldots, \mathbb{E}(Y_{n-1}))^T \tag{6.71}$$

and the covariance matrix of $\boldsymbol{Y}$ is

$$\hat{\Sigma} = \begin{pmatrix} \mathbb{E}(\bar{Y}_1^2) & \mathbb{E}(\bar{Y}_1\bar{Y}_2) & \cdots & \mathbb{E}(\bar{Y}_1\bar{Y}_{n-1}) \\ \mathbb{E}(\bar{Y}_2\bar{Y}_1) & \mathbb{E}(\bar{Y}_2^2) & \cdots & \mathbb{E}(\bar{Y}_2\bar{Y}_{n-1}) \\ \vdots & \vdots & \ddots & \vdots \\ \mathbb{E}(\bar{Y}_{n-1}\bar{Y}_1) & \mathbb{E}(\bar{Y}_{n-1}\bar{Y}_2) & \cdots & \mathbb{E}(\bar{Y}_{n-1}^2) \end{pmatrix}, \tag{6.72}$$

where $\bar{Y}_1, \ldots, \bar{Y}_{n-1}$ denote the components of the *centred random vector*, defined as

$$\bar{\boldsymbol{Y}} = \boldsymbol{Y} - \hat{\boldsymbol{\mu}} = (Y_1 - \mathbb{E}(Y_1), \ldots, Y_{n-1} - \mathbb{E}(Y_{n-1}))^T. \tag{6.73}$$

For example, suppose $n = 4$ then the 3-dimensional random vector $\boldsymbol{Y}$ has mean vector

$$\hat{\boldsymbol{\mu}} = \mathbb{E}(\boldsymbol{Y}) = (\mathbb{E}(Y_1), \mathbb{E}(Y_2), \mathbb{E}(Y_3))^T$$

and covariance matrix

$$\hat{\Sigma} = \mathrm{Var}(\boldsymbol{Y}) = \begin{pmatrix} \mathbb{E}(\bar{Y}_1^2) & \mathbb{E}(\bar{Y}_1\bar{Y}_2) & \mathbb{E}(\bar{Y}_1\bar{Y}_3) \\ \mathbb{E}(\bar{Y}_1\bar{Y}_2) & \mathbb{E}(\bar{Y}_2^2) & \mathbb{E}(\bar{Y}_2\bar{Y}_3) \\ \mathbb{E}(\bar{Y}_1\bar{Y}_3) & \mathbb{E}(\bar{Y}_2\bar{Y}_3) & \mathbb{E}(\bar{Y}_3^2) \end{pmatrix}.$$

Only those moments which depend on

$$Y_{n-1} = \max(X_{n-1}, X_n)$$

like $\mathbb{E}(\bar{Y}_{n-1})$ or $\mathbb{E}(\bar{Y}_1\bar{Y}_{n-1})$ have to be computed. All other moments remain unchanged and can be copied from $\boldsymbol{\mu}$ and Σ. More precisely, $\hat{\boldsymbol{\mu}}$ is obtained by deleting the last and replacing the second last component of $\boldsymbol{\mu}$. Likewise, $\hat{\Sigma}$ is obtained by deleting the last and replacing the second last row and column of Σ. The in-place computation of the mean vector $\hat{\boldsymbol{\mu}}$ and the covariance matrix $\hat{\Sigma}$ is done in $O(n)$ time.

6.3.2 Computation and Properties of Third Multivariate Cumulant

The skewness of a single random variable X was introduced by eq. (2.30) and it depends on the third central moment $\mathbb{E}((X - \mu_X)^3)$. A multivariate analogue of the third central moment of a random variable is the *third multivariate cumulant* $\kappa_3(\boldsymbol{Y}) \in \mathbb{R}^{(n-1)^2 \times (n-1)}$ of a random vector $\boldsymbol{Y}$, which is defined by definition 2.40 and can be written as

$$\kappa_3(\boldsymbol{Y}) = \mathbb{E}(\bar{\boldsymbol{Y}} \otimes \bar{\boldsymbol{Y}} \otimes \bar{\boldsymbol{Y}}^T), \tag{6.74}$$

where $\otimes$ denotes the Kronecker product. The reader should note that $\kappa_3(\boldsymbol{Y})$ contains all third order central moments $\mathbb{E}(\bar{Y}_i\bar{Y}_j\bar{Y}_k)$ with $1 \leq i, j, k \leq n-1$ conveniently arranged [Mori94, Franc10, Luca15]. For example, suppose $n = 4$ then the third multivariate cumulant of the 3-dimensional random vector $\boldsymbol{Y}$ is

$$\kappa_3(\boldsymbol{Y}) = \begin{pmatrix} \mathbb{E}(\bar{Y}_1^3) & \mathbb{E}(\bar{Y}_1^2\bar{Y}_2) & \mathbb{E}(\bar{Y}_1^2\bar{Y}_3) \\ \mathbb{E}(\bar{Y}_1^2\bar{Y}_2) & \mathbb{E}(\bar{Y}_1\bar{Y}_2^2) & \mathbb{E}(\bar{Y}_1\bar{Y}_2\bar{Y}_3) \\ \mathbb{E}(\bar{Y}_1^2\bar{Y}_3) & \mathbb{E}(\bar{Y}_1\bar{Y}_2\bar{Y}_3) & \mathbb{E}(\bar{Y}_1\bar{Y}_3^2) \\ \mathbb{E}(\bar{Y}_1^2\bar{Y}_2) & \mathbb{E}(\bar{Y}_1\bar{Y}_2^2) & \mathbb{E}(\bar{Y}_1\bar{Y}_2\bar{Y}_3) \\ \mathbb{E}(\bar{Y}_1\bar{Y}_2^2) & \mathbb{E}(\bar{Y}_2^3) & \mathbb{E}(\bar{Y}_2^2\bar{Y}_3) \\ \mathbb{E}(\bar{Y}_1\bar{Y}_2\bar{Y}_3) & \mathbb{E}(\bar{Y}_2^2\bar{Y}_3) & \mathbb{E}(\bar{Y}_2\bar{Y}_3^2) \\ \mathbb{E}(\bar{Y}_1^2\bar{Y}_3) & \mathbb{E}(\bar{Y}_1\bar{Y}_2\bar{Y}_3) & \mathbb{E}(\bar{Y}_1\bar{Y}_3^2) \\ \mathbb{E}(\bar{Y}_1\bar{Y}_2\bar{Y}_3) & \mathbb{E}(\bar{Y}_2^2\bar{Y}_3) & \mathbb{E}(\bar{Y}_2\bar{Y}_3^2) \\ \mathbb{E}(\bar{Y}_1\bar{Y}_3^2) & \mathbb{E}(\bar{Y}_2\bar{Y}_3^2) & \mathbb{E}(\bar{Y}_3^3) \end{pmatrix}.$$

For any non-singular matrix $A \in \mathbb{R}^{(n-1)\times(n-1)}$, theorem 2.37 has already established the relationship between the mean vectors and between the covariance matrices of the

random vectors $\boldsymbol{Y}$ and $A\boldsymbol{Y}$. A similar relationship holds between the third multivariate cumulant of $\boldsymbol{Y}$ and the third multivariate cumulant of $A\boldsymbol{Y}$ [Kollo05, p.190], as shown by the following theorem.

Theorem 6.4. *If $\boldsymbol{X}$ is a n-dimensional random vector with finite third multivariate cumulant $\kappa_3(\boldsymbol{X})$ and $A \in \mathbb{R}^{k\times n}$ has rank k, then the third multivariate cumulant $\kappa_3(\boldsymbol{Y}) \in \mathbb{R}^{k^2\times k}$ of the random vector $\boldsymbol{Y} = A\boldsymbol{X}$ is*

$$\kappa_3(\boldsymbol{Y}) = (A \otimes A)\kappa_3(\boldsymbol{X})A^T, \tag{6.75}$$

where $\otimes$ denotes the Kronecker product.

This property is particularly useful to simplify the following computations by considering only the standardized version $\boldsymbol{Z}$ of the random vector $\boldsymbol{Y}$. In eq. (2.30), the random variable X is being standardized to zero mean and unit variance by substracting its mean and dividing by its standard deviation. Similarly, the components of the standardized random vector $\boldsymbol{Z}$ are standardized to means of zero, standard deviations of one and all pairwise covariances are zero. The standardized random vector $\boldsymbol{Z}$ can be computed from the random vector $\boldsymbol{Y}$ by the linear transformation

$$\boldsymbol{Z} = \hat{L}^{-1}(\boldsymbol{Y} - \hat{\boldsymbol{\mu}}) = \hat{L}^{-1}\bar{\boldsymbol{Y}}, \tag{6.76}$$

where $\hat{L}^{-1}$ denotes the inverse of the lower Cholesky factor of the covariance matrix $\hat{\Sigma}$ of $\boldsymbol{Y}$, such that $\hat{L}\hat{L}^T = \hat{\Sigma}$. The covariance matrix $\hat{\Sigma}$ of $\boldsymbol{Y}$ is non-singular by assumption. To show this, theorem 2.37 is applied to compute the mean vector and the covariance matrix of $\boldsymbol{Z}$, which gives

$$\mathbb{E}(\boldsymbol{Z}) = \hat{L}^{-1}\boldsymbol{0} = \boldsymbol{0} \tag{6.77}$$

$$\mathrm{Var}(\boldsymbol{Z}) = \hat{L}^{-1}\hat{\Sigma}\hat{L}^{-T} = \hat{L}^{-1}\hat{L}\hat{L}^T\hat{L}^{-T} = I_{n-1}. \tag{6.78}$$

The efficient computation of $\hat{L}^{-1}$ is explained in section 6.4.

By definition, the third multivariate cumulant of a random vector is computed from the corresponding centred random vector, as defined by eq. (6.73). Therefore, $\boldsymbol{Y}$ and $\bar{\boldsymbol{Y}}$ share the same third multivariate cumulant so that theorem 6.4 can be applied to compute the third multivariate cumulant of the standardized random vector $\boldsymbol{Z} = \hat{L}^{-1}\bar{\boldsymbol{Y}}$, which yields

$$\kappa_3(\boldsymbol{Z}) = (\hat{L}^{-1} \otimes \hat{L}^{-1})\kappa_3(\boldsymbol{Y})\hat{L}^{-T}. \tag{6.79}$$

One of the interesting properties of this matrix is that the skewness $\mathrm{Sk}(Y_1)$ of the first component Y_1 of $\boldsymbol{Y}$ is in the top left corner of $\kappa_3(\boldsymbol{Z})$. This is because theorem 2.6 implies that the standard deviation σ_{Y_1} of Y_1 is in the top-left corner of $\hat{L}$. Therefore, the top-left corner of $\hat{L}^{-1}$ is $1/\sigma_{Y_1}$ (because $\hat{L}\hat{L}^{-1} = I_n$) and the above statement follows from eqs. (2.30) and (6.79).

In the following, $\kappa_3(\boldsymbol{Z})$ is called the *third standardized multivariate cumulant* of $\boldsymbol{Y}$. The third standardized multivariate cumulant of the random vector $\boldsymbol{Y}$ is of central importance for the estimation of the shape vector $\hat{\boldsymbol{\lambda}}$ in the next section.

6.3.3 Estimation of Shape Vector $\hat{\lambda}$

For a given skew-normal random variable X with known standard deviation σ_X and skewness $\mathrm{Sk}(X)$. The shape parameter $\hat{\boldsymbol{\lambda}}$ can be computed using eq. (6.26). This idea can be generalized to compute the shape vector of a skew-normal random vector, as shown by the following theorem.

Theorem 6.5. *Let $\boldsymbol{X} \sim \mathcal{SN}_n(\boldsymbol{\mu}, \Sigma, \boldsymbol{\lambda})$ denote a n-dimensional skew-normal random vector. If $\kappa_3(\boldsymbol{V})$ is the third standardized multivariate cumulant of $\boldsymbol{X}$, then the matrix*

$$M_{\boldsymbol{X}} = \kappa_3(\boldsymbol{V})^T \kappa_3(\boldsymbol{V}) \tag{6.80}$$

has an eigenvector

$$\boldsymbol{v} = \pm \frac{L^{-1}\boldsymbol{\lambda}}{||L^{-1}\boldsymbol{\lambda}||} \tag{6.81}$$

corresponding to the only non-zero eigenvalue

$$\psi = \left(2 - \frac{\pi}{2}\right)^2 (\boldsymbol{\lambda}^T \Sigma^{-1} \boldsymbol{\lambda})^3 = \left(2 - \frac{\pi}{2}\right)^2 ||L^{-1}\boldsymbol{\lambda}||^6, \tag{6.82}$$

where L denotes the lower Cholesky factor of Σ such that $\Sigma = LL^T$.

The $\pm$ in eq. (6.81) is due to the fact that the sign of an eigenvector of a matrix is not unique, because if $M_{\boldsymbol{X}}\boldsymbol{v} = \psi\boldsymbol{v}$ holds then so does $M_{\boldsymbol{X}}(-\boldsymbol{v}) = \psi(-\boldsymbol{v})$. Hence, the above theorem can be used to compute the shape vector of $\boldsymbol{X}$, except for its sign.

Although the random vector $\boldsymbol{Y}$ doesn't have a multivariate skew-normal distribution, the above theorem can still be applied to *approximate* the distribution of $\boldsymbol{Y}$ with a multivariate skew-normal distribution. Let $\kappa_3(\boldsymbol{Z})$ denote the third standardized multivariate cumulant of the random vector $\boldsymbol{Y}$, then the matrix

$$M_{\boldsymbol{Y}} = \kappa_3(\boldsymbol{Z})^T \kappa_3(\boldsymbol{Z}) \tag{6.83}$$

is called the *skewness matrix* of $\boldsymbol{Y}$ [Loper13, p.3]. Let $\boldsymbol{v}$ denote the dominant eigenvector corresponding to the dominant eigenvalue ψ of the skewness matrix $M_{\boldsymbol{Y}}$. From eq. (6.81) it follows that

$$\hat{\boldsymbol{\lambda}} = \pm ||\hat{L}^{-1}\hat{\boldsymbol{\lambda}}|| \hat{L}\boldsymbol{v}. \tag{6.84}$$

Using eq. (6.82) and $\hat{\Sigma}^{-1} = \hat{L}^{-T}\hat{L}^{-1}$, the euclidean norm of the vector $\hat{L}^{-1}\hat{\boldsymbol{\lambda}}$ is

$$||\hat{L}^{-1}\hat{\boldsymbol{\lambda}}|| = \sqrt{(\hat{L}^{-1}\hat{\boldsymbol{\lambda}})^T(\hat{L}^{-1}\hat{\boldsymbol{\lambda}})} = \sqrt{\hat{\boldsymbol{\lambda}}^T\hat{\Sigma}^{-1}\hat{\boldsymbol{\lambda}}} = \sqrt[6]{\frac{4\psi}{(\pi-4)^2}}. \tag{6.85}$$

This norm is well defined because eq. (6.83) implies that $M_{\boldsymbol{Y}}$ is positive semidefinite so that $\psi \geq 0$ according to lemma 2.2. However, any valid shape vector estimate $\hat{\boldsymbol{\lambda}}$ must satisfy inequality eq. (6.28), which might be violated if the distribution of $\boldsymbol{Y}$ differs

strongly from a multivariate skew-normal distribution. From eqs. (6.28) and (6.85) it follows that

$$\sqrt[3]{\frac{4\psi}{(\pi-4)^2}} < \frac{2}{\pi-2} \tag{6.86}$$

so that

$$\psi < 2\frac{(\pi-4)^2}{(\pi-2)^3} \approx 0.990566 \tag{6.87}$$

must hold for eq. (6.85) to be valid. If eq. (6.87) is not satisfied, then ψ is set to 0.99.

The sign of $\hat{\boldsymbol{\lambda}} = (\hat{\lambda}_1, \ldots, \hat{\lambda}_{n-1})^T$ is determined by noticing that $\hat{\lambda}_{n-1}$ represents the shape parameter of $\max(X_{n-1}, X_n)$. If the skewness of $\max(X_{n-1}, X_n)$, which is given by eqs. (2.30) and (A.11) to (A.13), is positive then $\hat{\lambda}_{n-1}$ should be positive. Likewise, if the skewness of $\max(X_{n-1}, X_n)$ is negative, then $\hat{\lambda}_{n-1}$ should also be negative.

More precisely, let

$$x = \operatorname{sign}(\operatorname{Sk}(\max(X_{n-1}, X_n))) \tag{6.88}$$

$$y = \operatorname{sign}\left(\hat{l}_{n-1}^T \boldsymbol{v}\right), \tag{6.89}$$

where $\hat{l}_{n-1}^T$ denotes the last row of $\hat{L}$ and $\operatorname{sign}(\cdot)$ denotes the signum function as defined in eq. (6.27). Then the estimate of the shape vector is

$$\hat{\boldsymbol{\lambda}} = \begin{cases} -||\hat{L}^{-1}\hat{\boldsymbol{\lambda}}||\hat{L}\boldsymbol{v} & \text{if } xy < 0 \\ +||\hat{L}^{-1}\hat{\boldsymbol{\lambda}}||\hat{L}\boldsymbol{v} & \text{if } xy \geq 0, \end{cases} \tag{6.90}$$

where the euclidean norm $||\hat{L}^{-1}\hat{\boldsymbol{\lambda}}||$ is given by eq. (6.85).

6.3.4 A Numerical Example

Given a 4-dimensional skew-normal random vector $\boldsymbol{X} \sim \mathcal{SN}_4(\boldsymbol{\mu}, \Sigma, \boldsymbol{\lambda})$ with

$$\boldsymbol{\mu} = \begin{pmatrix} -0.1 \\ 0.45 \\ -0.2 \\ 0.31 \end{pmatrix}, \quad \Sigma = \begin{pmatrix} 0.479 & 0.528 & -0.494 & -0.428 \\ 0.528 & 1.088 & -1.199 & -0.661 \\ -0.494 & -1.199 & 1.624 & 0.536 \\ -0.428 & -0.661 & 0.536 & 0.969 \end{pmatrix}, \quad \boldsymbol{\lambda} = \begin{pmatrix} 0.169 \\ 0.115 \\ 0.023 \\ 0.172 \end{pmatrix}.$$

At first, the mean vector $\hat{\boldsymbol{\mu}}$, the covariance matrix $\hat{\Sigma}$ and the third multivariate cumulant $\kappa_3(\boldsymbol{X})$ of the random vector

$$\boldsymbol{Y} = (X_1, X_2, \max(X_3, X_4))^T$$

are computed. The results are

$$\hat{\boldsymbol{\mu}} \approx \begin{pmatrix} -0.1 \\ 0.45 \\ 0.5885 \end{pmatrix}, \quad \hat{\Sigma} \approx \begin{pmatrix} 0.479 & 0.528 & -0.4502 \\ 0.528 & 1.088 & -0.8436 \\ -0.4502 & -0.8436 & 0.9722 \end{pmatrix}$$

and

$$\kappa_3(\boldsymbol{Y}) \approx \begin{pmatrix} 0.0021 & 0.0014 & 0.0028 \\ 0.0014 & 0.001 & 0.0115 \\ 0.0028 & 0.0115 & -0.006 \\ 0.0014 & 0.001 & 0.0115 \\ 0.001 & 0.0007 & 0.0866 \\ 0.0115 & 0.0866 & -0.0587 \\ 0.0028 & 0.0115 & -0.006 \\ 0.0115 & 0.0866 & -0.0587 \\ -0.0060 & -0.0587 & 0.0768 \end{pmatrix}.$$

Note that the top-left corner 2×2 submatrix of Σ also appears in $\hat{\Sigma}$ and that the subvector $(-0.1, 0.45)^T$ of $\boldsymbol{\mu}$ also appears in $\hat{\boldsymbol{\mu}}$. The inverse of the lower Cholesky factor of $\hat{\Sigma}$ is

$$\hat{L}^{-1} \approx \begin{pmatrix} 1.4449 & 0 & 0 \\ -1.5496 & 1.4058 & 0 \\ 0.3286 & 1.2316 & 1.7942 \end{pmatrix}$$

so that the third standardized multivariate cumulant is

$$\kappa_3(\boldsymbol{Z}) \approx \begin{pmatrix} 0.0062 & -0.0026 & 0.0155 \\ -0.0026 & 0.0011 & 0.0287 \\ 0.0155 & 0.0287 & 0.0543 \\ -0.0026 & 0.0011 & 0.0287 \\ 0.0011 & -0.0004 & 0.2299 \\ 0.0287 & 0.2299 & 0.2359 \\ 0.0155 & 0.0287 & 0.0543 \\ 0.0287 & 0.2299 & 0.2359 \\ 0.0543 & 0.2359 & 0.4882 \end{pmatrix}$$

and the skewness matrix is

$$M_{\boldsymbol{Y}} = \kappa_3(\boldsymbol{Z})^T \kappa_3(\boldsymbol{Z}) \approx \begin{pmatrix} 0.00514 & 0.02689 & 0.04195 \\ 0.02689 & 0.16302 & 0.2267 \\ 0.04195 & 0.2267 & 0.41035 \end{pmatrix}.$$

The eigenvalues of M are approximately 0.54948, 0.0285442 and 0.000476673. The eigenvector corresponding to the dominant eigenvalue $\psi \approx 0.54948$ is

$$\boldsymbol{v} \approx \pm(0.0911, 0.5086, 0.8562)^T.$$

Using eq. (6.85), the euclidean norm of $\hat{L}^{-1}\hat{\boldsymbol{\lambda}}$ is computed from ψ as $||\hat{L}^{-1}\hat{\boldsymbol{\lambda}}|| \approx 1.19979$. The lower Cholesky factor $\hat{L}$ of $\hat{\Sigma}$ is

$$\hat{L} \approx \begin{pmatrix} 0.6921 & 0 & 0 \\ 0.7629 & 0.7113 & 0 \\ -0.6505 & -0.4883 & 0.5574 \end{pmatrix},$$

so that the estimate of the shape vector of the skew-normal approximation of the random vector $\boldsymbol{Y}$ is given by eq. (6.84) as

$$\hat{\boldsymbol{\lambda}} \approx \pm(0.0757, 0.5174, 0.2035)^T.$$

The skewness of $\max(X_{n-1}, X_n)$ is positive according to the element in the bottom right corner of $\kappa_3(\boldsymbol{Y})$. Therefore, the final estimate of the shape vector is

$$\hat{\boldsymbol{\lambda}} \approx (0.0757, 0.5174, 0.2035)^T.$$

The approach described in this section has $O(n^4)$ worst-case runtime and space complexity due to the computation of the Kronecker product in eq. (6.79), which produces a $(n-1)^2 \times (n-1)^2$ matrix. The runtime complexity can be reduced to $O(n^2)$ without sacrificing the accuracy, as will be shown in section 6.5. An efficient method to compute and update the inverse of the lower Cholesky factor $\hat{L}^{-1}$ of $\hat{\Sigma}$ is presented in section 6.4.

6.4 Incremental Update of Inverse Cholesky Factor

The inverse Cholesky factor $\hat{L}^{-1}$ can be computed in an efficient and numerically stable manner by first computing the Cholesky factorization of $\hat{\Sigma}$ and then inverting the resulting Cholesky factor $\hat{L}$. However, if the Cholesky factorization of the covariance matrix Σ of the random vector $\boldsymbol{X}$ is available, it is more efficient to compute the inverse Cholesky factor of $\hat{\Sigma}$ by *updating* the inverse Cholesky factor of Σ.

Given the Cholesky factorization

$$\Sigma = LL^T, \tag{6.91}$$

then by left and right multiplication with L^{-1} and L^{-T}, respectively, this becomes

$$L^{-1}\Sigma L^{-T} = I_n, \tag{6.92}$$

where $I_n \in \mathbb{R}^{n\times n}$ denotes the identity matrix. Next, the matrices Σ and L^{-1} are partitioned as

$$\Sigma = \begin{pmatrix} A & \boldsymbol{a} \\ \boldsymbol{a}^T & e \end{pmatrix} \tag{6.93}$$

$$L^{-1} = \begin{pmatrix} B & \boldsymbol{0} \\ \boldsymbol{c}^T & d \end{pmatrix}, \tag{6.94}$$

where $A, B \in \mathbb{R}^{(n-1)\times(n-1)}$, $\boldsymbol{a}, \boldsymbol{c} \in \mathbb{R}^{n-1}$ and $e, d \in \mathbb{R}$. Then eq. (6.92) can be written as

$$\begin{pmatrix} B & \boldsymbol{0} \\ \boldsymbol{c}^T & d \end{pmatrix} \begin{pmatrix} A & \boldsymbol{a} \\ \boldsymbol{a}^T & e \end{pmatrix} \begin{pmatrix} B^T & \boldsymbol{c} \\ \boldsymbol{0}^T & d \end{pmatrix} = \begin{pmatrix} BA & B\boldsymbol{a} \\ \boldsymbol{c}^T A + d\boldsymbol{a}^T & \boldsymbol{c}^T\boldsymbol{a} + de \end{pmatrix} \begin{pmatrix} B^T & \boldsymbol{c} \\ \boldsymbol{0}^T & d \end{pmatrix} = I_n, \tag{6.95}$$

so that

$$\begin{pmatrix} BAB^T & BA\boldsymbol{c} + dB\boldsymbol{a} \\ \boldsymbol{c}^T AB^T + d\boldsymbol{a}^T B^T & \boldsymbol{c}^T A\boldsymbol{c} + 2d\boldsymbol{a}^T\boldsymbol{c} + d^2 e \end{pmatrix} = I_n, \tag{6.96}$$

which implies

$$BAB^T = I_{n-1}. \tag{6.97}$$

The matrix A is a principal submatrix of the covariance matrix Σ, which is symmetric positive definite by definition. Clearly, A is symmetric and corollary 2.5 implies that

A is also positive definite. Furthermore, B and therefore also B^{-1} is lower triangular. Then from eq. (6.92) and theorem 2.6 it follows that

$$A = B^{-1}B^{-T} \tag{6.98}$$

is the Cholesky factorization of A with lower triangular inverse Cholesky factor B. Since by definition, Σ without the last row and column is equal to A, it follows that the corresponding inverse Cholesky factor B can be efficiently computed by removing the last row and column from L^{-1}.

From section 6.3.1 it is known that $\hat{\Sigma}$ is obtained from Σ by removing the last and replacing the second last row and column Σ. To efficiently compute $\hat{L}^{-1}$ by exploiting the knowledge of L^{-1}, the idea is to remove the last two rows and columns of Σ and then add the last row and column of $\hat{\Sigma}$ to the resulting matrix to obtain $\hat{\Sigma}$. According to the above explanations, the removal of the last two rows and columns of Σ implies the removal of the last two rows and columns of L^{-1}. Afterwards, a new row and column must be added to the resulting lower triangular matrix to finally obtain $\hat{L}^{-1}$, which will be explained in the following.

The incremental update of the inverse Cholesky factor after adding a new row and column to the covariance matrix, such that the resulting matrix is symmetric positive definite, can again be derived from eq. (6.96), which implies that

$$BA\boldsymbol{c} + dB\boldsymbol{a} = \boldsymbol{0} \tag{6.99}$$

$$\boldsymbol{c}^T A\boldsymbol{c} + 2d\boldsymbol{a}^T\boldsymbol{c} + d^2 e = 1. \tag{6.100}$$

By replacing A using eq. (6.98), eq. (6.99) becomes

$$\boldsymbol{c} = -dB^T\boldsymbol{u}, \tag{6.101}$$

where

$$\boldsymbol{u} = B\boldsymbol{a}. \tag{6.102}$$

The scalar d can be computed from eq. (6.100), which can be simplified by replacing A and c using eqs. (6.98) and (6.101) as

$$1 = \boldsymbol{c}^T A\boldsymbol{c} + 2d\boldsymbol{a}^T\boldsymbol{c} + d^2 e \tag{6.103}$$

$$= d^2\boldsymbol{u}^T\boldsymbol{u} - 2d^2\boldsymbol{a}^T B^T\boldsymbol{u} + d^2 e \tag{6.104}$$

$$= d^2(e - \boldsymbol{u}^T\boldsymbol{u}). \tag{6.105}$$

From theorem 2.6 it follows that $d > 0$, so that the last equation implies $e - \boldsymbol{u}^T\boldsymbol{u} > 0$ and

$$d = \frac{1}{\sqrt{e - \boldsymbol{u}^T\boldsymbol{u}}}. \tag{6.106}$$

Therefore, the last row of $\hat{L}^{-1}$ can be efficiently computed with eqs. (6.101), (6.102) and (6.106) in $O(n^2)$.

6.5 Quadratic Time Algorithm for MAX-operation

This section shows how the skew-normal distribution based MAX-operation can be computed in $O(n^2)$ time without sacrificing the accuracy of the results. The main idea is explained for a special case in section 6.5.1. The following section 6.5.2 introduces a transformation, which makes this idea applicable in the general case. The optimized algorithm is finally presented in section 6.5.3. All proofs are given in section A.2.3.

6.5.1 Fast Algorithm based on Restricted Skew-Normal Distribution

This subsection shows that the shape vector $\hat{\boldsymbol{\lambda}}$ for the proposed MAX-operation can be efficiently computed if $\boldsymbol{X}$ has a 'restricted' variant of the skew-normal distribution.

Definition 6.6. Let $\boldsymbol{X}$ denote a n-dimensional skew-normal random vector and let $\mathcal{SN}_n(\boldsymbol{\mu}, \Sigma, \boldsymbol{\lambda})$ denote the distribution of $\boldsymbol{X}$. If the conditions

$$\Sigma_{n-1,i} = \Sigma_{n,i} \tag{6.107}$$

$$\lambda_i = 0 \tag{6.108}$$

are satisfied for all $1 \leq i \leq n-4$, then $\boldsymbol{X}$ is said to have a *restricted skew-normal distribution*.

In the following, eq. (6.107) is called *covariance condition* and eq. (6.108) is called *skewness condition*. If the random vector $\boldsymbol{X}$ has a restricted skew-normal distribution then the computation of the statistical MAX-operation simplifies because many of the joint central moments $\mathbb{E}(\bar{Y}_i\bar{Y}_j\bar{Y}_k)$ are zero, as shown by the following lemma.

Lemma 6.7. *Let $\boldsymbol{X} = (X_1, \ldots, X_n)^T$ denote a n-dimensional random vector and let the random vector $\boldsymbol{Y} = (Y_1, \ldots, Y_{n-1})^T$ be defined as*

$$\boldsymbol{Y} = (X_1, \ldots, X_{n-2}, \max(X_{n-1}, X_n))^T. \tag{6.109}$$

If $\boldsymbol{X}$ has a restricted skew-normal distribution, then

$$\mathbb{E}(\bar{Y}_i\bar{Y}_j\bar{Y}_k) = 0 \tag{6.110}$$

for all $i, j, k \in \mathbb{N}$ with $1 \leq i \leq n-4$ and $1 \leq j, k \leq n-1$, where $\bar{\boldsymbol{Y}} = (\bar{Y}_1, \ldots, \bar{Y}_{n-1})^T$ denotes the centred random vector corresponding to $\boldsymbol{Y}$.

In other words, if the conditions of the previous lemma are satisfied, then all elements in the third multivariate cumulant $\kappa_3(\boldsymbol{Y})$ of the random vector $\boldsymbol{Y}$, except for those corresponding to the $3^3 = 27$ joint central moments

$$\mathbb{E}(\bar{Y}_i\bar{Y}_j\bar{Y}_k) \text{ with } n-3 \leq i, j, k \leq n-1, \tag{6.111}$$

are zero. This leads to the main theorem of this chapter.

Theorem 6.8. *Let* $\boldsymbol{X} = (X_1, \ldots, X_n)^T$ *denote a n-dimensional random vector and let the random vector* $\boldsymbol{Y}$ *be defined as* $\boldsymbol{Y} = (X_1, \ldots, X_{n-2}, \max(X_{n-1}, X_n))^T$. *Furthermore, let* $\hat{\Sigma}$ *denote the covariance matrix of* $\boldsymbol{Y}$ *with Cholesky factorization* $\hat{\Sigma} = \hat{L}\hat{L}^T$ *and let* $\kappa_3(\boldsymbol{V})$ *denote the third multivariate cumulant of the random vector*

$$\boldsymbol{V} = (X_{n-3}, X_{n-2}, \max(X_{n-1}, X_n))^T. \tag{6.112}$$

If $\boldsymbol{X}$ *has a restricted skew-normal distribution, then the skewness matrix of* $\boldsymbol{Y}$ *is of the form*

$$M_{\boldsymbol{Y}} = \begin{pmatrix} \boldsymbol{0}_{n-4,n-4} & \boldsymbol{0}_{n-4,3} \\ \boldsymbol{0}_{3,n-4} & G \end{pmatrix}, \tag{6.113}$$

where $\boldsymbol{0}_{k,l}$ *is the* $(k \times l)$ *zero matrix,*

$$G = \hat{L}_{22}^{-1}\kappa_3(\boldsymbol{V})^T(\hat{L}_{22}^{-T} \otimes \hat{L}_{22}^{-T})(\hat{L}_{22}^{-1} \otimes \hat{L}_{22}^{-1})\kappa_3(\boldsymbol{V})\hat{L}_{22}^{-T} \tag{6.114}$$

and $\hat{L}_{22}^{-T}$ *is the bottom-right corner* 3×3 *submatrix of* $\hat{L}^{-1}$.

Clearly, $G \in \mathbb{R}^{3\times 3}$ is symmetric because it can be written as

$$G = \left((\hat{L}_{22}^{-1} \otimes \hat{L}_{22}^{-1})\kappa_3(\boldsymbol{V})\hat{L}_{22}^{-T}\right)^T (\hat{L}_{22}^{-1} \otimes \hat{L}_{22}^{-1})\kappa_3(\boldsymbol{V})\hat{L}_{22}^{-T}. \tag{6.115}$$

Let $\boldsymbol{v} \in \mathbb{R}^{n-1}$ denote the dominant eigenvector of the matrix $M_{\boldsymbol{Y}}$, such that

$$M_{\boldsymbol{Y}}\boldsymbol{v} = \psi\boldsymbol{v} \tag{6.116}$$

for some non-zero eigenvalue ψ. Then eq. (6.113) implies that $\boldsymbol{v}$ is of the form $\boldsymbol{v} = (\boldsymbol{0}, \bar{\boldsymbol{v}})^T$ and the subvector $\bar{\boldsymbol{v}} = (v_1, v_2, v_3)^T$ is the dominant eigenvector of G corresponding to the same eigenvalue ψ, so that

$$G\bar{\boldsymbol{v}} = \psi\bar{\boldsymbol{v}}. \tag{6.117}$$

Because $\hat{L}^{-1}$ is lower triangular, eq. (6.81) implies that the shape vector $\hat{\boldsymbol{\lambda}}$ is also of the form $\hat{\boldsymbol{\lambda}} = (\boldsymbol{0}, \bar{\boldsymbol{\lambda}})^T$, with subvector $\bar{\boldsymbol{\lambda}} \in \mathbb{R}^3$. Indeed, theorem 6.5 can be applied to the eigendecomposition of G to approximate the subvector $\bar{\boldsymbol{\lambda}}$ by replacing $\hat{L}$ and $\hat{L}^{-1}$ with $\hat{L}_{22}$ and $\hat{L}_{22}^{-1}$, respectively, where $\hat{L}_{22}$ is the inverse of $\hat{L}_{22}^{-1}$. This approach will be further optimized as follows. By replacing $\boldsymbol{v}$ using eq. (6.81) and multiplication with $\hat{L}_{22}$, the last equation becomes

$$\hat{L}_{22}G\frac{\hat{L}_{22}^{-1}\bar{\boldsymbol{\lambda}}}{||\hat{L}_{22}^{-1}\bar{\boldsymbol{\lambda}}||} = \psi\hat{L}_{22}\frac{\hat{L}_{22}^{-1}\bar{\boldsymbol{\lambda}}}{||\hat{L}_{22}^{-1}\bar{\boldsymbol{\lambda}}||} = \psi\frac{\bar{\boldsymbol{\lambda}}}{||\hat{L}_{22}^{-1}\bar{\boldsymbol{\lambda}}||}. \tag{6.118}$$

Let $\hat{\Sigma}_{22}^{-1}$ denote the bottom-right corner (3×3) submatrix of the inverse covariance matrix $\hat{\Sigma}^{-1}$. Then because $\hat{L}^{-T}$ is upper triangular,

$$\hat{L}_{22}^{-T}\hat{L}_{22}^{-1} = \hat{\Sigma}_{22}^{-1} \tag{6.119}$$

must hold. Using eqs. (2.10) and (6.114), eq. (6.118) becomes

$$\left(\kappa_3(\boldsymbol{V})^T(\hat{\Sigma}_{22}^{-1}\otimes\hat{\Sigma}_{22}^{-T})\kappa_3(\boldsymbol{V})\hat{\Sigma}_{22}^{-1}\right)\bar{\boldsymbol{\lambda}}=\psi\bar{\boldsymbol{\lambda}}, \tag{6.120}$$

which implies that the subvector $\bar{\boldsymbol{\lambda}}$ is a scalar multiple of an eigenvector of the matrix

$$H:=\hat{L}_{22}G\hat{L}_{22}^{-1}=\kappa_3(\boldsymbol{V})^T(\hat{\Sigma}_{22}^{-1}\otimes\hat{\Sigma}_{22}^{-T})\kappa_3(\boldsymbol{V})\hat{\Sigma}_{22}^{-1} \tag{6.121}$$

corresponding to the eigenvalue ψ. Because H is a similarity transformation of G, G and H have the same eigenvalues and ψ is the dominant eigenvalue of G and H that must also satisfy eq. (6.87).

If $\bar{\boldsymbol{\lambda}}/||\bar{\boldsymbol{\lambda}}||$ is the normalized eigenvector of H associated with the dominant eigenvalue ψ, then by defining

$$x=\frac{\bar{\boldsymbol{\lambda}}^T}{||\bar{\boldsymbol{\lambda}}||}\hat{\Sigma}_{22}^{-1}\frac{\bar{\boldsymbol{\lambda}}}{||\bar{\boldsymbol{\lambda}}||} \tag{6.122}$$

it follows from eq. (6.82) that

$$\psi=\left(2-\frac{\pi}{2}\right)^2||\bar{\boldsymbol{\lambda}}||^6x^3 \tag{6.123}$$

and finally

$$||\bar{\boldsymbol{\lambda}}||=\sqrt[6]{\frac{4\psi}{x^3(\pi-4)^2}}. \tag{6.124}$$

The correct sign of $\bar{\boldsymbol{\lambda}}$ can again be determined from the sign of the skewness of $\max(X_{n-1},X_n)$, as described in section 6.3.3.

From the above it is apparent that the runtime of the algorithm in section 6.3 can be reduced to $O(n^2)$ if $\boldsymbol{X}$ has a restricted skew-normal distribution, because not only the matrix H but also its eigendecomposition can be computed very efficiently. In particular, a very fast non-iterative algorithm can be used to efficiently compute the eigendecomposition of the matrix $H\in\mathbb{R}^{3\times3}$.

6.5.2 Transformation to Restricted Skew-Normal Distribution

This subsection shows, that the runtime of the algorithm for the proposed skew-normal distribution based MAX-operation can be reduced to $O(n^2)$ in the general case, where the random vector $\boldsymbol{X}$ has an arbitrary skew-normal distribution. The main idea is to transform the random vector $\boldsymbol{X}$ to a random vector $\boldsymbol{W}$ with restricted skew-normal distribution using an invertible linear transformation that can be computed very efficiently. Afterwards, the fast algorithm for the restricted skew-normal distribution is applied to the random vector $\boldsymbol{W}$. The shape vector $\hat{\boldsymbol{\lambda}}$ is finally obtained by applying the inverse linear transformation to the shape vector of the skew-normal distribution approximation of $\boldsymbol{W}$. In the following, $\boldsymbol{X}$ is assumed to be an n-dimensional skew-normal random vector with $n>4$.

To satisfy the covariance condition eq. (6.107), the idea is to define a linear transformation which maps the n-dimensional random vector $\boldsymbol{X}$ to a n-dimensional random vector $\boldsymbol{W}$ such that $W_i = X_i + q_i X_{n-2}$ for all $1 \leq i \leq n-4$ and $W_i = X_i$ otherwise. Then it follows from eq. (6.63) that the covariances between W_i and the last two random variables W_{n-1} and W_n are

$$\begin{aligned}\mathrm{Cov}(W_{n-1}, W_i) &= \mathrm{Cov}(X_{n-1}, X_i + q_i X_{n-2}) = \mathrm{Cov}(X_{n-1}, X_i) + q_i \mathrm{Cov}(X_{n-1}, X_{n-2}) \\ \mathrm{Cov}(W_{n-0}, W_i) &= \mathrm{Cov}(X_{n-0}, X_i + q_i X_{n-2}) = \mathrm{Cov}(X_{n-0}, X_i) + q_i \mathrm{Cov}(X_{n-0}, X_{n-2}).\end{aligned}$$

Let $c_j = \mathrm{Cov}(X_{n-1}, X_j) - \mathrm{Cov}(X_n, X_j)$ for all $1 \leq j \leq n$. It follows that if $c_{n-2} \neq 0$ and $q_i = -c_i / c_{n-2}$ then $\mathrm{Cov}(W_{n-1}, W_i) = \mathrm{Cov}(W_n, W_i)$ for all $1 \leq i \leq n-4$.

A similar transformation can be applied to satisfy the skewness condition eq. (6.108). Suppose a linear transformation maps the n-dimensional random vector $\boldsymbol{X}$ to an n-dimensional random vector $\boldsymbol{W}$ such that $W_i = X_i + r_i X_{n-3}$ for all $1 \leq i \leq n-4$ and $W_i = X_i$ otherwise. Let $\boldsymbol{\lambda_X}$ and $\boldsymbol{\lambda_W}$ denote the shape vector of $\boldsymbol{X}$ and $\boldsymbol{W}$, respectively. Then eq. (6.64) implies that the shape parameter $\lambda_{\boldsymbol{W},i}$ with $1 \leq i \leq n-4$ of the random variable W_i is

$$\lambda_{\boldsymbol{W},i} = \lambda_{\boldsymbol{X},i} + r_i \lambda_{\boldsymbol{X},n-3}. \tag{6.125}$$

It follows that if $\lambda_{\boldsymbol{X},n-3} \neq 0$ and $r_i = -\lambda_{\boldsymbol{X},i} / \lambda_{\boldsymbol{X},n-3}$, then $\lambda_{\boldsymbol{W},i} = 0$ for all $1 \leq i \leq n-4$.

Both transformations can be combined into a single linear transformation, which leads to the following lemma.

Lemma 6.9. *Let $\boldsymbol{X} \sim \mathcal{SN}_n(\boldsymbol{\mu}, \Sigma, \boldsymbol{\lambda})$ be a n-dimensional skew-normal random vector and let $c_j := Cov(X_{n-1}, X_j) - Cov(X_n, X_j)$ for all $1 \leq j \leq n$. If $c_{n-3}\lambda_{n-2} - c_{n-2}\lambda_{n-3} \neq 0$, then there exists an invertible linear transformation, which maps $\boldsymbol{X} = (X_1, \ldots, X_n)^T$ to a n-dimensional random vector $\boldsymbol{W} = (W_1, \ldots, W_n)^T$ with*

$$W_i = \begin{cases} X_i + q_i X_{n-2} + r_i X_{n-3} & \text{for } 1 \leq i \leq n-4 \\ X_i & \text{for } n-3 \leq i \leq n, \end{cases} \tag{6.126}$$

such that $\boldsymbol{W}$ has a restricted skew-normal distribution.

In the special case $c_{n-3}\lambda_{n-2} - c_{n-2}\lambda_{n-3} = 0$, the transformation in lemma 6.9 can be modified e.g. by using a suitable permutation of variables, as described by the following remark.

Remark 6.10. Let $\boldsymbol{X}$ denote a n-dimensional skew-normal random vector, which doesn't satisfy the covariance condition eq. (6.107) and the skewness condition eq. (6.108) of a restricted skew-normal distribution and $c_{n-3}\lambda_{n-2} - c_{n-2}\lambda_{n-3} = 0$, where

$$c_i := \mathrm{Cov}(X_{n-1}, X_i) - \mathrm{Cov}(X_n, X_i) \tag{6.127}$$

for $1 \leq i \leq n$. If there exists an $1 \leq i, j \leq n-2$ such that $c_j\lambda_i - c_i\lambda_j \neq 0$, then X_{n-2} and X_{n-3} only need to be exchanged by the variables X_i and X_j. If there is no such pair, then $c_j\lambda_i - c_i\lambda_j = 0$ must hold for all $1 \leq i, j \leq n-2$ and one of the following three cases must be satisfied.

(i) If $\exists i \in \{1,\dots,n-2\}$ such that $c_i \neq 0$ and $\lambda_i = 0$, then $\lambda_j = 0\ \forall j \in \{1,\dots,n-2\}$

(ii) If $\exists i \in \{1,\dots,n-2\}$ such that $c_i = 0$ and $\lambda_i \neq 0$, then $c_j = 0\ \forall j \in \{1,\dots,n-2\}$.

(iii) If $c_i \neq 0$ and $\lambda_i \neq 0\ \forall i \in \{1,\dots,n-2\}$, then $\frac{c_i}{\lambda_i} = \frac{c_j}{\lambda_j}$ for all $i,j \in \{1,\dots,n-2\}$.

In each case, a single variable X_i with $c_i \neq 0$ or $\lambda_i \neq 0$ can be exchanged with X_{n-2} or X_{n-3} and the transformations detailed at the beginning of this subsection can be used.

6.5.3 Description of Quadratic Time Algorithm

This subsection combines the results of the previous subsections to create a quadratic time algorithm for the computation of the shape vector $\hat{\boldsymbol{\lambda}}$. The mean vector $\hat{\boldsymbol{\mu}}$ and covariance matrix estimate $\hat{\Sigma}$ are computed as described in section 6.3.1 in linear time.

It is assumed that $\boldsymbol{X}$ doesn't satisfy the conditions of the restricted skew-normal distribution. Therefore, the linear transformation described in section 6.5.2 must be applied to obtain the random vector $\boldsymbol{W}$ of restricted skew-normal distribution. Let $A \in \mathbb{R}^{n\times n}$ denote a full-rank matrix such that

$$\boldsymbol{W} = A\boldsymbol{X} \tag{6.128}$$

realizes the linear transformation defined by eq. (6.126) in lemma 6.9. Furthermore, let $A_{-n,-n}$ denote the matrix obtained by removing the last row and column of matrix A. By replacing $\boldsymbol{X}$ with $\boldsymbol{W}$ in theorem 6.8, this theorem can be used to estimate the shape vector of the random vector

$$A_{-n,-n}\boldsymbol{Y} = (W_1,\dots,W_{n-2},\max(W_{n-1},W_n))^T. \tag{6.129}$$

By definition, the transformation described by lemma 6.9 does not change the last four variables of $\boldsymbol{X}$, so that the random vector $\boldsymbol{V}$ is not affected by the transformation. Then theorem 6.8 can be used to estimate the shape vector of the random vector $A_{-n,-n}\boldsymbol{Y}$ by computing the eigendecomposition of a matrix $\tilde{H} \in \mathbb{R}^{3\times 3}$. The matrix $\tilde{H}$ is obtained from eq. (6.121) by replacing the submatrix $\hat{\Sigma}_{22}^{-1}$ with the bottom-right corner (3×3) submatrix of the inverse covariance matrix of $A_{-n,-n}\boldsymbol{Y}$. This submatrix, which will be denoted by $\hat{S}_{22}^{-1}$, can be computed in $O(n^2)$ time as follows.

If $\hat{\Sigma}$ is the covariance matrix of the random vector

$$\boldsymbol{Y} = (X_1,\dots,X_{n-2},\max(X_{n-1},X_n))^T, \tag{6.130}$$

and $\hat{S}$ denotes the covariance matrix of the random vector $A_{-n,-n}\boldsymbol{Y}$ then theorem 2.37 implies that

$$\hat{S} = A_{-n,-n}\hat{\Sigma}(A_{-n,-n})^T \tag{6.131}$$

so that its inverse is

$$\hat{S}^{-1} = (A_{-n,-n})^{-T}\hat{\Sigma}^{-1}(A_{-n,-n})^{-1} = (A_{-n,-n})^{-T}\hat{L}^{-T}\hat{L}^{-1}(A_{-n,-n})^{-1}. \tag{6.132}$$

Equation (6.126) states that $X_{n-2} = W_{n-2}$ and $X_{n-3} = W_{n-3}$ so that the inverse linear transformation $\boldsymbol{X} = A^{-1}\boldsymbol{X}$ is

$$X_i = \begin{cases} W_i - q_i W_{n-2} - r_i W_{n-3} & \text{for } 1 \leq i \leq n-4 \\ W_i & \text{for } n-3 \leq i \leq n. \end{cases} \tag{6.133}$$

Hence, A^{-1} can be obtained from A in $O(n)$ time by changing the signs of the coefficients $q_1, \ldots, q_{n-4}$ and $r_1, \ldots, r_{n-4}$. Another implication is that

$$(A_{-n,-n})^{-1} = (A^{-1})_{-n,-n}. \tag{6.134}$$

Let $B^{-1} \in \mathbb{R}^{(n-1)\times 3}$ denote the matrix obtained by removing all but the last three columns of $(A^{-1})_{-n,-n}$. It follows that

$$\hat{S}_{22}^{-1} = B^{-T}\hat{L}^{-T}\hat{L}^{-1}B^{-1} = (\hat{L}^{-1}B^{-1})^T(\hat{L}^{-1}B^{-1}) \tag{6.135}$$

and hence, the matrix $\hat{S}_{22}^{-1} \in \mathbb{R}^{3\times 3}$ can be computed in $O(n^2)$ time because the computation of the matrix $\hat{L}^{-1}B^{-1} \in \mathbb{R}^{(n-1)\times 3}$ can be done in $O(n^2)$ time.

As explained above, the matrix H, defined in eq. (6.121), is modified accordingly by replacing $\hat{\Sigma}_{22}^{-1}$ with $\hat{S}_{22}^{-1}$, which gives

$$\tilde{H} = \kappa_3(\boldsymbol{V})^T(\hat{S}_{22}^{-1} \otimes \hat{S}_{22}^{-T})\kappa_3(\boldsymbol{V})\hat{S}_{22}^{-1}. \tag{6.136}$$

Let $\tilde{\psi}$ denote the dominant eigenvalue of the matrix $\tilde{H}$ corresponding to the eigenvector $\tilde{\boldsymbol{\lambda}} = (\tilde{\lambda}_1, \tilde{\lambda}_2, \tilde{\lambda}_3)^T$. The norm of $\tilde{\boldsymbol{\lambda}}$ is computed using eqs. (6.122) and (6.124) by replacing $\hat{\Sigma}_{22}^{-1}$, $\bar{\lambda}$ and ψ with $\hat{S}_{22}^{-1}$, $\tilde{\lambda}$ and $\tilde{\psi}$, respectively.

Then the $(n-1)$-dimensional shape vector estimate for the random vector $A_{-n,-n}\boldsymbol{Y}$ is $(0, \ldots, 0, \tilde{\lambda}_1, \tilde{\lambda}_2, \tilde{\lambda}_3)^T$. According to eqs. (6.64) and (6.133), the estimate of the $(n-1)$-dimensional shape vector for the distribution of the original random vector $\boldsymbol{Y}$ is obtained by the inverse linear transformation

$$\hat{\boldsymbol{\lambda}} = (A_{-n,-n})^{-1}(0, \ldots, 0, \tilde{\lambda}_1, \tilde{\lambda}_2, \tilde{\lambda}_3)^T, \tag{6.137}$$

which can be written as

$$\hat{\lambda}_i = \begin{cases} \tilde{\lambda}_i - q_i\tilde{\lambda}_2 - r_i\tilde{\lambda}_1 & \text{for } 1 \leq i \leq n-4 \\ \tilde{\lambda}_{i-(n-4)} & \text{for } n-3 \leq i \leq n-1 \end{cases} \tag{6.138}$$

for all $1 \leq i \leq n-1$, where q_i and r_i are computed as shown in the proof of lemma 6.9 in section A.2.3.

6.6 Application to the Computation of $\max(X_1, \ldots, X_n)$

As with the normal distribution based MAX-operation, the skew-normal distribution based MAX-operation can be applied multiple times. During each application, two

random variables are replaced by a third random variable so that each application reduces the number of random variables by one.

Let $\boldsymbol{X} = (X_1^{(0)}, \ldots, X_n^{(0)})^T$ be a n-dimensional skew-normal random vector. The first application of the skew-normal distribution based MAX-operation approximates the random vector

$$(X_1^{(0)}, \ldots, X_{n-2}^{(0)}, \max(X_{n-1}^{(0)}, X_n^{(0)}))^T$$

with a $(n-1)$-dimensional skew-normal random vector $(X_1^{(1)}, \ldots, X_{n-1}^{(1)})^T$. The second application of the skew-normal distribution based MAX-operation approximates the random vector

$$(X_1^{(1)}, \ldots, X_{n-3}^{(1)}, \max(X_{n-2}^{(1)}, X_{n-1}^{(1)}))^T$$

with a $(n-2)$-dimensional skew-normal random vector $(X_1^{(2)}, \ldots, X_{n-2}^{(2)})^T$ and so on.

In the second last application, the random vector

$$(X_1^{(n-3)}, \max(X_2^{(n-3)}, X_3^{(n-3)}))^T$$

is approximated by the random vector $(X_1^{(n-2)}, X_2^{(n-2)})^T$ with a bivariate skew-normal distribution using a simplified version of the presented algorithm.

Finally, the last application uses the method of moments (moment matching) to approximate the distribution of the random variable $\max(X_1^{(n-2)}, X_2^{(n-2)})$ with a new random variable $X_1^{(n-1)}$ of univariate skew-normal distribution using eqs. (6.24) to (6.26), which gives the final result. Therefore, the distribution of the maximum $\max(X_1, \ldots, X_n)^T$ is obtained by $n-1$ applications of the skew-normal distribution based MAX-operation.

6.7 Conclusion

The statistical SUM and MAX-operation are the fundamental operations of block-based statistical timing analysis. While the SUM-operation can usually be computed efficiently, the efficient computation of the MAX-operation is, on the other hand, one of the most challenging problems of block-based statistical timing analysis. For a given random vector

$$\boldsymbol{X} = (X_1, \ldots, X_n)^T, \tag{6.139}$$

the MAX-operation must approximate the distribution of the random vector

$$\boldsymbol{Y} = (X_1, \ldots, X_{n-2}, \max(X_{n-1}, X_n))^T \tag{6.140}$$

with a member of the same family of probability distributions as the random vector $\boldsymbol{X}$. So far, block-based statistical timing analysis has relied on the normal distribution based MAX-operation [Clark61], which can cause large approximation errors because it approximates the distribution of $\boldsymbol{Y}$ with a normal distribution.

To minimize the approximation error, this chapter introduces the skew-normal distribution based MAX-operation. Compared to the normal distribution based MAX-operation,

the proposed MAX-operation is defined on the far more flexible skew-normal distribution, which allows the accurate approximation of the random vector $\boldsymbol{Y}$ with another skew-normal distribution. The results in section 7.5 consistently show a significant reduction of the approximation error by up to 90%. However, the proposed algorithm has a slightly larger runtime complexity which might limit its application to $n < 1000$.

CHAPTER 7

Experimental Evaluation

7.1 Implementation

All algorithms were implemented in C/C++ and executed on Intel Core i7-2600K processor workstations with 32GiB RAM. For the generation of the circuit instances with random delay values, the high-performance implementations of the Box-Muller transform and the Mersenne Twister pseudo-random number generator from the Intel Math Kernel Library is used [Intel15].

To accurately measure the runtime of the presented algorithms and individual steps, the hardware performance counters of the processor are accessed using the perfmon interface. These hardware counters are configured to count the number of elapsed CPU clock cycles for selected regions of code. All results are stored in a MySQL database. The runtime is obtained by dividing the total number of clock cycles by the CPU clock frequency.

7.2 Benchmark Circuits

The proposed algorithms were evaluated on several large benchmark circuits, kindly provided by NXP. The circuits were synthesized and optimized for speed using a commercial synthesis tool. The NanGate 45 nm Open Cell Library [nan11] was chosen as a target library. The benchmark circuit characteristics are shown in table 7.1, where #PPI and #PPO denote the number of pseudo-primary inputs and pseudo-primary outputs, respectively. The last three columns will be explained at the end of this section.

It was assumed, that every delay value X_i of a gate has a normal distribution with mean μ_i and standard deviation σ_i. The mean μ_i is set to the nominal delay value that

▼ **Table 7.1 —** Benchmark circuit characteristics

circuit	#PPI	#PPO	#gates	$Q_{0.6}$ [ps]	$Q_{0.8}$ [ps]	$Q_{0.95}$ [ps]
p35k	2912	2229	28115	1163.6	1266.8	1412.7
p45k	3739	2550	26954	889.3	968.4	1084.8
p77k	3487	3400	41797	5805.2	6377.2	7137.5
p78k	3148	3484	57535	1213.3	1325.4	1485.2
p81k	4029	3952	91756	1018.5	1109.7	1238.6
p100k	5902	5829	61749	1400.1	1533.9	1710.6
p267k	17332	16621	138912	787.9	856.2	951.6
p330k	18010	17468	184425	1023.5	1116.9	1246.4

was extracted from the Standard Delay Format (SDF) description of the synthesized netlists. The standard deviation of X_i was defined as

$$\sigma_i := c_v |\mu_i|, \tag{7.1}$$

where $|\mu_i|$ denotes the absolute value of the nominal delay and c_v is a variation coefficient. To study large process variations in innovative technology nodes and other sources of delay variations like model inadequacy and environmental variations, a variation coefficient of $c_v = 0.25$ was chosen [Ye10].

Manufacturing data from a Intel microprocessor in 65nm technology [Kuhn08] has shown that inter-die variations accounts for roughly 3% and intra-die variations causes approximately 2.5% variability of the clock frequency of on-chip ring oscillators. Furthermore, intra-die variations are expected to dominate in future technology nodes, due to a shift to purely random and independent physical variations like random dopant fluctuation and line edge roughness [Li10, Agarw07]. To account for inter-die and intra-die variations, every delay value X_i of a gate was defined as

$$X_i = \frac{1}{\sqrt{2}} \left(Z_{\text{inter}} + Z_{\text{intra},i} \right) \sigma_i + \mu_i, \tag{7.2}$$

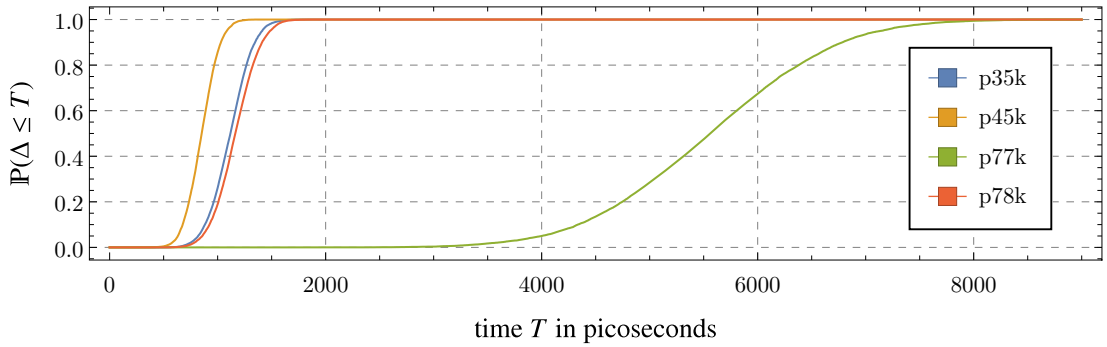
where Z_{inter} and $Z_{\text{intra},i}$ are independent random variables of standard normal distribution. While a separate random variable $Z_{\text{intra},i}$ exists for each X_i, the random variable Z_{inter} is shared by all delay values of all gates to account for inter-die variations, which affects all delays of the gates on the same die in a similar way. For example, to consider the components of process variation which occur predominantly from wafer-to-wafer and lot-to-lot. From eq. (2.28) and theorem 2.24 it follows that

$$\begin{aligned} \operatorname{Var}(X_i) &= \operatorname{Var}\left(\frac{1}{\sqrt{2}} \left(Z_{\text{inter}} + Z_{\text{intra},i} \right) \sigma_i \right) && (7.3) \\ &= \frac{1}{2} (\operatorname{Var}(Z_{\text{inter}}) + \operatorname{Var}(Z_{\text{intra},i})) \sigma_i^2 && (7.4) \\ &= \sigma_i^2. && (7.5) \end{aligned}$$

As can be seen from eq. (7.4), half of the variance of X_i is due to inter-die and the other half is due to intra-die variations.

The creation of circuit layouts was omitted to avoid an unnecessary complex experimental setup. Consequently, spacial correlations have not been considered. The interconnect delays have been estimated during the synthesis and are considered to be part of the gate delay.

The clock cycle time T_{clk} is determined from the distribution of the *circuit delay* Δ, which is defined as follows. In this experimental setup, the *circuit delay* is defined by the maximum delay of any path that is sensitized by a given large set of test vector-pairs under the impact of delay variations. To compute the circuit delay distribution, all available test vector-pairs were simulated with the nominal circuit instance. The test vector-pairs were then sorted according to the arrival time of the last transition at the circuit outputs. Afterwards, the 250 test vector-pairs with the latest transition arrival time at the circuit outputs were simulated with a Monte Carlo simulation of 10^4 iterations. In each iteration, the arrival time of the last transition at the circuit outputs across all test vector-pairs was stored in a database. The hereby obtained approximation of the circuit delay distribution is shown in fig. 7.1 for the first four benchmark circuits.



▲ **Figure 7.1 —** CDF of the circuit delay Δ for several NXP benchmark circuits

The 0.6, 0.8 and 0.95-quantile of the circuit delay distribution is presented in the last three columns of table 7.1. For a benchmark circuit, the p-quantile of the circuit delay distribution is the time $T \in \mathbb{R}$, such that the probability that the circuit delay of a randomly chosen circuit instance is less or equal T is equal to p, that is

$$\mathbb{P}(\Delta \leq T) = p. \tag{7.6}$$

For example, the 0.95-quantile $Q_{0.95}$ denotes the time where 5% of the defect-free manufactured chips would fail the timing requirements due to delay variations.

A path with delay Y is a critical path according to definition 2.17 if

$$\mathbb{P}(Y > T_{clk}) > p_{cri}, \tag{7.7}$$

where $p_{cri} = 1 - \Phi(3) \approx 0.0013499$ was chosen in this experimental setup to achieve high accuracy. If μ denotes the mean and σ denotes the standard deviation of Y, then

eq. (7.7) is equivalent to

$$\mu + 3\sigma > T_{clk} \tag{7.8}$$

because the path delays are normally distributed in this experimental setup.

7.3 Probabilistic Sensitization Analysis

This section presents the experimental results for the probabilistic sensitization analysis. At first, the details of the experimental setup are explained. Afterwards, the experimental results are discussed.

For each benchmark circuit, the 10000 longest paths were found using a commercial static timing analysis tool. During that process, the number of paths which were allowed to terminate at the same primary output was limited to 100. The resulting set of paths was subsequently sensitized using a commercial ATPG tool. On average, 2536 test vector-pairs were generated for each benchmark circuit. The only notable exception is p77k, where only 114 test vector-pairs had been generated.

Section 7.3.1 evaluates the accuracy and speedup of the Monte Carlo simulation of the subcircuit $\mathcal{S}$, compared to the Monte Carlo simulation of the complete circuit. The experimental results for the simplified probabilistic sensitization analysis are presented in section 7.3.2.

7.3.1 Evaluation of Representative Subcircuit

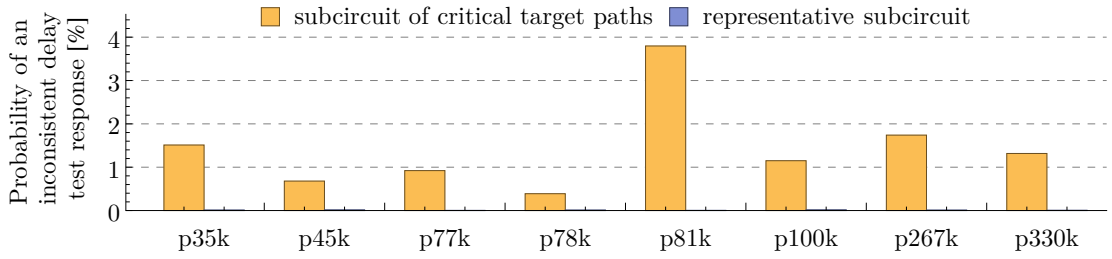
The proposed representative subcircuit $\mathcal{S}$ was constructed as described in section 4.4 using 100 circuit instances $\theta_0, \ldots, \theta_{99}$, including the nominal circuit instance θ_0. Storing all circuit instances in memory did not require more than 8 GiB of RAM per process.

To show the importance of the controlling paths for the delay test, another subcircuit $\bar{\mathcal{S}}$, consisting only of the critical target paths, was constructed as follows. At first, the given test vector-pair was simulated with the nominal circuit instance θ_0 and all critical paths, which are sensitized in the nominal circuit instance, are identified. Afterwards, the subcircuit $\bar{\mathcal{S}}$ is constructed from all gates and interconnects which lie on at least one of these paths. All floating gate input nodes are set to their respective non-controlling values. For the floating off-path input node of a XOR/XNOR gate, the logic value, which the off-path input had at the arrival time of the on-path transition, is used. Finally, all floating circuit outputs are set to the value observed at the clock cycle time.

To evaluate the suitability of the subcircuits for the simulation of path delay fault tests with the given test vector-pair, a Monte Carlo simulation with 10^4 iterations (circuit instances) was performed. Following the simulation of a circuit instance θ, both subcircuits $\mathcal{S}$ and $\bar{\mathcal{S}}$ were simulated using the same random delay values as in θ. Afterwards, the logic values at the outputs of the circuit instance and the outputs of the subcircuits were observed at the clock cycle time T_{clk} and compared to the expected fault free stable output values. An *inconsistent delay test result* occurs if a delay fault is detected in only either the simulation of the circuit instance or in the simulation

of the subcircuit. The following figs. 7.2 to 7.4 present average results for path delay fault tests with otherwise defect free circuits, where the clock cycle time was chosen corresponding to the 0.95-quantile of the circuit delay distribution.

The average probability of an inconsistent delay test result is shown in fig. 7.2. It can be seen that the probability of an inconsistent delay test result is quite high for the subcircuit containing only the critical target paths. After extending the subcircuit to a representative subcircuit, the probability of an inconsistent delay test result drops to almost zero. This shows that the delays of the controlling paths play a key role during delay testing by determining which critical paths are sensitized by the test vector-pair in any circuit instance.

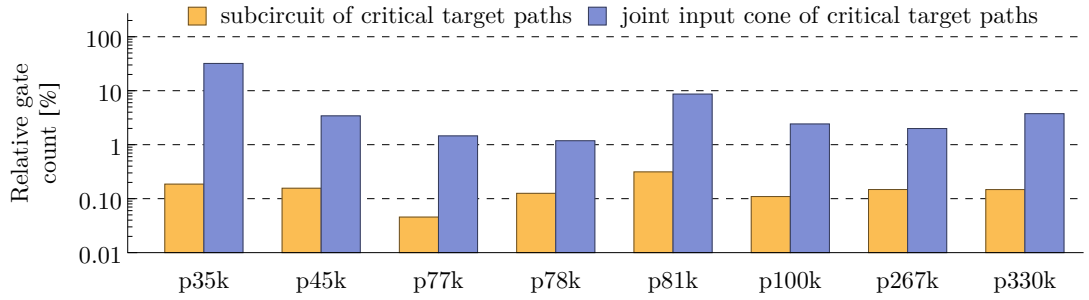


▲ **Figure 7.2 —** Average probability of observing an inconsistent delay test result with the subcircuit

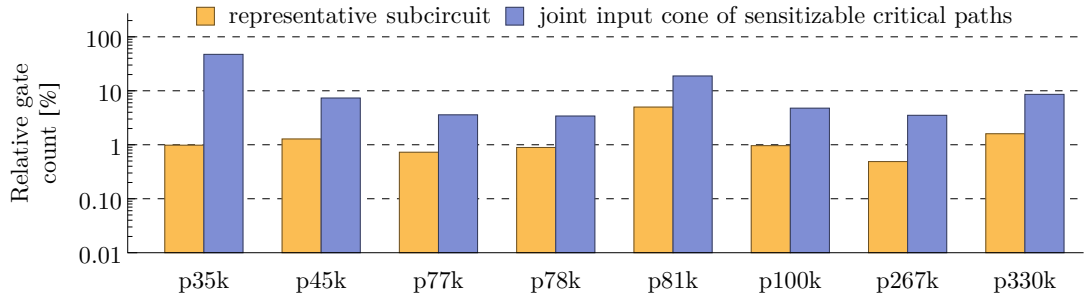
The relative size of a subcircuit is defined as the number of gates in the subcircuit, divided by the number of gates in the respective (complete) benchmark circuit. Figure 7.3 shows the average relative size of the subcircuits and the average relative size of the joint input cone of all critical paths that exist in the respective subcircuit.

As shown by fig. 7.3a, the relative size of the subcircuit $\bar{\mathcal{S}}$ that consists only of the critical target paths is on average more than 10× smaller than the joint input cone of all critical paths that exist in $\bar{\mathcal{S}}$. Extending the subcircuit to the representative subcircuit $\mathcal{S}$ results in only a small increase in the relative subcircuit size. As shown in fig. 7.3b, the average relative size of the representative subcircuit is about 1%. The only exception is p81k, which is not surprising because fig. 7.2 has already shown that the controlling paths have a strong impact on the delay test result for this benchmark circuit. Therefore, the construction of a representative subcircuit requires a greater number of critical and controlling paths, which increases the average relative size of the representative subcircuit to 5%. In general, many other critical paths may be sensitized by the test vector-pair in randomly chosen circuit instances. Therefore, the joint input cone of all critical paths in the representative subcircuit may on average contain up to 47% of the gates of the complete circuit.

The small size of the representative subcircuit $\mathcal{S}$ and the low probability of an inconsistent delay test result motivates the use of the representative subcircuit for more complex statistical timing analysis problems, such as for the computation of the probability that a particular target path is sensitized by a given test vector-pair, as explained in



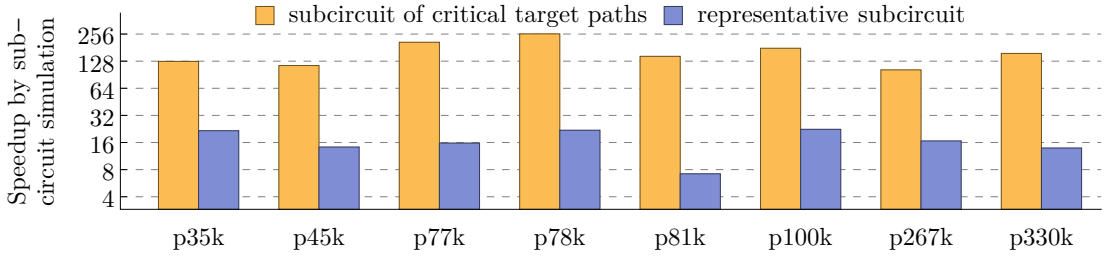
(a) Subcircuit that consists only of critical target paths



(b) Representative subcircuit

▲ **Figure 7.3 —** Relative size of subcircuit and relative size of joint input cone of all critical paths that exist in the subcircuit

section 4.4. The *average speedup* is defined as the ratio of the average runtime of the Monte Carlo simulation of the complete circuit, divided by the average runtime for the construction and Monte Carlo simulation of the subcircuit. The average speedup gained by this approach is presented in fig. 7.4 for different benchmark circuits. The results show that it is between 8 and 256 times faster to first generate the subcircuit and then perform a Monte Carlo simulation of the subcircuit instead of directly performing a Monte Carlo simulation of the complete circuit.



▲ **Figure 7.4 —** Speedup of Monte Carlo simulation by constructing and simulating only the subcircuit, compared to Monte Carlo simulation of complete circuit

The detailed numerical results are presented in table B.1a for path delay fault tests with

otherwise defect free circuits. Additional results for small delay fault tests in circuits with a marginally detectable small delay fault are shown in table B.1b. Both tables present average results over all test vector-pairs of a benchmark circuit.

The first column shows the name of the NXP benchmark circuit. The second column "T_{clk}" shows the clock cycle time, corresponding to the 0.6, 0.8 and 0.95-quantile of the circuit delay distribution. For small delay fault tests, only the clock cycle time corresponding to the 0.95-quantile is considered.

The following five columns ("subcircuit $\bar{\mathcal{S}}$") show the experimental results for the subcircuit $\bar{\mathcal{S}}$, which was constructed from θ_0 without using probabilistic sensitization analysis. The experimental results for the representative subcircuit S, which was constructed using the proposed probabilistic sensitization analysis, are shown in the last six columns ("representative subcircuit $\mathcal{S}$").

The average number of critical paths, which are sensitized by the test vector-pair in at least one of the considered circuit instances, is shown in column "#sens.crit. paths". The number of critical paths is much smaller for $\bar{\mathcal{S}}$ because only the nominal circuit instance is considered and many other critical paths may be sensitized by the test vector-pair in randomly chosen circuit instances. The column "|cone|", presents the average relative number of gates in the joint input cone of the critical paths in the subcircuit, compared to number of gates in the benchmark circuit. As expected from the greater number of sensitized critical paths in the representative subcircuit $\mathcal{S}$, the number of gates in the joint input cone of these paths is also much larger for $\mathcal{S}$.

The average relative number of gates in each subcircuit, compared to the number of gates in the benchmark circuit, is shown in the columns "$|\bar{\mathcal{S}}|$" and "$|\mathcal{S}|$", respectively. The results show that the representative subcircuit $\mathcal{S}$ is usually only slightly larger than the subcircuit $\bar{\mathcal{S}}$. The column "#iter" gives the average number of iterations, during which the subcircuit was extended and simulated with the delay values of all considered circuit instances $\theta_0, \ldots, \theta_{99}$.

The probability of an inconsistent delay test result is given in the column "P_{err}" in percent [%]. The average speedup is shown in column "SU". The results show a large average speedup of up to 257$\times$ can be achieved by the Monte Carlo simulation of $\bar{\mathcal{S}}$. However, as expected, the probability of an inconsistent delay test result is quite high and increases rapidly with the clock frequency, which shows the large impact of the controlling paths on the delay test.

On the other hand, the simulation of the representative subcircuit $\mathcal{S}$ is very accurate and the probability of an inconsistent delay test result "P_{err}" is almost zero. Furthermore, the last column "SU" shows that it is still up to 32$\times$ faster to construct the representative subcircuit and then perform a Monte Carlo simulation of the representative subcircuit, instead of directly performing a Monte Carlo simulation of the complete circuit. The speedup is particularly large for the Monte Carlo simulation of circuits with a marginally detectable small delay fault for the evaluation of small delay fault tests.

7.3.2 Simplified Probabilistic Sensitization Analysis

This subsection demonstrates the efficiency and flexibility of the proposed simplified probabilistic sensitization analysis for guiding the delay test generation process. At first, important path sensitization conditions are reviewed in section 7.3.2.

Path Sensitization Conditions

The tolerance of a path delay fault test towards delay variability is influenced by the sensitization conditions that are satisfied by the off-path inputs of the path. These sensitization conditions are of limited use for delay testing under the impact of delay variations [Sauer12], because they consider only very few details of the actual waveforms, such as the initial and the final value. Nevertheless, the most common sensitization conditions are briefly reviewed here to describe important details of the experimental results.

Table 7.2 shows the sensitization condition satisfied by a single off-path input for different on- and off-path input transitions [Fuchs94, Krsti95]. The logic symbols are explained below.

- S0 (S1) represents a waveform with only one transition (v_0, t_0) and $v_0 = 0$ ($v_0 = 1$)
- X0 (X1) means that the final value of a waveform is 0 (1)
- 01 (10) means the initial value of the waveform is 0 (1) and the final value is 1 (0)

The most stringent sensitization condition considered in the experiments is the *robust* sensitization condition and the least stringent is the *functional* sensitization condition. The relaxations made by going from one sensitization condition to the next less stringent conditions are emphasized with bold letters.

The classification of a path delay fault test is determined by the weakest sensitization condition satisfied by any of its off-path inputs. For example, if some of the off-path inputs satisfy the robust sensitization condition while the remaining off-path inputs only satisfy the non-robust sensitization condition, then the path is tested by a *non-robust test*.

A *robust test* can detect a path delay fault, regardless of the delays of other paths in the circuit. However, a large number of paths have no robust test [Lin87, Fuchs91]. Although synthesis for 100% robust testability is generally possible, this approach has

▼ **Table 7.2 —** Sensitization condition satisfied by an off-path input

	off-path input					
sensitization condition	Robust		Non-robust		functional	
on-path input	rising	falling	rising	falling	rising	falling
AND/NAND gate	X1	S1	X1	**X1**	X1	**10**
OR/NOR gate	S0	X0	**X0**	X0	**01**	X0
XOR/XNOR gate	S0/S1	S0/S1	**X0/X1**	**X0/X1**	X0/X1	X0/X1

several major disadvantages such as a significant area overhead and possibly many additional primary inputs [Jha92, Chakr00].

The sensitization conditions of a *non-robust test* are less stringent than those of a robust test. However, the sensitization of a target path by a *non-robust test* depends on the delays of other paths in the circuit. It is therefore likely, that a non-robust test sensitizes a target path in only a subset of the manufactured circuits. For example, the orange path a-b-c-d in fig. 1.5 is tested by a non-robust test, which can fail to sensitize the target path even if all transitions at the off-path inputs arrive well before the respective transition at the on-path input.

A logical path which is neither robustly nor non-robustly testable can still cause a timing failure during normal operation of the circuit. For this reason, the category of *functionally sensitizable* paths was introduced in [Krsti95].

Experimental Results

Each of the n test vector-pairs of a benchmark circuit was simulated with the nominal circuit instance θ_0 and the sensitized critical paths were assumed to be the target paths of the test vector-pair. In the following, $p_1, \ldots, p_{m_i}$ will be used to denote the m_i target paths of the ith test vector-pair v_i.

To evaluate the impact of delay variations on the sensitization of the target paths, the test vector-pair was then simulated with 99 randomly chosen circuit instances $\theta_1, \ldots, \theta_{99}$ and it was checked which of those target paths was sensitized in $\theta_1, \ldots, \theta_{99}$. The simplified probabilistic sensitization analysis was then applied to all inconsistently controlled target paths.

The results are presented in table B.2a for path delay fault tests with otherwise defect free circuits. Table B.2b presents additional results for small delay fault tests of marginally detectable size, which will be used in section 7.4. Column "T_{clk}" shows the clock cycle time, corresponding to the 0.6, 0.8 and 0.95-quantile of the circuit delay distribution. The following three columns show the average number of robustly ("#rob"), non-robustly ("#nr") and functionally ("#fs") tested target paths per test vector-pair.

The following four columns present the experimental results for non-robust path delay faults tests, considering a randomly chosen test vector-pair. Column "P_{ci}" shows the probability that in a randomly chosen circuit instance at least one non-robustly tested target path is not sensitized. This probability can be formally described by introducing the function f, which is defined as follows. If during the application of test vector-pair v_i to the circuit instance θ_j, the non-robust test of a target path p_k fails to sensitize p_k then $f(v_i, \theta_j, p_k) = 1$, else $f(v_i, \theta_j, p_k) = 0$. Then the probability P_{ci} is formally defined as

$$P_{\text{ci}} = \frac{1}{100n} \sum_{i=1}^{n} \sum_{j=0}^{99} \max_{k=1,\ldots,m_i} (f(v_i, \theta_j, p_k)). \tag{7.9}$$

The probability that a randomly chosen non-robustly tested target path is not sensitized in at least one circuit instance is given in column "P_{id}". This probability is computed as

$$P_{\text{id}} = \frac{1}{m_1 + ... + m_n} \sum_{i=1}^{n} \sum_{k=1}^{m_i} \max_{j=0,...,99} (f(\nu_i, \theta_j, p_k)). \tag{7.10}$$

The results show that this probability is almost one, so that it is very likely that a non-robust test is invalidated in at least one circuit instance. It may be surprising to observe that for some benchmark circuits, P_{ci} is quite small but P_{id} is almost one. This can be understood by a quite common example, where many non-robustly tested target paths start with the same sequence of gates, called *path segment*. If this common path segment is not sensitized in one circuit instance, then the delay tests of all target paths that run through this path segment are invalidated.

The next column "P_{in}" presents the probability that a randomly chosen non-robustly tested target path is not sensitized in a randomly chosen circuit instance. This probability has been computed as

$$P_{\text{in}} = \frac{1}{100(m_1 + ... + m_n)} \sum_{i=1}^{n} \sum_{j=0}^{99} \sum_{k=1}^{m_i} f(\nu_i, \theta_j, p_k). \tag{7.11}$$

The results in column "P_{in}" once again show that non-robust path delay fault test invalidation is very likely under the impact of delay variations.

Each invalidated non-robust path delay fault test was further analysed to determine if the propagation of the transition along the path was blocked by a waveform inconsistency at one of the inputs of the gate. The probability that the invalidation of a randomly chosen non-robust test of a target path in a randomly chosen circuit instance is caused by a waveform inconsistency at one of the on-path/off-path inputs of the path is shown in column "$P_{\text{wi|in}}$". The results show that non-robust path delay fault test invalidation is most likely caused by variations in the arrival time of the on-path/off-path transitions. However, in up to 31% of all invalidated non-robust path delay fault tests, the invalidation is instead caused by a waveform inconsistency at one of the on-path/off-path inputs. In this case, the invalidation of the path delay fault test is prevented by correcting this waveform inconsistency.

Finally, the average runtime per test vector-pair is presented in the last two columns. The runtime of the nominal circuit instance simulation and the identification of the target paths is shown in column "T_{sim}". Column "T_{diag}" presents the time required by the proposed probabilistic sensitization analysis. The runtime for the fault simulation [Fuchs91] itself is not included, since the fault simulation is merely used to separate non-robust tests from other path delay fault tests for the purpose of this presentation. The runtime for the random number generation to create the 99 randomly chosen circuit instances is also excluded, since the generation is done only once and the same circuit instances are shared by all test vector-pairs.

7.4 Computation of Target Paths Delay Fault Probability

This section compares the accuracy and runtime of the proposed non-incremental and incremental algorithms with extensive Monte Carlo simulations for a large number of marginally detectable small delay faults. The proposed algorithms approximate the target paths delay fault probability while the Monte Carlo simulation computes the delay fault detection probability.

Definition 7.1 (delay fault detection probability)**.** The probability of detecting a delay fault during delay testing (see section 1.2.1) of a randomly chosen circuit instance with a given set of test vector-pairs $\mathcal{T}$ and delay test parameters is called *delay fault detection probability* and denoted by Ξ.

To evaluate the effectiveness of a given set of test vector-pairs for the detection of small delay faults, definition 7.1 is applied to a modified set of circuit instances, which is obtained by injecting a marginally detectable small delay fault of fixed size into all circuit instances in Θ.

The first section 7.4.1 presents the experimental setup. The results obtained using the non-incremental algorithm are shown in section 7.4.2. The following section 7.4.3 presents the experimental results obtained using the incremental algorithm. The last section 7.4.4 applies the obtained experimental results to find a suitable trade-off between the test set size and the delay test quality.

7.4.1 Experimental Setup

A benchmark of 20000 randomly chosen single small delay faults was created for each circuit. Afterwards, a set of test vector-pairs was generated for each small delay fault. For each small delay fault, the 1000 longest paths through the fault site were identified with a commercial static timing analysis tool, where at most 100 paths were allowed to end at the same circuit node. Afterwards, the selected paths were sensitized using a commercial ATPG tool. The following experiments were conducted only with those small delay faults, for which at least 20 test vector-pairs had been generated. To study marginally detectable small delay faults, the fault size was set to the slack of the longest sensitized path $\tilde{\pi}$ in the nominal circuit instance, which passes through the fault site.

From the set of test vector-pairs that was generated for a small delay fault, four different test subsets $\mathcal{T}_1, \ldots, \mathcal{T}_4$ are derived as follows. The first test subset $\mathcal{T}_1$ consists of only a single test vector-pair, which sensitizes $\tilde{\pi}$ in the nominal circuit instance. Afterwards, $\mathcal{T}_{i+1}$ is derived from $\mathcal{T}_i$ by adding test vector-pairs in decreasing order of the delay of the longest target path through the fault site. The number of test vector-pairs to be added is chosen such that $\mathcal{T}_2$ contains five, $\mathcal{T}_3$ contains ten and $\mathcal{T}_4$ consists of twenty test vector-pairs. The critical target path sensitization by the test vector-pair in $\mathcal{T}_1$ has been evaluated with the simplified probabilistic sensitization analysis in table B.2b, which shows the strong impact of delay variations on the mostly non-robustly and in some cases only functionally sensitized critical target paths.

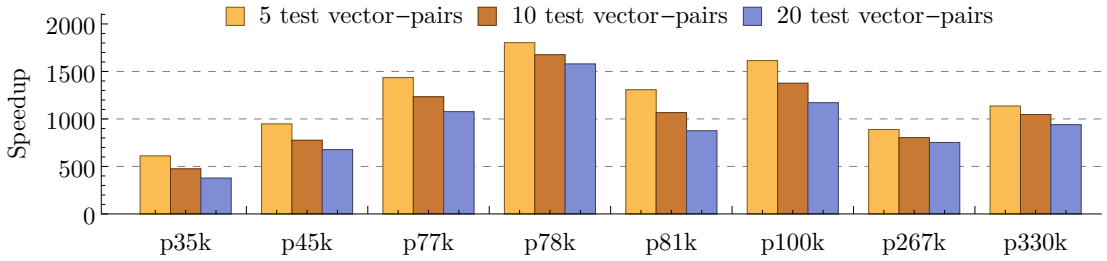
The clock cycle time T_{clk} was set to the 0.95-quantile of the circuit delay distribution. In other words, T_{clk} was chosen such that 5% of the defect-free manufactured chips are expected to fail the timing requirement due to process variations.

For each injected small delay fault, the delay fault detection probability Ξ (see definition 7.1) was computed for all test subsets $\mathcal{T}_1, \ldots, \mathcal{T}_4$ by a Monte Carlo simulation of 10^4 iterations. Assuming that the approximation error of the Monte Carlo simulation has a normal distribution, then the chosen number of iterations provides a 0.0098 half-length of the 95% confidence interval. In other words, the true delay fault detection probability lies within $\pm 0.98\%$ of the Monte Carlo simulation estimate with 95% confidence. The computational cost of the simulation is dominated by the large number of the random delay values required for each circuit instance (Monte Carlo iteration).

7.4.2 Non-Incremental Computation

In this section, the effectiveness of the non-incremental algorithm is demonstrated. The following figs. 7.5 and 7.6a presents average results over all small delay faults in a benchmark circuit.

The speedup achieved by the proposed non-incremental algorithm is shown in fig. 7.5 for different benchmark circuits. The *average speedup* is defined as the average runtime of the Monte Carlo simulation divided by the average runtime of the proposed algorithm. It can be seen that the average speedup of the proposed algorithm is about three orders of magnitude. While the runtime of the critical target path identification increases slowly with the test subset size $|\mathcal{T}|$, the relative contribution of the numerical integration to the overall runtime of the proposed algorithm increases significantly. Compared to this, the Monte Carlo simulation becomes more efficient, because the high computational cost for generating the large number of random delay values is shared among a greater number of test vector-pairs. Consequently, the average speedup decreases with the size of the test subset $|\mathcal{T}|$.



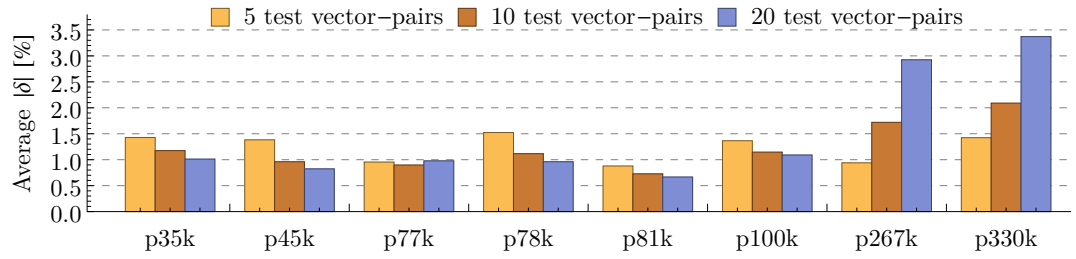
▲ **Figure 7.5 —** Average speedup of non-incremental algorithm compared to Monte Carlo simulation

The proposed algorithm is applied to approximate the delay fault detection probability. For a given test subset $\mathcal{T}$, the *error* δ of this approximation is defined as

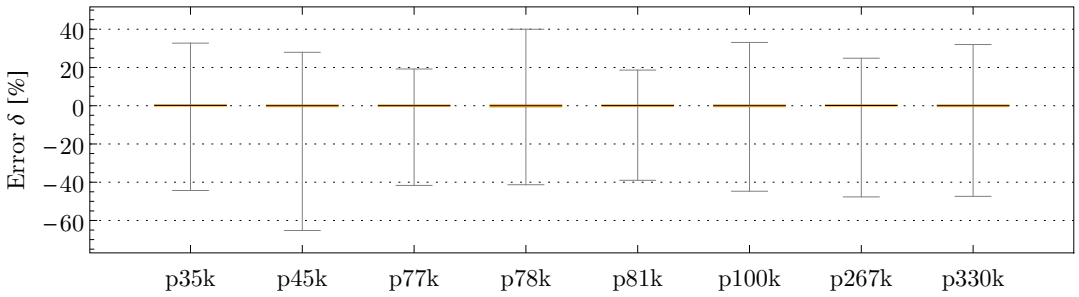
$$\delta = \Xi - \hat{\Psi}, \tag{7.12}$$

where the delay fault detection probability Ξ is obtained by the described Monte Carlo simulation and $\hat{\Psi}$ denotes the approximation of the target paths delay fault probability, obtained by the proposed non-incremental algorithm. This error is almost exclusively caused by the impact of delay variations on the sensitization of the critical paths. On the one hand, the delay fault detection probability may be overestimated if a critical target path is not sensitized in a randomly chosen circuit instance. On the other hand, a critical path which is not sensitized in the nominal circuit instance might be sensitized in a randomly chosen circuit instance, contributing to an underestimation of the delay fault detection probability.

The average absolute approximation error of the proposed non-incremental algorithm is presented in fig. 7.6a. The diagram shows that the error is quite small and tends to decrease with the test subset size. However, more than 1000 critical target paths were found for some delay faults in p267k and p330k, so that the normal distribution based MAX-operation was applied to reduce the number of random variables. The inherent inaccuracy of this operation causes the average absolute error to increase with the test subset size for p267k and p330k. A box plot (see section 2.4.2) of the error δ is shown in fig. 7.6b for the evaluation of the test subset $\mathcal{T}_2$ with five test vector-pairs. It is apparent that the error δ is almost zero for at least 50% of the experimental results and that in rare extreme cases (extreme outliers) the delay fault detection probability is underestimated by up to 40% or overestimated by up to 65%. These extreme outliers



(a) Average absolute error $|\delta|$ for test subsets $\mathcal{T}_2$, $\mathcal{T}_3$ and $\mathcal{T}_4$



(b) Box plot of the error δ for test subset $\mathcal{T}_2$ with 5 test vector-pairs

▲ **Figure 7.6 —** Error δ caused by approximating the delay fault detection probability with the target paths delay fault probability approximation of the non-incremental algorithm

are caused by the impact of delay variations on the sensitization of the critical paths and dominate the average absolute error in fig. 7.6a. Furthermore, the lower whiskers are slightly longer than the upper whiskers, which indicates a slight tendency towards overestimation of the delay fault detection probability.

Table B.3 presents the average results over all small delay faults in a benchmark circuit in detail. The name of the circuit is given in column "circuit" and column "$|\mathcal{T}|$" shows the number of test vector-pairs in the test subsets $\mathcal{T}_1, \ldots, \mathcal{T}_4$. Column "$|\Pi|$" presents the average number of critical target paths, sensitized by all test vector-pairs in the test subset. It can be observed that the number of critical target paths slowly saturates with the increasing number of test vector-pairs in the test subsets. This is because the additional test vector-pairs predominantly sensitize shorter paths and many critical paths may have already been sensitized by the smaller test subsets.

For low dimension of the multivariate normal integral (eq. (5.15)), very efficient specialized algorithms [Genz04] are used. For larger dimensions, a very efficient Monte Carlo numerical integration algorithm [Genz92] is used, which was published by its author in the Fortran routine MVNDST. The absolute error of the numerical integration was estimated to be below $5 \cdot 10^{-3}$ with 95% confidence with at most 10^5 Monte Carlo iterations.

Column "δ" shows the average and column "$|\delta|$" the average absolute value of the error δ (see eq. (7.12)) over all small delay faults in a benchmark circuit. The average error indicates a slight tendency towards overestimation for a single test vector-pair and a slight tendency towards underestimation of the delay fault detection probability for multiple test vector-pairs.

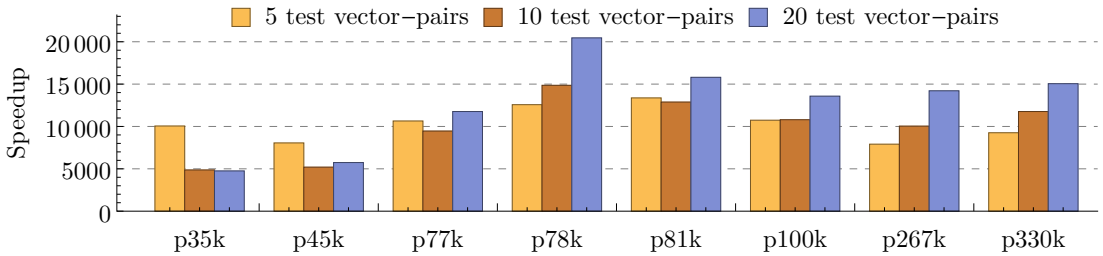
The average runtime of the proposed algorithm is shown in column "T_{PA}". This runtime is dominated by the numerical integration (41.8%), followed by the identification of the critical target paths (34.6%) and the simulation of the nominal circuit instance (23.6%). Column "T_{MC}" presents the average runtime of the Monte Carlo simulation. The average speedup of the proposed algorithm is given in column "SU".

7.4.3 Incremental Computation

In this section, the effectiveness of the incremental algorithm is demonstrated. An initially empty test subset is gradually extended by inserting single test vector-pairs in decreasing order of the longest target path delay in the nominal circuit instance. The following figs. 7.7 and 7.8a presents average results over all small delay faults in a benchmark circuit.

Suppose a test vector-pair has been added to the test subset, then the *average speedup* of the proposed incremental algorithm is defined as the average runtime of the Monte Carlo simulation of the extended test subset divided by the average runtime of the proposed algorithm to update the approximation of the target paths delay fault probability accordingly. The average speedup after the insertion of a test vector-pair is presented in fig. 7.7 for different benchmark circuits. The diagram shows a very large

average speedup of up to four orders of magnitude, which also tends to increase with the number of test vector-pairs under consideration.



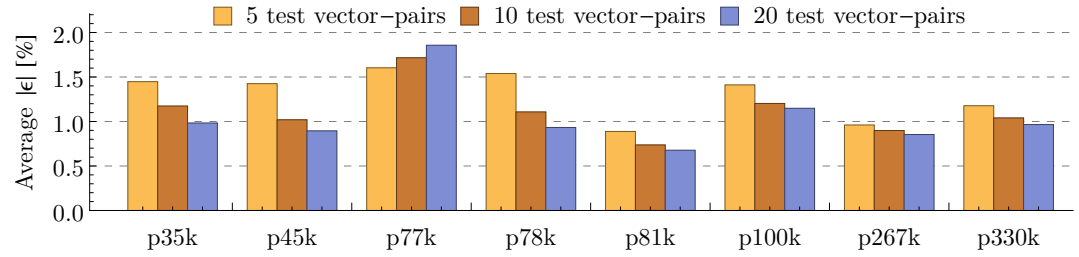
▲ **Figure 7.7 —** Average speedup of incremental algorithm after insertion of test vector-pair, compared to a Monte Carlo simulation of the extended test subset

The proposed incremental algorithm is applied to approximate the delay fault detection probability (see definition 7.1). The *total error* ϵ of this approximation is defined as

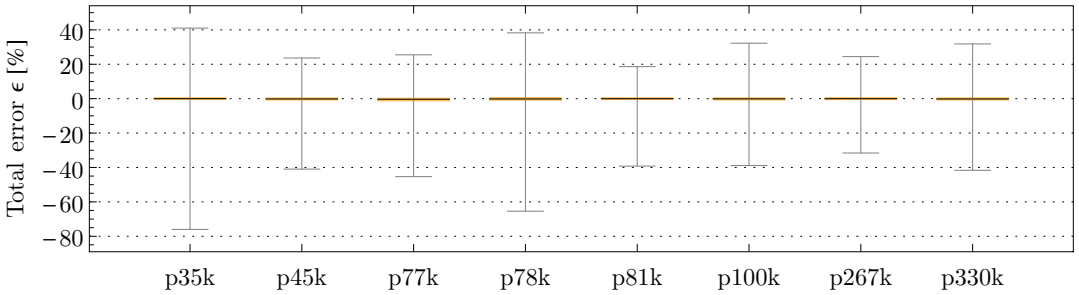
$$\epsilon = \Xi - \hat{\Psi}, \tag{7.13}$$

where the delay fault detection probability Ξ is obtained by the Monte Carlo simulation and $\hat{\Psi}$ denotes the approximation of the target paths delay fault probability, obtained by the proposed incremental algorithm. The average absolute total error is shown in fig. 7.8a. It can be seen that the absolute total error is comparable to the absolute error of the non-incremental algorithm. Although the absolute total error for p77k is slightly larger, the incremental algorithm doesn't require the normal distribution based MAX-operation for those small delay faults in p267k and p330k, where the number of critical target paths exceeds 1000. Instead, the proposed extension of the normal distribution based MAX-operation is used, which reduces the total error for p267k and p330k by up to 70%. A box plot (see section 2.4.2) of the total error ϵ is shown in fig. 7.8b for the evaluation of the test subset $\mathcal{T}_2$ with five test vector-pairs. It is apparent that the error ϵ is almost zero for at least 50% of the experimental results and that in rare extreme cases (extreme outliers) the delay fault detection probability is underestimated by up to 41% or overestimated by up to 76%. These extreme outliers are again caused by the impact of delay variations on the sensitization of the critical paths and dominate the average absolute total error $|\epsilon|$ in fig. 7.8a. Furthermore, a slight tendency towards overestimation of the delay fault detection probability is visible because the lower whiskers are slightly longer than the upper whiskers.

The detailed average results over all small delay faults in a benchmark circuit are presented in table B.4. The name of the benchmark circuit is shown in column "NXP circuit". Column "$|\mathcal{T}|$" presents the number of test vector-pairs in the test subset after an insertion or removal of a single test vector-pair. Only those updates, which resulted in a test subset of size 1, 5, 10 or 20 test vector-pairs, are presented here.



(a) Average absolute total error $|\epsilon|$ for test subsets $\mathcal{T}_2$, $\mathcal{T}_3$ and $\mathcal{T}_4$



(b) Box plot of the total error ϵ for test subset $\mathcal{T}_2$ with 5 test vector-pairs

▲ **Figure 7.8 —** Error ϵ caused by approximating the delay fault detection probability with the target paths delay fault probability approximation of the incremental algorithm

Column "T_{MC}" presents the average runtime of the Monte Carlo simulation. The results for the insertion of a single test vector-pair are shown in column "`INSERT`(test vector-pair)". The average number of critical target paths of all test vector-pairs in a test subset is given in column "Π". The part of the error caused by the impact of delay variations on the sensitization of the critical paths by the test subset is defined as

$$\delta = \Xi - \left(1 - \mathbb{P}\left(X_1 \leq T_{clk}, \ldots, X_{|\Pi|} \leq T_{clk}\right)\right) \tag{7.14}$$

and the average absolute value of δ (eq. (7.14)) is shown in column "$|\delta|$". The experimental results show that this error decreases rapidly as the number of test vector-pairs in the test subset increases. Column "T_{SA}" shows the average runtime of the sensitization analysis for a single test vector-pair.

If no critical target path was identified, the following steps of the algorithm are omitted. The update of the normal distribution approximation of the random vector $\boldsymbol{Y}$ (see eq. (5.17)) using the proposed extension of the normal distribution based MAX-operation is evaluated in the following two columns. The numerical error due to the (extended) normal distribution based MAX-operation is defined as

$$\begin{aligned} \varepsilon &= \left(1 - \mathbb{P}\left(X_1 \leq T_{clk}, \ldots, X_{|\Pi|} \leq T_{clk}\right)\right) - \left(1 - \mathbb{P}\left(Y_1 \leq T_{clk}, \ldots, Y_{|\mathcal{T}|} \leq T_{clk}\right)\right) \\ &= \mathbb{P}\left(Y_1 \leq T_{clk}, \ldots, Y_{|\mathcal{T}|} \leq T_{clk}\right) - \mathbb{P}\left(X_1 \leq T_{clk}, \ldots, X_{|\Pi|} \leq T_{clk}\right). \end{aligned} \tag{7.15}$$

Columns "$|\varepsilon|$" shows the average absolute value of ε, which is very small. This demonstrates the superior accuracy of the proposed covariance function for the normal

distribution based MAX-operation. However, additional experiments with other statistical timing analysis problems have shown that this error starts to increase rapidly as soon as the distributions of the maxima show a significant degree of skewness.

The average runtime for updating the normal distribution approximation of the random vector $\boldsymbol{Y}$ is shown in column "T_{DE}". The runtime is very small and it increases very slowly with the test subset size $|\mathcal{T}|$.

Some test vector-pairs in the test subsets for p330k sensitize thousands of critical target paths, with several critical paths being targeted by multiple test vector-pairs of the same test subset. As an optimization, it is possible to ignore all recurrences of a critical target path without affecting the accuracy of the target paths delay fault probability. However, care must be taken if such a critical target path is deleted by a subsequent removal of a test vector-pair or by the masking of an observable circuit output.

The target paths delay fault probability is computed using numerical integration with the FORTRAN routines `BVND`, `TVTL` and `MVNDST`, which were developed by Genz [Genz92, Genz04]. The average runtime of this computation is shown in column "T_{NI}". The runtime increases rapidly for small and slowly for bigger test subsets $|\mathcal{T}| \geq 10$. The parameters for the numerical integration were chosen, such that the estimated absolute error does not exceed $5 \cdot 10^{-3}$ with 95% confidence and no more than 10^5 Monte Carlo iterations. These routines were also used to calculate the errors δ and ε of the individual steps of the proposed algorithm.

The average absolute total error is shown in column "$|\epsilon|$" after the insertion or the removal of a test vector-pair. The error is small and mainly caused by the impact of delay variations on the sensitization of the critical paths.

The very large speedup of the proposed algorithm is shown in column "Speedup". The incremental algorithm is much faster than the non-incremental algorithm but may be slightly less accurate. The speedup for the removal of a single test vector-pair is even larger since only a rerun of the numerical integration is required. The update of other delay test parameters can be estimated from the presented results of the necessary individual steps of the algorithm. For example, the runtime of the algorithm after the reduction of the clock cycle time or the modification of the masking/unmasking of observable circuit outputs is roughly the sum of the runtimes given in columns "T_{DE}" and "T_{NI}".

7.4.4 Application to Variation-Aware Pattern Selection

In this subsection, the accuracy of the proposed approximation algorithm is demonstrated in the context of delay variation aware pattern selection for marginally detectable small delay faults. The delay fault detection probability (definition 7.1) is used to find a suitable compromise between the statistical delay test quality and the test subset size.

The results of the non-incremental algorithm in section 7.4.2 were stored in a database and are now applied to select a suitable test subset $\mathcal{T}_i \in \{\mathcal{T}_1, \ldots, \mathcal{T}_4\}$ for each small delay fault. For a given small delay fault and the test subset $\mathcal{T}_i$ with $1 \leq i \leq 4$, let Ξ_i

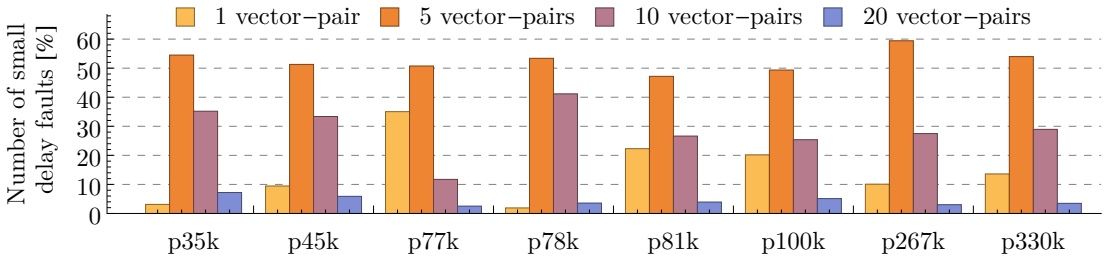
denote the delay fault detection probability and let $\hat{\Psi}_i$ denote the approximation of the target paths delay fault probability by the non-incremental algorithm.

Clearly, if the probability of detecting a particular small delay fault is not significantly increased by applying additional test vector-pairs, then these additional test vector-pairs are not required for the detection of this particular small delay fault. The distinction between a significant and a non-significant increase can be made based on a threshold b. In this application example, a threshold value of $b = 0.1$ is chosen so that this application never reduces the delay fault detection probability by more than 10%.

At first, a suitable test subset is selected for each small delay fault based on the delay fault detection probability. In this example, the delay fault detection probability Ξ_4 of the largest test set $\mathcal{T}_4$ with 20 test vector-pairs is compared to the delay fault detection probabilities Ξ_1, Ξ_2 and Ξ_3 obtained with the smaller test subsets $\mathcal{T}_1$, $\mathcal{T}_2$ and $\mathcal{T}_3$, respectively. In order to minimize the number of test vector-pairs while preserving a sufficiently high delay fault detection probability, the smallest test set $\mathcal{T}_i$ with a delay fault detection probability Ξ_i satisfying

$$\Xi_i \geq \Xi_4 - b \tag{7.16}$$

is selected. Based on this criterion, a suitable test subset is selected for each small delay fault in a benchmark circuit. The results are shown in fig. 7.9. It can be seen that for more 50% of all delay faults, the test subset $\mathcal{T}_2$ provides a sufficiently high delay fault detection probability with only 5 test vector-pairs.

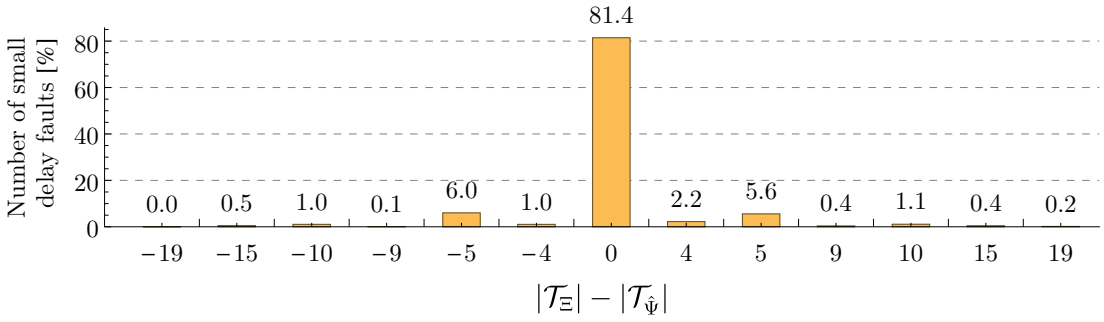


▲ **Figure 7.9 —** Average number of test vector-pairs required for the detection of a small delay fault of fixed size under the impact of delay variations

Afterwards, the whole process is repeated using the approximation $\hat{\Psi}_i$ of the target paths delay fault probability Ψ_i obtained by the non-incremental algorithm, instead of the delay fault detection probability Ξ_i with $1 \leq i \leq 4$. The results are compared in fig. 7.10. The difference in the test subset size $|\mathcal{T}_\Xi| - |\mathcal{T}_{\hat{\Psi}}|$ is shown on the abscissa, where $\mathcal{T}_\Xi$ and $\mathcal{T}_{\hat{\Psi}}$ denote the test subset chosen based on the delay fault detection probability (Ξ) and the approximation of the target paths delay fault probability ($\hat{\Psi}$), respectively. The ordinate shows the relative frequency of the test subset size difference over all small delay faults in all benchmark circuits.

For a small minority of small delay faults, this causes a slight underestimation or overestimation of the test subset size, as shown in fig. 7.10. It can be seen that the

test subset selection results match for more than 81% of the small delay faults. In only few cases, a slightly smaller or slightly larger test subset is selected, when the selection is based on the approximation of the target paths delay fault probability $\hat{\Psi}$. The distribution of this error is in line with expectations given by the box plot in fig. 7.6b and the explanation that this error is mainly caused by the impact of delay variations on the sensitization of the critical paths.



▲ **Figure 7.10 —** Error in test subset size when using target paths delay fault probability approximation instead of delay fault detection probability

7.5 SUM and MAX-Operations based on Skew-Normal Distribution

This section presents the experimental results for the proposed skew-normal distribution based MAX-operation. The skew-normal distribution based SUM-operation doesn't require experimental evaluation, as its high efficiency is apparent from theorem 6.3.

Let $\boldsymbol{X} = (X_1, \ldots, X_n)^T \sim \mathcal{N}_n(\boldsymbol{\mu}, \Sigma)$ denote a n-dimensional normal random vector. As explained in section 6.6, the skew-normal distribution based MAX-operation is repeatedly applied to approximate the distribution of $\max(X_1, \ldots, X_n)$ by a random variable Z with univariate skew-normal distribution. The approximation error ϵ is defined by the Kolmogorov-Smirnov statistic, which is the maximum absolute difference between the cumulative distribution function of Z and $\max(X_1, \ldots, X_n)$ at any point $t \in \mathbb{R}$, that is

$$\epsilon = \max_{t \in \mathbb{R}} \left| \mathbb{P}(\max(X_1, \ldots, X_n) \leq t) - \mathbb{P}(Z \leq t) \right|. \tag{7.17}$$

For every considered normal distribution of $(X_1, \ldots, X_n)^T$, the accurate distribution of $\max(X_1, \ldots, X_n)$ was obtained by a Monte Carlo simulation of 10^9 iterations. The large number of iterations is sufficient to compute the probability $\mathbb{P}(\max(X_1, \ldots, X_n) \leq t)$ for any $t \in \mathbb{R}$ with a maximum absolute error below 3.16228^{-5} with 95% confidence, which is typically much more accurate than computing the same probability with $\mathbb{P}(Z \leq t)$ when Z is obtained by the repeated application of the skew-normal distribution based MAX-operation. The results are compared to those obtained by the repeated application of the normal distribution based MAX-operation using Clark's

algorithm, which approximates the distribution of $\max(X_1, \ldots, X_n)$ by a univariate normal distribution. The *relative absolute error* of the skew-normal distribution based MAX-operation is defined as

$$\frac{|\epsilon_{sn}| - |\epsilon_n|}{|\epsilon_n|}, \tag{7.18}$$

where ϵ_n and ϵ_{sn} denote the error of the repeated application of the normal and skew-normal distribution based MAX-operation, as defined in eq. (7.17), respectively.

The details of the experimental results for different distributions of $(X_1, \ldots, X_n)^T$ are presented in the following subsections. The experimental results demonstrate the superior accuracy of the proposed skew-normal distribution based MAX-operation. Although the worst case runtime complexity is quadratic in the dimension of the random vector $\boldsymbol{X}$, the runtime of the proposed algorithm is very small even for several hundreds of random variables.

7.5.1 Results for $\max(X_1, \ldots, X_n)$ with Random μ and Σ

Let $\boldsymbol{X} = (X_1, \ldots, X_n)^T \sim \mathcal{N}_n(\boldsymbol{\mu}, \Sigma)$ be a random vector of multivariate normal distribution with mean vector

$$\boldsymbol{\mu} = (\mu_1, \ldots, \mu_n)^T \tag{7.19}$$

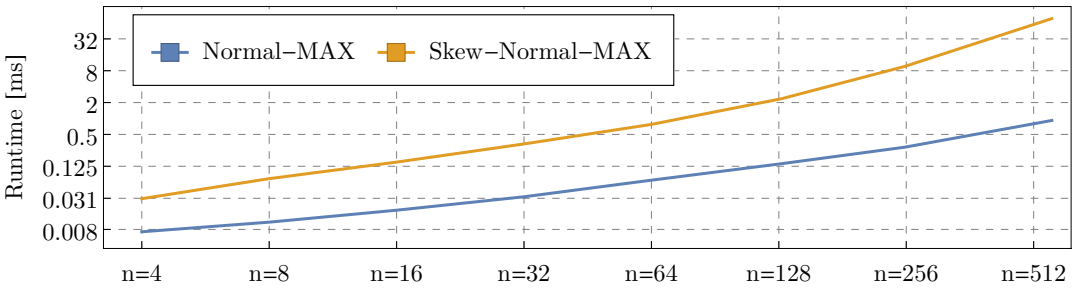
and covariance matrix

$$\Sigma = DRD, \tag{7.20}$$

where $R \in \mathbb{R}^{n \times n}$ is a random correlation matrix and $D \in \mathbb{R}^{n \times n}$ is a random diagonal matrix with the (positive) standard deviations $\sigma_1, \ldots, \sigma_n$ of $X_1, \ldots, X_n$ on the main diagonal. For all $i \in \mathbb{N}$ with $1 \leq i \leq n$, the mean μ_i and standard deviation σ_i of X_i were independently generated from a uniform distribution such that $\mu_i \in [0.9, 1.1]$ and $\sigma_i \in [0.8, 1.2]$. The proposed MAX-operation has been evaluated for 8 dimensions $n_1 = 4$, $n_2 = 8$, $n_3 = 16$, $n_4 = 32$, $n_5 = 64$, $n_6 = 128$, $n_7 = 256$ and $n_8 = 512$ of the random vector $\boldsymbol{X}$. For each dimension, 3000 random multivariate normal distributions with large variations of the correlation coefficients in R and 1000 random multivariate normal distributions with small variations of the correlation coefficients in R have been considered.

A log-log plot of the runtime of the normal distribution and skew-normal distribution based MAX-operations is presented in fig. 7.11. It can be seen that the runtime of the proposed algorithm for the skew-normal distribution based MAX-operation is comparable to Clark's algorithm for the normal distribution based-MAX operation when n is small. However, the runtime of the proposed algorithm increases faster as n becomes larger, but still remains very small even for several hundreds of random variables.

The average absolute error of the normal and skew-normal distribution based MAX-operation is presented in fig. 7.12 in percent [%] for different dimensions of $\boldsymbol{X}$ and large variations in the correlation coefficients in R. It can be seen that on average, the skew-normal distribution based MAX-operation reduces the approximation error by



▲ **Figure 7.11 —** Log-log plot of the runtime for the approximation of $\max(X_1, \ldots, X_n)$, where $(X_1, \ldots, X_n) \sim \mathcal{N}_n(\boldsymbol{\mu}, \Sigma)$ with random $\boldsymbol{\mu}$ and Σ and large correlation coefficient variations

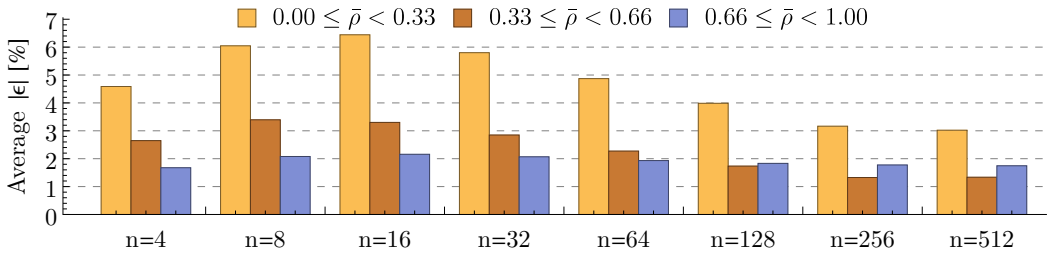
up to 60%. Furthermore, the box plot in fig. 7.12c shows a significant reduction of the maximum error when the average correlation between any two random variables is between 0.33 and 0.66. The variability of the correlation coefficients decreases with the dimension of $\boldsymbol{X}$, which causes the average absolute error to decrease with n.

The detailed numerical results are shown in table B.5a for small and in table B.5b for large variations of the correlation coefficients in R. The average correlation between any two random variables is given in column "$\bar{\rho}$". The columns "$|\epsilon_n|$" and "$|\epsilon_{sn}|$" show the average absolute error of the normal distribution based MAX-operation and the proposed skew-normal distribution based MAX-operation, respectively. The next column "$\frac{|\epsilon_{sn}|-|\epsilon_n|}{|\epsilon_n|}$" presents the relative absolute error of the skew-normal distribution based MAX-operation, compared to the normal distribution based MAX-operation. On average, the skew-normal distribution based MAX-operation reduces the approximation error by up to 90% for small and by up to 61% for large variations in the correlation coefficients in R. This demonstrates the superior accuracy of the proposed skew-normal distribution based MAX-operation. The last two columns "T_n" and "T_{sn}" show the runtime for the repeated application of the normal distribution based MAX-operation and the skew-normal distribution based MAX-operation, respectively.

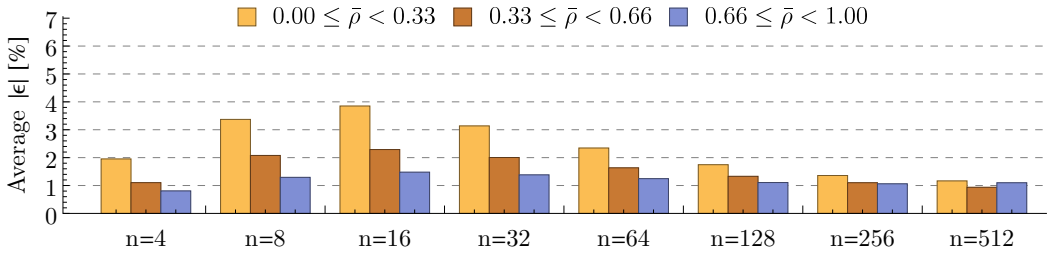
7.5.2 Results for $\max(X_1, \ldots, X_n)$ for Critical Target Path Delays $X_1, \ldots, X_n$

The proposed skew-normal distribution based MAX-operation was applied to compute the distribution of the maximum delay of the target paths, as defined in definition 1.4. Similar to the experimental setup for the evaluation of the target paths delay fault probability computation in section 7.4.1, different test subsets with 1, 5, 10 and 20 test vector-pairs were considered for each small delay fault in a benchmark circuit. For each test subset, the set of critical target paths was determined using the sensitization analysis in section 5.1. The delays of n critical target paths $X_1, \ldots, X_n$ were grouped into a random vector

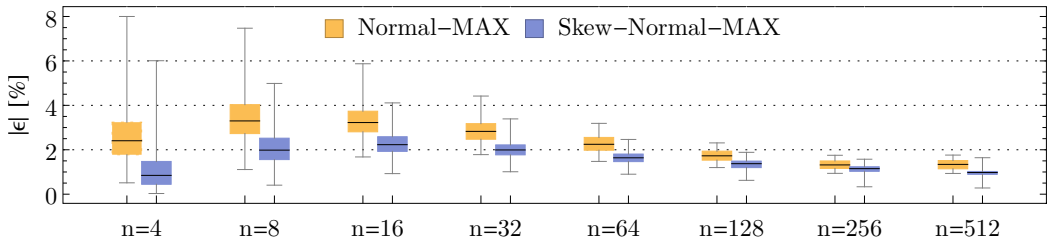
$$(X_1, \ldots, X_n)^T \sim \mathcal{N}_n(\boldsymbol{\mu}, \Sigma) \tag{7.21}$$



(a) Using normal distribution based MAX-operation



(b) Using skew-normal distribution based MAX-operation



(c) Box plot of absolute error for $0.33 \leq \bar{\rho} < 0.66$

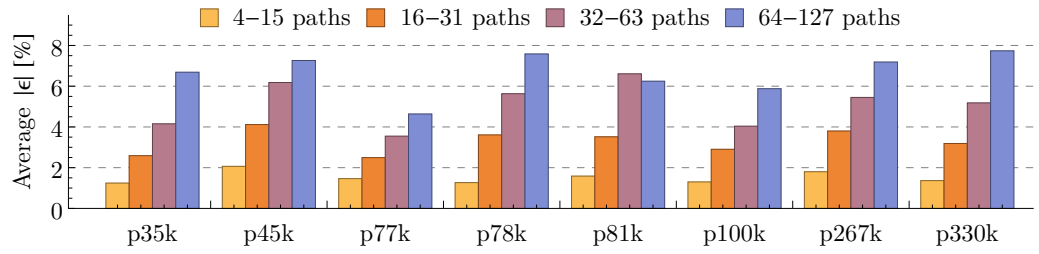
▲ **Figure 7.12 —** Error ϵ of $\max(X_1, \ldots, X_n)$ approximation, where $(X_1, \ldots, X_n) \sim \mathcal{N}_n(\boldsymbol{\mu}, \Sigma)$ with random $\boldsymbol{\mu}$ and Σ and large correlation coefficient variations

of n-dimensional normal distribution. Afterwards, the distribution of the maximum delay of the target paths $\max(X_1, \ldots, X_n)$ was computed by the repeated application of the normal and skew-normal distribution based MAX-operation, respectively. For each benchmark circuit, 5000 critical target paths delay distributions of different dimensions have been considered.

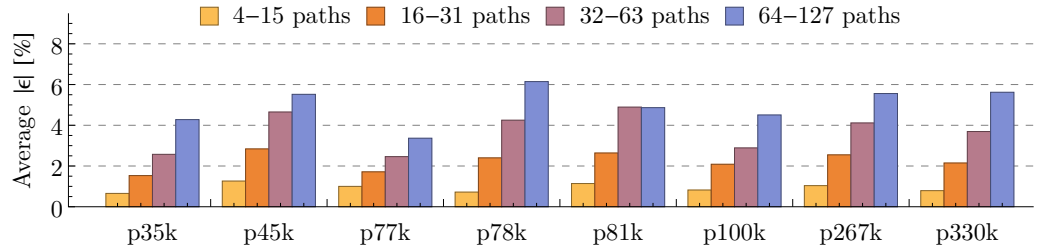
As expected, the runtime for the normal and skew-normal distribution based MAX-operation was almost exactly the same as the runtime observed for random multivariate normal distributions in fig. 7.11. Even the computation of the maximum of several hundreds of path delays, which requires several hundreds of applications of the skew-normal distribution based MAX-operation, can be computed in a couple of milliseconds. Hence, the proposed skew-normal distribution based MAX-operation is well suited for block-based statistical timing analysis.

The average absolute error (eq. (7.17)) of the normal and skew-normal distribution based MAX-operation is presented in fig. 7.13 in percent [%] for different numbers

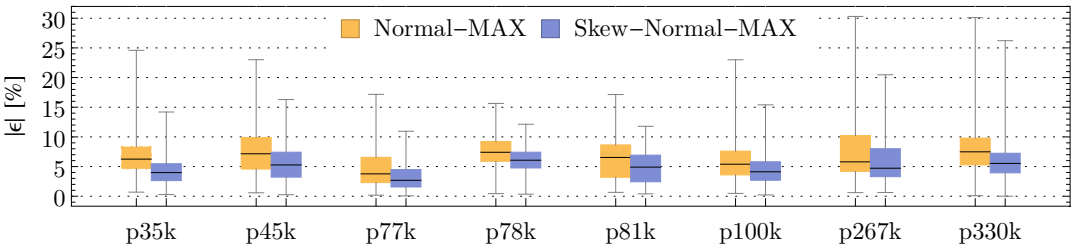
of critical target paths. It can be seen that on average, the skew-normal distribution based MAX-operation reduces the approximation error by up to 50%. In particular the maximum absolute error is significantly reduced, as shown in the box plot in fig. 7.13c for between 64 and 127 critical target path delays. This error will be further reduced in section 7.5.3.



(a) Using normal distribution based MAX-operation



(b) Using skew-normal distribution based MAX-operation



(c) Box plot of absolute error for $64 \leq n \leq 127$

▲ **Figure 7.13** — Absolute error $|\epsilon|$ of the approximation of the maximum $\max(X_1, \ldots, X_n)$ of critical target path delays $X_1, \ldots, X_n$

The detailed numerical results are shown in table B.6. The columns "$|\epsilon_n|$" and "$|\epsilon_{sn}|$" show the average absolute error of the normal and skew-normal distribution based MAX-operation, respectively. The next column "$\frac{|\epsilon_{sn}|-|\epsilon_n|}{|\epsilon_n|}$" shows the relative absolute error of the proposed skew-normal distribution based MAX-operation, which is defined by eq. (7.18). The results show that the proposed skew-normal distribution based MAX-operation reduces the average absolute error by up to 50%. The last two columns "T_n" and "T_{sn}" show the runtime for the repeated application of the normal and skew-normal distribution based MAX-operation, respectively. For very large dimensions $n > 1000$, the runtime is dominated by the initial Cholesky factorization of the covariance matrix Σ, which has $O(n^3)$ worst case runtime complexity.

7.5.3 Analysis and Further Reduction of Approximation Error

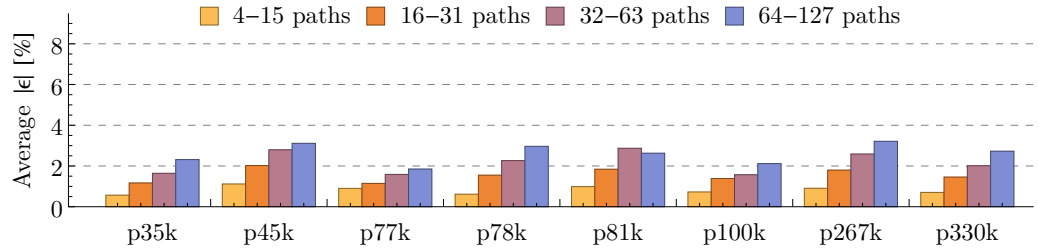
The small error of the skew-normal distribution based MAX-operation was found to be mainly caused by three reasons. At first, the exact distribution of $\max(X_1, \ldots, X_n)$ might not be accurately approximated by any skew-normal distribution. For example, if the random variables have the same mean but very different standard deviations. However, such extreme cases rarely occur in statistical timing analysis.

The second main reason for the approximation error lies within the efficient design of the proposed algorithm. The proposed algorithm computes the shape vector $\hat{\boldsymbol{\lambda}}$ from the dominant eigenvalue ψ and corresponding eigenvector of the matrix $\tilde{H}$, which is defined in eq. (6.136). If the dominant eigenvalue is not at least 10 times larger than the next smaller eigenvalue, then the estimation of $\hat{\lambda}$ may be inaccurate. It was frequently observed that the dominant eigenvalue was of the same magnitude than the next smaller eigenvalue, in particular during the experiments in section 7.5.2.

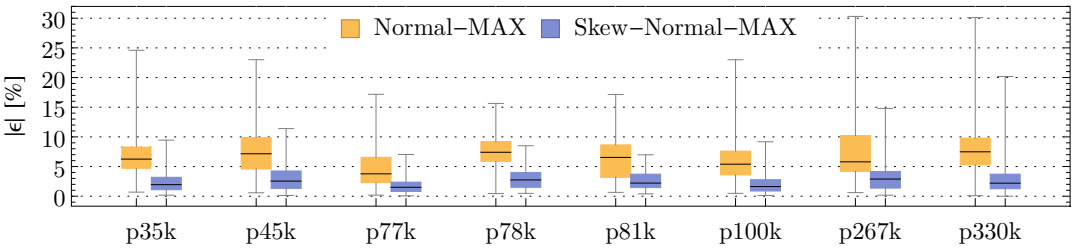
The third main reason is that the shape parameter $\hat{\boldsymbol{\lambda}}$ of a skew-normal distribution cannot be chosen independently from the covariance matrix $\hat{\boldsymbol{\Sigma}}$ because $\hat{\boldsymbol{\lambda}}^T \hat{\boldsymbol{\Sigma}}^{-1} \hat{\boldsymbol{\lambda}} < \frac{2}{\pi - 2}$ must hold according to definition 6.2. This condition is satisfied if and only if the dominant eigenvalue ψ satisfies eq. (6.87). The error caused by the third main reason was particularly pronounced for the computation of the maximum delay of the target paths in section 7.5.2 such that $\mathbb{E}(\max(X_1, \ldots, X_n))$ was significantly overestimated.

In order to improve the accuracy of the results, it is possible to ignore this constraint because the proposed algorithm is also applicable if only the subvector (X_{n-1}, X_n) of the last two random variables of $\boldsymbol{X}$ has a bivariate skew-normal distribution. In additional experiments it was found that this relaxation can significantly reduce the approximation error, but the variance $\mathrm{Var}(\max(X_{n-1}, X_n))$ in the bottom right corner of $\hat{\boldsymbol{\Sigma}}$ may become so small that the covariance matrix $\hat{\boldsymbol{\Sigma}}$ is no longer positive definite. To avoid this, the idea is to multiply the entire covariance matrix $\boldsymbol{\Sigma}$ by a positive scalar $x \in \mathbb{R}^+$ slightly smaller than one (e.g. $x = 0.98$) every time the dominant eigenvalue ψ is significantly reduced (e.g. by more than 20%) in order to satisfy eq. (6.136). If $\hat{\boldsymbol{\Sigma}}$ is positive definite then theorem 2.37 with A being a diagonal matrix with all diagonal elements equal to x implies that $A\hat{\boldsymbol{\Sigma}}A^T$ and therefore $x\hat{\boldsymbol{\Sigma}}$ is positive definite. After scaling the covariance matrix, the inverse Cholesky factor $\hat{L}^{-1}$ can be efficiently updated by noticing that $x\hat{\boldsymbol{\Sigma}} = (\sqrt{x}\hat{L})(\sqrt{x}\hat{L})^T$ and $(\sqrt{x}\hat{L})^{-1} = x^{-1/2}\hat{L}^{-1}$.

If applied carefully, the above described scaling of the covariance matrix is a simple but effective way to consider the strong non-linear relationships between the random variables in $\boldsymbol{Y}$, which cannot be accurately represented by a multivariate skew-normal distribution. However, the scaling inevitably decreases the variance of the approximation of $\max(X_1, \ldots, X_n)$, which must be compensated during the last application of the skew-normal distribution based MAX-operation. This compensation is achieved by multiplying $\hat{\boldsymbol{\Sigma}}$ with x^{-k}, where k denotes the number of applications of the skew-normal distribution based MAX-operation during which the dominant eigenvalue ψ was significantly reduced to satisfy eq. (6.87). By using the proposed covariance matrix scaling, the average absolute error $|\epsilon|$ is reduced by up to 70%, as shown in fig. 7.14.



(a) Using skew-normal distribution based MAX-operation



(b) Box plot of absolute error for $64 \leq n \leq 127$

▲ **Figure 7.14 —** Absolute error $|\epsilon|$ of the approximation of $\max(X_1, \ldots, X_n)$ using covariance matrix scaling, where $X_1, \ldots, X_n$ are critical target path delays

The detailed numerical results are presented in table B.7. The results show that the average absolute error $|\epsilon|$ can be reduced by up to 70% by the proposed skew-normal distribution based MAX-operation with covariance matrix scaling, compared to the normal distribution based MAX-operation. For example, the average absolute error $|\epsilon|$ of computing the distribution of the maximum of between 256 and 512 critical path delays is reduced from 9% to only 2.7% for benchmark circuit p77k.

Compared to the results in table B.6 obtained without covariance matrix scaling, the average absolute error $|\epsilon|$ is consistently reduced. However, the scaling of the covariance matrix and the inverse Cholesky factor slightly increases the runtime of the skew-normal distribution based MAX-operation. But the increase in runtime is very small because the covariance matrix scaling is only performed for very few applications of the skew-normal distribution based MAX-operation.

7.5.4 Empirical Runtime Complexity of Algorithm for MAX-Operation

This subsection investigates the runtime of the proposed algorithm for the skew-normal distribution based MAX-operation for very large dimensions of the skew-normal random vector $\boldsymbol{X} = (X_1, \ldots, X_n)^T \sim \mathcal{SN}_n(\boldsymbol{\mu}, \Sigma, \boldsymbol{\lambda})$. The goal is to test that the asymptotic behaviour of the implementation matches the theoretical worst-case runtime complexity of the algorithm for the skew-normal distribution based MAX-operation. For this, the skew-normal distribution based MAX-operation was applied to approximate the distribution of the random vector

$$(X_1, \ldots, X_{n-2}, \max(X_{n-1}, X_n))^T \tag{7.22}$$

with a $(n-1)$-dimensional skew-normal distribution. Except for the very first approximation step, the inverse Cholesky factor L^{-1} is obtained by updating the inverse Cholesky factor of the previous application of the skew-normal distribution based MAX-operation. To represent this typical scenario, it is assumed that the inverse Cholesky factor L^{-1} is available, and hence, the runtime of the initial computation of L^{-1} with $LL^T = \Sigma$ was ignored. As a consequence, the runtime appears to be lower than in fig. 7.11.

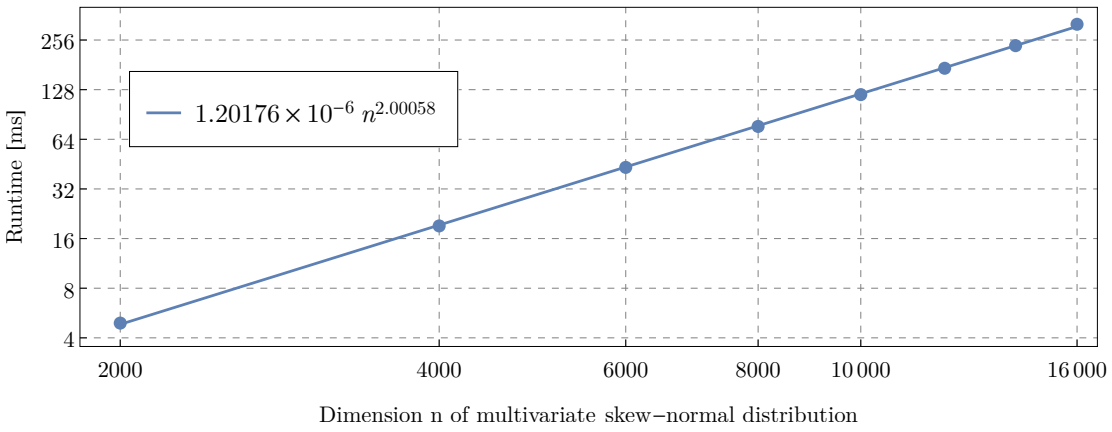
The runtime of the algorithm in section 6.5.3 is measured for very large dimensions

$$n \in \{2000, 4000, 6000, 8000, 10000, 12000, 14000, 16000\}.$$

Therefore, even the smallest size ($n = 2000$) is large enough for the high order terms to dominate all other terms. The result of these measurements consists of 8 data points (n_i, t_i) for $1 \leq i \leq 8$, as shown in fig. 7.15. In order to predict the runtime $t(n)$ with the power-law function

$$t(n) = an^b, \tag{7.23}$$

linear regression is applied to the set of data points $(\log(n_i), \log(t_i))$. The resulting coefficients are $a \approx 1.20176 \times 10^{-6}$ and $b \approx 2$, which confirms the theoretical quadratic worst case runtime complexity $O(n^2)$ of the proposed algorithm for the skew-normal distribution based MAX-operation.



▲ **Figure 7.15 —** Log-log plot of the runtime of the proposed algorithm for the skew-normal distribution based MAX-operation for a huge number of random variables

CHAPTER 8

Conclusions

Tremendous advances in semiconductor process technology have created new delay test challenges for digital integrated circuits. The complexity of state-of-the-art manufacturing processes does not only exacerbate the problem of process variability, it also makes today's integrated circuits more prone to defects such as resistive shorts and opens. The resulting large delay variations severely degrade the quality and reliability of all delay tests. A delay test might detect a delay fault of fixed size in only a subset of all manufactured circuits, which can potentially result in a large number of test escapes.

Statistical timing analysis is an integral component of any variation-aware delay test generation method and is required to analyse and predict the effectiveness of delay tests in a population of circuits which are functionally identical but have varying timing properties. Efficient statistical timing analysis of large circuits is a well known hard problem. Under the impact of delay variations, a path is sensitized by a particular test vector-pair only with some probability. Furthermore, the event that any of the sensitized paths has a path delay fault is also described by a probability. Recent variation-aware delay test generation methods must therefore be guided by the probability that at least one of the sensitized paths has a path delay fault. One of the most challenging problems of statistical timing analysis is the efficient computation of the statistical MAX-operation. The normal distribution based MAX-operation [Clark61] approximates the skewed distribution of the result with a normal distribution, leading to large approximation errors.

8.1 Contributions of this Work

This work targets key statistical timing analysis problems, which arise in delay test applications for innovative technology nodes. Novel and efficient statistical timing

analysis algorithms for path and small delay fault testing applications have been presented. In addition, accurate skew-normal distribution based SUM and MAX-operations are proposed, which provide the foundation for the efficient statistical delay fault simulation.

Probabilistic sensitization analysis is proposed to guide the delay test generation process into generating path delay fault tests, which are more tolerant towards delay variations. The analysis not only provides the location of the gate which blocks the propagation of the transition along the target path, but it also identifies those paths that are responsible for the invalidation of the delay test. Further important quality characteristics of the given test vector-pair can be efficiently computed by a Monte-Carlo simulation of a small subcircuit, which is constructed by the proposed analysis.

For the detection of small delay faults, two efficient algorithms for the computation of the target paths delay fault probability are proposed. The non-incremental algorithm provides high accuracy but may become inefficient if the delay test parameters are frequently modified. To minimize the computational cost after delay test parameter modifications, an efficient incremental algorithm has also been presented, which is suitable for the inner loop of automatic test pattern generation methods. Compared to extensive Monte-Carlo simulations, the experimental results show a very large speedup with only a small approximation error, which is mainly due to the impact of delay variations on the sensitization of the critical paths.

To minimize the error of block-based statistical timing analysis, the more accurate skew-normal distribution based SUM and MAX-operations are introduced. Compared to the normal distribution based MAX-operation [Clark61], the proposed MAX-operation is defined on the far more flexible skew-normal distribution, which allows the accurate approximation of the result with another skew-normal distribution. Although the worst case runtime complexity of the proposed algorithm is quadratic in the number of random variables, the runtime remains very small even for several hundreds of random variables. The superior accuracy and low runtime makes the skew-normal distribution based MAX-operation ideal for block-based statistical timing analysis and many other statistical timing analysis problems.

The experimental results demonstrate the high efficiency of the proposed algorithms.

8.2 Ongoing Research and Future Work

While the probabilistic sensitization analysis presented in chapter 4 can easily consider crosstalk and power droop effects, a non-trivial extension [Tang14b] would be required in chapter 5 to adapt the joint delay distributions of the critical target paths to consider these effects during the computation of the target paths delay fault probability.

In general, gate level models are designed for high speed simulation and provide only limited accuracy. If only small process variations need to be considered during statistical timing analysis, then the gate delays can usually be accurately described by a normal distribution. However, with increasing process variability, the circuit timing behaviour

is largely determined by the impact of these variations on the electrical properties of transistors and interconnects, which leads to skewed gate delay distributions and the complex nonlinear relationships between the gate delays become more pronounced [Zhang05]. In order to get meaningful results from statistical timing analysis, more realistic gate and interconnect models at transistor level must be applied in an efficient manner. In particular, these models must consider the complex relationships between the electrical parameter variation of gates and interconnects, based on a given circuit layout. To efficiently solve the statistical timing analysis problems, the techniques and algorithms developed in this thesis must be adopted for large and complex circuit models on transistor level.

Modern VLSI circuits are prone to defects, that affect the electrical properties of logic gates and interconnects. With continued shrinking of semiconductor feature size, ever more new defect mechanisms are discovered, which require accurate analysis on transistor level [Hapke11, Tang14c].

In [Tang14c], the authors perform mutation analysis to study the impact of various types of defects on the electrical properties of standard cells, such as INV, NAND, NOR and XOR. The authors evaluate these defects by generating mutants considering various mutations such as incorrect connections of polarity devices, splitting high degree nodes into two and keeping a transistor conducting/non-conducting. Afterwards, the electrical properties of these mutants were analysed using SPICE simulations. The experimental results show that a large fraction of defects in CMOS gates manifest themselves as stuck-at-unknown (X). If U_{dd} denotes the supply voltage, then stuck-at-unknown means that the gate output voltage remains close to $0.5U_{dd}$ and does not stabilize to either logic one or logic zero. This is for example caused by a bridging fault between the input and the output of an inverter. Clearly, this fault behaviour can only be accurately represented at electrical level.

Another important future work is the application of the proposed techniques and algorithms as part of a delay test generation process. This development will lead to efficient delay test generation methods, which can reduce the test cost while improving the delay test quality under the impact of large process variations.

Mathematical Details

A.1 Moments involving the Maximum $\max(X_{n-1}, X_n)$

This section defines all moments, which are required to compute the mean vector, the covariance matrix and the third multivariate cumulant of the random vector

$$\boldsymbol{Y} = (X_1, \ldots, X_{n-2}, \max(X_{n-1}, X_n))^T, \tag{A.1}$$

where $\boldsymbol{X} = (X_1, \ldots, X_n)^T$ is a n-dimensional skew-normal random vector. For compactness of notation, let $b := \sqrt{2/\pi}$.

The formulas for computing the moments that involve the maximum $\max(X_{n-1}, X_n)$ have been obtained from a probabilistic representation of the multivariate skew-normal distribution based on the multivariate normal distribution. The following proposition is adopted from [Azzal99, Proposition 1]:

Proposition A.1. *Suppose that*

$$\begin{pmatrix} Z_0 \\ \boldsymbol{Z} \end{pmatrix} \sim \mathcal{N}_{n+1}(\boldsymbol{0}, \Omega^*), \quad \Omega^* = \begin{pmatrix} 1 & \delta^T \\ \delta & \bar{\Omega} \end{pmatrix} \tag{A.2}$$

where Z_0 is a scalar component and Ω^* is a correlation matrix. Then

$$\boldsymbol{X} = \begin{cases} +\boldsymbol{Z} & \text{if } Z_0 > 0 \\ -\boldsymbol{Z} & \text{if } Z_0 \leq 0, \end{cases} \tag{A.3}$$

is a n-dimensional skew-normal random vector with parameters $\xi = \boldsymbol{0}$, $\Omega = \bar{\Omega}$ and

$$\boldsymbol{\alpha} = \frac{\bar{\Omega}^{-1}\delta}{\sqrt{1 - \delta^T \bar{\Omega}^{-1} \delta}}. \tag{A.4}$$

Let $(X_i, X_j, X_k, X_l)^T$ denote a 4-dimensional subvector of the random vector $\boldsymbol{X}$ with Azzalini-parameterization $\boldsymbol{\xi} = (\xi_i, \xi_j, \xi_k, \xi_l)^T$, $\boldsymbol{\delta} = (\delta_i, \delta_j, \delta_k, \delta_l)^T$ and

$$\Omega = \begin{pmatrix} \omega_i^2 & \omega_{i,j} & \omega_{i,k} & \omega_{i,l} \\ \omega_{i,j} & \omega_j^2 & \omega_{j,k} & \omega_{j,l} \\ \omega_{i,k} & \omega_{j,k} & \omega_k^2 & \omega_{k,l} \\ \omega_{i,l} & \omega_{j,l} & \omega_{k,l} & \omega_l^2 \end{pmatrix}.$$

The moments $\mathbb{E}(\max(X_i, X_j)^m)$ for $m = 1, 2, \ldots$ can be computed as follows. Similar to the approach in section 5.4, eq. (A.3) and the definition of $\max(x_i, x_j)$ over the real numbers induces a partition of the sample space Θ:

$$\begin{aligned} A_1 &= \{\theta \in \Theta : Z_0(\theta) > 0, Z_i(\theta) > Z_j(\theta)\} \\ A_2 &= \{\theta \in \Theta : Z_0(\theta) > 0, Z_i(\theta) \leq Z_j(\theta)\} \\ A_3 &= \{\theta \in \Theta : Z_0(\theta) \leq 0, Z_i(\theta) > Z_j(\theta)\} \\ A_4 &= \{\theta \in \Theta : Z_0(\theta) \leq 0, Z_i(\theta) \leq Z_j(\theta)\}, \end{aligned}$$

were Z_0 and $\boldsymbol{Z} = (Z_1, \ldots, Z_n)^T$ are as defined in proposition A.1. Using eq. (2.1), $\mathbb{E}(\max(X_i, X_j)^m)$ can formally be written as

$$\begin{aligned} \mathbb{E}(\max(X_i, X_j)^m) = &\mathbb{E}(X_i(\theta)^m \mathbb{1}_{A_1}(\theta)) \\ &+ \mathbb{E}(X_j(\theta)^m \mathbb{1}_{A_2}(\theta)) \\ &+ \mathbb{E}((-X_j(\theta))^m \mathbb{1}_{A_3}(\theta)) \\ &+ \mathbb{E}((-X_i(\theta))^m \mathbb{1}_{A_4}(\theta)), \end{aligned} \tag{A.5}$$

where Z_i and Z_j have been replaced by $\pm X_i$ and $\pm X_j$, respectively, according to eq. (A.3). Then theorem 2.43 implies that

$$\begin{aligned} \mathbb{E}(\max(X_i, X_j)^m) = &\mathbb{E}\left(X_i^m \,|A_1\right)\mathbb{P}(A_1) \\ &+ \mathbb{E}\left(X_j^m \,|A_2\right)\mathbb{P}(A_2) \\ &+ \mathbb{E}\left((-X_j)^m \,|A_3\right)\mathbb{P}(A_3) \\ &+ \mathbb{E}\left((-X_i)^m \,|A_4\right)\mathbb{P}(A_4), \end{aligned} \tag{A.6}$$

where θ has been omitted for compactness of notation. The formulas for the computation of the above moments have been presented in [Talli61] and [Arism13]. After extensive simplifications and using the following definitions

$$a = \sqrt{\omega_i^2 - 2\omega_{i,j} + \omega_j^2} \tag{A.7}$$

$$s_1 = (\xi_i - \xi_j)/a \tag{A.8}$$

$$s_2 = (\delta_i \omega_i - \delta_j \omega_j)/a \tag{A.9}$$

$$s_3 = s_1/\sqrt{1 - s_2^2}, \tag{A.10}$$

the first three moments of $\max(X_i, X_j)$ are

$$\begin{aligned}\mathbb{E}(\max(X_i, X_j)^1) &= \mathbb{E}(X_j^1) + 2(\xi_i - \xi_j)\Phi_2(s_1, 0; s_2) + b(\delta_i\omega_i - \delta_j\omega_j)\Phi(s_3) \\ &+ 2a\phi(s_1)\Phi(-s_2 s_3) \end{aligned} \tag{A.11}$$

$$\begin{aligned}\mathbb{E}(\max(X_i, X_j)^2) &= \mathbb{E}(X_j^2) + 2\left((\xi_i^2 + \omega_i^2) - (\xi_j^2 + \omega_j^2)\right)\Phi_2(s_1, 0; s_2) \\ &+ 2a(\xi_i + \xi_j)\phi(s_1)\Phi(-s_2 s_3) + 2b(\xi_i\omega_i\delta_i - \xi_j\omega_j\delta_j)\Phi(s_3) \\ &+ ab\sqrt{1 - s_2^2}(\delta_i\omega_i + \delta_j\omega_j)\phi(s_3) \end{aligned} \tag{A.12}$$

$$\begin{aligned}\mathbb{E}(\max(X_i, X_j)^3) &= \mathbb{E}(X_j^3) + 2\left((\xi_i^3 + 3\xi_i\omega_i^2) - (\xi_j^3 + 3\xi_j\omega_j^2)\right)\Phi_2(s_1, 0; s_2) \\ &+ \frac{2}{a}\left(a^2(\xi_i^2 + \xi_i\xi_j + \xi_j^2) + a^2\omega_j^2 + 2\omega_i^2(\omega_i^2 - \omega_{i,j}) - \omega_{i,j}^2 + \omega_j^4\right)\phi(s_1)\Phi(-s_2 s_3) \\ &+ b\left(3(\xi_i^2 + \omega_i^2)\omega_i\delta_i - \delta_i^3\omega_i^3 - 3(\xi_j^2 + \omega_j^2)\omega_j\delta_j + \delta_j^3\omega_j^3\right)\Phi(s_3) \\ &+ ab\sqrt{1 - s_2^2}\left((2\xi_i + \xi_j)\omega_i\delta_i + (\xi_i + 2\xi_j)\omega_j\delta_j\right)\phi(s_3), \end{aligned} \tag{A.13}$$

where Φ_2 is given by eq. (2.65) and can be efficiently computed using the algorithm proposed in [Genz04]. The moments $\mathbb{E}(X_j^1)$, $\mathbb{E}(X_j^2)$ and $\mathbb{E}(X_j^3)$ can be computed with eqs. (6.9), (6.10) and (6.13), respectively.

The computation of the remaining moments can be reduced to the computation of $\mathbb{E}(\max(X_i, X_j)^2)$ and $\mathbb{E}(\max(X_i, X_j)^3)$. For example, the formula for the computation of $\mathbb{E}(\max(X_i, X_j)X_k)$ is obtained by considering the sum

$$\max(X_i, X_j) + sX_k = \max(X_i + sX_k, X_j + sX_k)$$

with $s \in \mathbb{R}$. Using theorem 6.3, the parameters of the 2-dimensional skew-normal random vector

$$(W_1, W_2)^T := (X_i + sX_k, X_j + sX_k)^T \tag{A.14}$$

are determined as a function of s. Afterwards, the moment $\mathbb{E}(\max(W_1, W_2)^2)$ is evaluated symbolically with eq. (A.12) and expanded using the linearity of the expectation operator to obtain

$$\begin{aligned}\mathbb{E}(\max(W_1, W_2)^2) &= \mathbb{E}((\max(X_i, X_j) + sX_k)^2) && \text{(A.15)} \\ &= \mathbb{E}(\max(X_i, X_j)^2 + 2s\max(X_i, X_j)X_k + s^2X_k^2) && \text{(A.16)} \\ &= \mathbb{E}(\max(X_i, X_j)^2) + 2s\mathbb{E}(\max(X_i, X_j)X_k) + s^2\mathbb{E}(X_k^2). && \text{(A.17)}\end{aligned}$$

Hence, the moment $\mathbb{E}(\max(W_1, W_2)^2)$ is a second order polynomial of s and the moment $\mathbb{E}(\max(X_i, X_j)X_k)$ is easily obtained from the second coefficient of this polynomial.

The moments $\mathbb{E}(\max(X_i, X_j)^2 X_k)$ and $\mathbb{E}(\max(X_i, X_j) X_k^2)$ are obtained in the same way from the symbolic evaluation of $\mathbb{E}(\max(W_1, W_2)^3)$. The solutions are:

$$\begin{aligned}
&\mathbb{E}(\max(X_i, X_j)X_k) = \mathbb{E}(X_j X_k) + 2(as_1\xi_k + \omega_{i,k} - \omega_{j,k})\Phi_2(s_1, 0; s_2) \\
&+ 2a\xi_k\phi(s_1)\Phi(-s_2 s_3) + ab(s_2\xi_k + s_1\omega_k\delta_k)\Phi(s_3) + ab\sqrt{1 - s_2^2}\omega_k\delta_k\phi(s_3)
\end{aligned} \tag{A.18}$$

$$\begin{aligned}
&\mathbb{E}(\max(X_i, X_j)X_k^2) = \mathbb{E}(X_j X_k^2) + 2(2\xi_k(\omega_{i,k} - \omega_{j,k}) + as_1(\xi_k^2 + \omega_k^2))\Phi_2(s_1, 0; s_2) \\
&+ 2/a((\omega_{i,k} - \omega_{j,k})^2 + a^2(\xi_k^2 + \omega_k^2))\phi(s_1)\Phi(-s_2 s_3) \\
&+ b(as_2\xi_k^2 + 2\omega_k\delta_k(as_1\xi_k + \omega_{i,k} - \omega_{j,k}) - as_2(\delta_k^2 - 1)\omega_k^2)\Phi(s_3) \\
&+ 4\sqrt{1 - s_2^2}\xi_k\omega_k\delta_k a\phi(s_3)/\sqrt{2\pi}
\end{aligned} \tag{A.19}$$

$$\begin{aligned}
&\mathbb{E}(\max(X_i, X_j)^2 X_k) = \mathbb{E}(X_j^2 X_k) \\
&+ 2(\xi_i^2\xi_k + 2\xi_i\omega_{i,k} - \xi_k(\xi_j^2 - \omega_i^2 + \omega_j^2) - 2\xi_j\omega_{j,k})\Phi_2(s_1, 0; s_2) \\
&+ (2a^2 s_1\xi_k + 4/a((\omega_i^2 - \omega_{i,j})\omega_{i,k} + (\omega_j^2 - \omega_{i,j})\omega_{j,k} + \xi_j\xi_k a^2))\phi(s_1)\Phi(-s_2 s_3) \\
&+ b\Big(2\omega_i\delta_i(\xi_i\xi_k + \omega_{i,k}) - 2\omega_j\delta_j(\xi_j\xi_k + \omega_{j,k}) - \omega_i^2\delta_i^2\omega_k\delta_k \\
&+ \omega_k\delta_k(\xi_i^2 - \xi_j^2 + \omega_i^2 - \omega_j^2 + \omega_j^2\delta_j^2)\Big)\Phi(s_3) \\
&+ 2a\sqrt{1 - s_2^2}(\xi_k(\omega_i\delta_i + \omega_j\delta_j) + \omega_k\delta_k(\xi_i + \xi_j))\phi(s_3)/\sqrt{2\pi}
\end{aligned} \tag{A.20}$$

Similarly, the moment $\mathbb{E}(\max(X_i, X_j)X_k X_l)$ is obtained by considering the sum

$$\max(X_i, X_j) + sX_k + tX_l = \max(X_i + sX_k + tX_l, X_j + sX_k + tX_l)$$

with $s, t \in \mathbb{R}$ and symbolic evaluation of eq. (A.13). The solution is:

$$\begin{aligned}
&\mathbb{E}(\max(X_i, X_j)X_k X_l) = \mathbb{E}(X_j X_k X_l) \\
&+ 2(\xi_l(\omega_{i,k} - \omega_{j,k}) + \xi_k(\omega_{i,l} - \omega_{j,l}) + as_1(\xi_k\xi_l + \omega_{k,l}))\Phi_2(s_1, 0; s_2) \\
&+ 2/a((\omega_{i,k} - \omega_{j,k})(\omega_{i,l} - \omega_{j,l}) + a^2(\xi_k\xi_l + \omega_{k,l}))\phi(s_1)\Phi(-s_2 s_3) \\
&+ b\Big(as_2(\xi_k\xi_l + \omega_{k,l} - \delta_k\delta_l\omega_k\omega_l) + \delta_k\omega_k(\xi_i\xi_l - \xi_j\xi_l + \omega_{i,l} - \omega_{j,l}) \\
&+ \delta_l\omega_l(\xi_i\xi_k - \xi_j\xi_k + \omega_{i,k} - \omega_{j,k})\Big)\Phi(s_3) \\
&+ 2a\sqrt{1 - s_2^2}(\delta_k\xi_l\omega_k + \delta_l\xi_k\omega_l)\phi(s_3)/\sqrt{2\pi}.
\end{aligned} \tag{A.21}$$

A.2 Proofs

This section presents the proofs for the skew-normal distribution based SUM and MAX-operation.

A.2.1 Statistical SUM-Operation

Theorem 6.3. *If $\boldsymbol{X} \sim \mathcal{SN}_n(\boldsymbol{\mu}, \Sigma, \boldsymbol{\lambda})$, $\boldsymbol{b} \in \mathbb{R}^k$ and $A \in \mathbb{R}^{k \times n}$ has rank k, then the random vector*

$$\boldsymbol{Y} = A\boldsymbol{X} + \boldsymbol{b} \tag{6.61}$$

has a k-dimensional skew-normal distribution $\mathcal{SN}_k(\boldsymbol{\mu}_Y, \Sigma_Y, \boldsymbol{\lambda}_Y)$ with parameters

$$\boldsymbol{\mu}_Y = A\boldsymbol{\mu} + \boldsymbol{b} \tag{6.62}$$

$$\Sigma_Y = A\Sigma A^T \tag{6.63}$$

$$\boldsymbol{\lambda}_Y = A\boldsymbol{\lambda}. \tag{6.64}$$

Proof. The moment generating function $M_X(\boldsymbol{t}) = \mathbb{E}(\exp(\boldsymbol{t}^T\boldsymbol{X}))$ of the skew-normal random vector $\boldsymbol{X}$ is given by eq. (6.31) as

$$M_X(\boldsymbol{t}) = 2\exp\left(\boldsymbol{t}^T(\boldsymbol{\mu} - \boldsymbol{\lambda}) + \frac{1}{2}\boldsymbol{t}^T(\Sigma + \boldsymbol{\lambda}\boldsymbol{\lambda}^T)\boldsymbol{t}\right)\Phi\left(\boldsymbol{\lambda}^T\boldsymbol{t}/b\right). \tag{A.22}$$

Then the moment generating function $M_Y(\boldsymbol{t})$ of the random vector $\boldsymbol{Y}$ is

$$\begin{aligned}
M_Y(\boldsymbol{t}) &= \mathbb{E}(\exp(\boldsymbol{t}^T\boldsymbol{Y})) = \mathbb{E}(\exp(\boldsymbol{t}^T(A\boldsymbol{X} + \boldsymbol{b}))) \\
&= \exp(\boldsymbol{t}^T\boldsymbol{b})\mathbb{E}(\exp(\boldsymbol{t}^T A\boldsymbol{X})) \\
&= \exp(\boldsymbol{t}^T\boldsymbol{b})M_X(A^T\boldsymbol{t}) \\
&= 2\exp\left((A^T\boldsymbol{t})^T(\boldsymbol{\mu} - \boldsymbol{\lambda}) + \frac{1}{2}(A^T\boldsymbol{t})^T(\Sigma + \boldsymbol{\lambda}\boldsymbol{\lambda}^T)A^T\boldsymbol{t} + \boldsymbol{t}^T\boldsymbol{b}\right)\Phi\left(\boldsymbol{\lambda}^T A^T\boldsymbol{t}/b\right)
\end{aligned}$$

To proof the theorem and compute the parameters $(\boldsymbol{\mu}_Y, \Sigma_Y, \boldsymbol{\lambda}_Y)$, $M_Y(\boldsymbol{t})$ must be brought into the structure

$$M_Y(\boldsymbol{t}) = 2\exp\left(\boldsymbol{t}^T(\boldsymbol{\mu}_Y - \boldsymbol{\lambda}_Y) + \frac{1}{2}\boldsymbol{t}^T(\Sigma_Y + \boldsymbol{\lambda}_Y\boldsymbol{\lambda}_Y^T)\boldsymbol{t}\right)\Phi\left(\boldsymbol{\lambda}_Y^T\boldsymbol{t}/b\right). \tag{A.23}$$

From

$$\Phi\left(\boldsymbol{\lambda}_Y^T\boldsymbol{t}/b\right) = \Phi\left(\boldsymbol{\lambda}^T A^T\boldsymbol{t}/b\right) \tag{A.24}$$

it follows that $\boldsymbol{\lambda}_Y = A\boldsymbol{\lambda}$. Afterwards,

$$(A^T\boldsymbol{t})^T(\boldsymbol{\mu} - \boldsymbol{\lambda}) + \frac{1}{2}(A^T\boldsymbol{t})^T(\Sigma + \boldsymbol{\lambda}\boldsymbol{\lambda}^T)A^T\boldsymbol{t} + \boldsymbol{t}^T\boldsymbol{b} \tag{A.25}$$

$$= \boldsymbol{t}^T(A\boldsymbol{\mu} - A\boldsymbol{\lambda}) + \frac{1}{2}\boldsymbol{t}^T(A\Sigma A^T + A\boldsymbol{\lambda}\boldsymbol{\lambda}^T A^T)\boldsymbol{t} + \boldsymbol{t}^T\boldsymbol{b} \tag{A.26}$$

$$= \boldsymbol{t}^T(A\boldsymbol{\mu} + \boldsymbol{b} - \boldsymbol{\lambda}_Y) + \frac{1}{2}\boldsymbol{t}^T(A\Sigma A^T + \boldsymbol{\lambda}_Y\boldsymbol{\lambda}_Y^T)\boldsymbol{t} \tag{A.27}$$

$$= \boldsymbol{t}^T(\boldsymbol{\mu}_Y - \boldsymbol{\lambda}_Y) + \frac{1}{2}\boldsymbol{t}^T(\Sigma_Y + \boldsymbol{\lambda}_Y\boldsymbol{\lambda}_Y^T)\boldsymbol{t} \tag{A.28}$$

and therefore, $\boldsymbol{\mu}_Y = A\boldsymbol{\mu} + \boldsymbol{b}$ and $\Sigma_Y = A\Sigma A^T$. Furthermore, Σ_Y is symmetric and also positive definite according to theorem 2.4 and

$$\boldsymbol{\lambda}_Y^T\Sigma_Y^{-1}\boldsymbol{\lambda}_Y = \boldsymbol{\lambda}^T A^T(A^{-T}\Sigma^{-1}A^{-1})A\boldsymbol{\lambda} = \boldsymbol{\lambda}^T\Sigma^{-1}\boldsymbol{\lambda} < \frac{2}{\pi - 2} \tag{A.29}$$

holds by assumption. □

A.2.2 Statistical MAX-Operation

Theorem 6.4. *If $\boldsymbol{X}$ is a n-dimensional random vector with finite third multivariate cumulant $\kappa_3(\boldsymbol{X})$ and $A \in \mathbb{R}^{k\times n}$ has rank k, then the third multivariate cumulant $\kappa_3(\boldsymbol{Y}) \in \mathbb{R}^{k^2\times k}$ of the random vector $\boldsymbol{Y} = A\boldsymbol{X}$ is*

$$\kappa_3(\boldsymbol{Y}) = (A \otimes A)\kappa_3(\boldsymbol{X})A^T, \tag{6.75}$$

where $\otimes$ denotes the Kronecker product.

Proof. Let $\boldsymbol{\mu}_{\boldsymbol{X}}$ and $\boldsymbol{\mu}_{\boldsymbol{Y}}$ denote the mean vector of $\boldsymbol{X}$ and $\boldsymbol{Y}$ respectively. According to definition 2.40 and eq. (2.53), the third multivariate cumulant of $\boldsymbol{X}$ is defined as

$$\kappa_3(\boldsymbol{X}) = \mathbb{E}((\boldsymbol{X} - \boldsymbol{\mu}_{\boldsymbol{X}}) \otimes (\boldsymbol{X} - \boldsymbol{\mu}_{\boldsymbol{X}}) \otimes (\boldsymbol{X} - \boldsymbol{\mu}_{\boldsymbol{X}})^T) \tag{A.30}$$

and the third multivariate cumulant of $\boldsymbol{Y} = A\boldsymbol{X}$ is

$$\begin{aligned}\kappa_3(\boldsymbol{Y}) &= \mathbb{E}((\boldsymbol{Y} - \boldsymbol{\mu}_{\boldsymbol{Y}}) \otimes (\boldsymbol{Y} - \boldsymbol{\mu}_{\boldsymbol{Y}}) \otimes (\boldsymbol{Y} - \boldsymbol{\mu}_{\boldsymbol{Y}})^T) &\text{(A.31)}\\ &= \mathbb{E}(A(\boldsymbol{X} - \boldsymbol{\mu}_{\boldsymbol{X}}) \otimes A(\boldsymbol{X} - \boldsymbol{\mu}_{\boldsymbol{X}}) \otimes (\boldsymbol{X} - \boldsymbol{\mu}_{\boldsymbol{X}})^T A^T). &\text{(A.32)}\end{aligned}$$

The above equation can be transformed using eq. (2.10), which gives

$$\kappa_3(\boldsymbol{Y}) = \mathbb{E}((A \otimes A)((\boldsymbol{X} - \boldsymbol{\mu}_{\boldsymbol{X}}) \otimes (\boldsymbol{X} - \boldsymbol{\mu}_{\boldsymbol{X}})) \otimes (\boldsymbol{X} - \boldsymbol{\mu}_{\boldsymbol{X}})^T A^T). \tag{A.33}$$

By using the linearity of the expectation operator it follows that

$$\begin{aligned}\kappa_3(\boldsymbol{Y}) &= (A \otimes A)\mathbb{E}((\boldsymbol{X} - \boldsymbol{\mu}_{\boldsymbol{X}}) \otimes (\boldsymbol{X} - \boldsymbol{\mu}_{\boldsymbol{X}}) \otimes (\boldsymbol{X} - \boldsymbol{\mu}_{\boldsymbol{X}})^T)A^T &\text{(A.34)}\\ &= (A \otimes A)\kappa_3(\boldsymbol{X})A^T, &\text{(A.35)}\end{aligned}$$

which completes the proof. □

Theorem 6.5. *Let $\boldsymbol{X} \sim \mathcal{SN}_n(\boldsymbol{\mu}, \Sigma, \boldsymbol{\lambda})$ denote a n-dimensional skew-normal random vector. If $\kappa_3(\boldsymbol{V})$ is the third standardized multivariate cumulant of $\boldsymbol{X}$, then the matrix*

$$M_{\boldsymbol{X}} = \kappa_3(\boldsymbol{V})^T\kappa_3(\boldsymbol{V}) \tag{6.80}$$

has an eigenvector

$$v = \pm\frac{L^{-1}\boldsymbol{\lambda}}{||L^{-1}\boldsymbol{\lambda}||} \tag{6.81}$$

corresponding to the only non-zero eigenvalue

$$\psi = \left(2 - \frac{\pi}{2}\right)^2 (\boldsymbol{\lambda}^T\Sigma^{-1}\boldsymbol{\lambda})^3 = \left(2 - \frac{\pi}{2}\right)^2 ||L^{-1}\boldsymbol{\lambda}||^6, \tag{6.82}$$

where L denotes the lower Cholesky factor of Σ such that $\Sigma = LL^T$.

Proof. According to eq. (6.38) and theorem 6.4, the third standardized multivariate cumulant of a skew-normal random vector $\boldsymbol{X}$ is

$$\begin{aligned}\kappa_3(\boldsymbol{V}) &= (L^{-1} \otimes L^{-1})\kappa_3(\boldsymbol{X})L^{-T}, \\ &= \left(2 - \frac{\pi}{2}\right)(L^{-1} \otimes L^{-1})(\boldsymbol{\lambda} \otimes \boldsymbol{\lambda} \otimes \boldsymbol{\lambda}^T)L^{-T}\end{aligned} \tag{A.36}$$

which can be simplified by using eqs. (2.8) and (2.10) to

$$\begin{aligned}\kappa_3(\boldsymbol{V}) &= \left(2 - \frac{\pi}{2}\right)(L^{-1}\boldsymbol{\lambda} \otimes L^{-1}\boldsymbol{\lambda} \otimes \boldsymbol{\lambda}^T L^{-T}) \\ &= \left(2 - \frac{\pi}{2}\right)(L^{-1}\boldsymbol{\lambda} \otimes L^{-1}\boldsymbol{\lambda})(L^{-1}\boldsymbol{\lambda})^T.\end{aligned} \tag{A.37}$$

Starting from the definition of the skewness matrix of $\boldsymbol{X}$ given by eq. (6.80) and using eq. (A.37), the resulting expression is then simplified by the application of eqs. (2.9) and (2.10) to obtain

$$\begin{aligned}M_{\boldsymbol{X}} &= \kappa_3(\boldsymbol{V})^T\kappa_3(\boldsymbol{V}) \\ &= \left(2 - \frac{\pi}{2}\right)^2 (L^{-1}\boldsymbol{\lambda})(L^{-1}\boldsymbol{\lambda} \otimes L^{-1}\boldsymbol{\lambda})^T(L^{-1}\boldsymbol{\lambda} \otimes L^{-1}\boldsymbol{\lambda})(L^{-1}\boldsymbol{\lambda})^T \\ &= \left(2 - \frac{\pi}{2}\right)^2 (L^{-1}\boldsymbol{\lambda})((L^{-1}\boldsymbol{\lambda})^T \otimes (L^{-1}\boldsymbol{\lambda})^T)(L^{-1}\boldsymbol{\lambda} \otimes L^{-1}\boldsymbol{\lambda})(L^{-1}\boldsymbol{\lambda})^T \\ &= \left(2 - \frac{\pi}{2}\right)^2 (L^{-1}\boldsymbol{\lambda})(\boldsymbol{\lambda}^T\Sigma^{-1}\boldsymbol{\lambda} \otimes \boldsymbol{\lambda}^T\Sigma^{-1}\boldsymbol{\lambda})(L^{-1}\boldsymbol{\lambda})^T \\ &= \left(2 - \frac{\pi}{2}\right)^2 (L^{-1}\boldsymbol{\lambda})(\boldsymbol{\lambda}^T\Sigma^{-1}\boldsymbol{\lambda})^2(L^{-1}\boldsymbol{\lambda})^T.\end{aligned} \tag{A.38}$$

Hence, $M_{\boldsymbol{X}}$ is a symmetric real valued matrix of rank one with exactly one positive eigenvalue ψ. By definition there must be a non-zero $\boldsymbol{v}$ such that $M_{\boldsymbol{X}}\boldsymbol{v} = \psi\boldsymbol{v}$. This means that

$$\left(2 - \frac{\pi}{2}\right)^2 (\boldsymbol{\lambda}^T\Sigma^{-1}\boldsymbol{\lambda})^2(L^{-1}\boldsymbol{\lambda})(L^{-1}\boldsymbol{\lambda})^T\boldsymbol{v} = \psi\boldsymbol{v}, \tag{A.39}$$

where the expression $(L^{-1}\boldsymbol{\lambda})^T\boldsymbol{v}$ on the LHS is a scalar. Using $\psi \neq 0$ it follows that $\boldsymbol{v}$ must be a scalar multiple of $L^{-1}\boldsymbol{\lambda}$, specifically

$$\boldsymbol{v} = \frac{\left(2 - \frac{\pi}{2}\right)^2 (\boldsymbol{\lambda}^T\Sigma^{-1}\boldsymbol{\lambda})^2(L^{-1}\boldsymbol{\lambda})^T\boldsymbol{v}}{\psi} L^{-1}\boldsymbol{\lambda}. \tag{A.40}$$

Therefore, the normalized eigenvector is

$$\boldsymbol{v} = \pm\frac{L^{-1}\boldsymbol{\lambda}}{||L^{-1}\boldsymbol{\lambda}||} = \frac{L^{-1}\boldsymbol{\lambda}}{\sqrt{\boldsymbol{\lambda}^T\Sigma^{-1}\boldsymbol{\lambda}}}. \tag{A.41}$$

Replacing $\boldsymbol{v}$ in eq. (A.40) with the above expression gives

$$\frac{L^{-1}\boldsymbol{\lambda}}{||L^{-1}\boldsymbol{\lambda}||} = \left(\frac{\left(2 - \frac{\pi}{2}\right)^2 (\boldsymbol{\lambda}^T\Sigma^{-1}\boldsymbol{\lambda})^2(L^{-1}\boldsymbol{\lambda})^T L^{-1}\boldsymbol{\lambda}}{\psi}\right) \frac{L^{-1}\boldsymbol{\lambda}}{||L^{-1}\boldsymbol{\lambda}||} \tag{A.42}$$

and therefore

$$1 = \frac{\left(2 - \frac{\pi}{2}\right)^2 (\boldsymbol{\lambda}^T \Sigma^{-1} \boldsymbol{\lambda})^2 (L^{-1}\boldsymbol{\lambda})^T L^{-1}\boldsymbol{\lambda}}{\psi}, \tag{A.43}$$

so that the only non-zero eigenvalue is

$$\psi = \left(2 - \frac{\pi}{2}\right)^2 (\boldsymbol{\lambda}^T \Sigma^{-1} \boldsymbol{\lambda})^3 = \left(2 - \frac{\pi}{2}\right)^2 ||L^{-1}\boldsymbol{\lambda}||^6, \tag{A.44}$$

which completes the proof. □

A.2.3 Quadratic Time Algorithm for MAX-operation

Lemma 6.7. *Let* $\boldsymbol{X} = (X_1, \ldots, X_n)^T$ *denote a n-dimensional random vector and let the random vector* $\boldsymbol{Y} = (Y_1, \ldots, Y_{n-1})^T$ *be defined as*

$$\boldsymbol{Y} = (X_1, \ldots, X_{n-2}, \max(X_{n-1}, X_n))^T. \tag{6.109}$$

If $\boldsymbol{X}$ *has a restricted skew-normal distribution, then*

$$\mathbb{E}(\bar{Y}_i \bar{Y}_j \bar{Y}_k) = 0 \tag{6.110}$$

for all $i, j, k \in \mathbb{N}$ *with* $1 \leq i \leq n-4$ *and* $1 \leq j, k \leq n-1$, *where* $\bar{\boldsymbol{Y}} = (\bar{Y}_1, \ldots, \bar{Y}_{n-1})^T$ *denotes the centred random vector corresponding to* $\boldsymbol{Y}$.

Proof. For any $j \in \mathbb{N}$ with $1 \leq j \leq n-1$, let $\hat{\mu}_j := \mathbb{E}(Y_j)$ and let $\bar{Y}_j := Y_j - \hat{\mu}_j$ denote the centred random variable corresponding to the random variable Y_j. In order to proof this lemma, it must be shown that

$$\mathbb{E}(\bar{Y}_i \bar{Y}_j \bar{Y}_k) = 0 \tag{A.45}$$

holds for any choice of $i, j, k \in \mathbb{N}$ with $1 \leq i \leq n-4$ and $1 \leq j, k \leq n-1$. According to the properties of the restricted skew-normal distribution,

$$\lambda_i = b\omega_i\delta_i = 0 \tag{A.46}$$

$$\omega_{n,i} = \omega_{n-1,i}, \tag{A.47}$$

must hold for any $1 \leq i \leq n-4$ and eq. (A.46) implies $\delta_i = 0$ because $\omega_i > 0$.

For any choice of $i, j, k \in \mathbb{N}$ with $1 \leq i \leq n-4$ and $1 \leq j, k \leq n-2$, the correctness of eq. (A.45) can easily be shown using eq. (6.36) with $\mathbb{E}(\bar{Y}_i) = \mathbb{E}(\bar{Y}_j) = \mathbb{E}(\bar{Y}_k) = 0$ and $\lambda_i = 0$.

To complete the proof, it must be shown that eq. (A.45) also holds if it involves $Y_{n-1} = \max(X_{n-1}, X_n)$. For this, k is fixed to $k := n-1$ and it is shown that the moments $\mathbb{E}(\bar{Y}_i \bar{Y}_{n-1}^2)$, $\mathbb{E}(\bar{Y}_i^2 \bar{Y}_{n-1})$ and $\mathbb{E}(\bar{Y}_i \bar{Y}_j \bar{Y}_{n-1})$ evaluate to zero. It follows from eq. (6.49), $\mathbb{E}(\bar{Y}_i) = 0$ and $\lambda_i = 0$ that the location parameter $\bar{\xi}_i$ of the centred random variable $\bar{Y}_i$ is $\bar{\xi}_i = 0$ for all $1 \leq i \leq n-4$.

The formula to compute the moment $\mathbb{E}(\bar{Y}_i\bar{Y}_{n-1}^2)$ is given by eq. (A.20), which simplifies by using $\bar{\xi}_i = 0$, $\delta_i = 0$ and eq. (A.47) to

$$\begin{aligned}\mathbb{E}(\bar{Y}_i\bar{Y}_{n-1}^2) = 2\omega_{n-1,i}\,\big[&\bar{\xi}_n + b\omega_n\delta_n + b(\delta_{n-1}\omega_{n-1} - \delta_n\omega_n)\Phi(s_3)\\ &+ 2a\phi(s_1)\Phi(-s_2s_3) + 2(\bar{\xi}_{n-1} - \bar{\xi}_n)\Phi_2(s_1, 0; s_2)\big]\,.\end{aligned}$$

Then $\bar{\xi}_{n-1} := \xi_{n-1} - \hat{\mu}_{n-1}$ and $\bar{\xi}_n := \xi_n - \hat{\mu}_{n-1}$ implies that $\mathbb{E}(\bar{Y}_i\bar{Y}_{n-1}^2) = 0$.

The formula to compute the moment $\mathbb{E}(\bar{Y}_i^2\bar{Y}_{n-1})$ is given by eq. (A.19), which simplifies by using $\bar{\xi}_i = 0$, $\delta_i = 0$ and eq. (A.47) to

$$\begin{aligned}\mathbb{E}(\bar{Y}_i^2\bar{Y}_{n-1}) = \omega_i^2\,\big[&\bar{\xi}_n + b\delta_n\omega_n + b(\delta_{n-1}\omega_{n-1} - \delta_n\omega_n)\Phi(s_3)\\ &+ 2a\phi(s_1)\Phi(-s_2s_3) + 2(\bar{\xi}_{n-1} - \bar{\xi}_n)\Phi_2(s_1, 0; s_2)\big]\,.\end{aligned}$$

Then $\bar{\xi}_{n-1} := \xi_{n-1} - \hat{\mu}_{n-1}$ and $\bar{\xi}_n := \xi_n - \hat{\mu}_{n-1}$ implies that $\mathbb{E}(\bar{Y}_i^2\bar{Y}_{n-1}) = 0$.

The last required moment $\mathbb{E}(\bar{Y}_i\bar{Y}_j\bar{Y}_{n-1})$ can be computed by using eq. (A.21). Then by using $\bar{\xi}_i = 0$, $\delta_i = 0$ and eq. (A.47), eq. (A.21) simplifies to

$$\begin{aligned}&\mathbb{E}(\bar{Y}_i\bar{Y}_j\bar{Y}_{n-1}) = -b\omega_{n-1,i}\omega_j\delta_j + \bar{\xi}_n\omega_{j,i} + b(\omega_{n-1,i}\omega_j\delta_j + \omega_{j,i}\omega_n\delta_n)\\ &+ b(\omega_{n-1}\delta_{n-1} - \omega_n\delta_n)\omega_{j,i}\Phi(s_3) + 2\omega_{j,i}(a\phi(s_1)\Phi(-s_2s_3) + (\bar{\xi}_{n-1} - \bar{\xi}_n)\Phi_2(s_1, 0; s_2)).\end{aligned}$$

Then $\bar{\xi}_{n-1} := \xi_{n-1} - \hat{\mu}_{n-1}$ and $\bar{\xi}_n := \xi_n - \hat{\mu}_{n-1}$ implies that $\mathbb{E}(\bar{Y}_i\bar{Y}_j\bar{Y}_{n-1}) = 0$. □

Theorem 6.8. *Let $\boldsymbol{X} = (X_1, \ldots, X_n)^T$ denote a n-dimensional random vector and let the random vector $\boldsymbol{Y}$ be defined as $\boldsymbol{Y} = (X_1, \ldots, X_{n-2}, \max(X_{n-1}, X_n))^T$. Furthermore, let $\hat{\Sigma}$ denote the covariance matrix of $\boldsymbol{Y}$ with Cholesky factorization $\hat{\Sigma} = \hat{L}\hat{L}^T$ and let $\kappa_3(\boldsymbol{V})$ denote the third multivariate cumulant of the random vector*

$$\boldsymbol{V} = (X_{n-3}, X_{n-2}, \max(X_{n-1}, X_n))^T. \tag{6.112}$$

If $\boldsymbol{X}$ has a restricted skew-normal distribution, then the skewness matrix of $\boldsymbol{Y}$ is of the form

$$M_{\boldsymbol{Y}} = \begin{pmatrix} \boldsymbol{0}_{n-4,n-4} & \boldsymbol{0}_{n-4,3} \\ \boldsymbol{0}_{3,n-4} & G \end{pmatrix}, \tag{6.113}$$

where $\boldsymbol{0}_{k,l}$ is the $(k \times l)$ zero matrix,

$$G = \hat{L}_{22}^{-1}\kappa_3(\boldsymbol{V})^T(\hat{L}_{22}^{-T} \otimes \hat{L}_{22}^{-T})(\hat{L}_{22}^{-1} \otimes \hat{L}_{22}^{-1})\kappa_3(\boldsymbol{V})\hat{L}_{22}^{-T} \tag{6.114}$$

and $\hat{L}_{22}^{-T}$ is the bottom-right corner 3×3 submatrix of $\hat{L}^{-1}$.

Proof. Let $\bar{\boldsymbol{Y}}$ denote the centred random vector (see eq. (6.73)) corresponding to $\boldsymbol{Y}$ and consider the partition of the centred random vector

$$\bar{\boldsymbol{Y}} = \begin{pmatrix} \bar{\boldsymbol{U}} \\ \bar{\boldsymbol{V}} \end{pmatrix}, \tag{A.48}$$

into the $(n-4)$-dimensional skew-normal random vector $\bar{\boldsymbol{U}}$ and the 3-dimensional random vector $\bar{\boldsymbol{V}}$. The third multivariate cumulant of $\boldsymbol{Y}$, which is defined by definition 2.40, can be written in block matrix form as

$$\kappa_3(\boldsymbol{Y}) = \mathbb{E}\left(\begin{pmatrix} \bar{\boldsymbol{U}} \\ \bar{\boldsymbol{V}} \end{pmatrix} \otimes \bar{\boldsymbol{Y}} \otimes \begin{pmatrix} \bar{\boldsymbol{U}} \\ \bar{\boldsymbol{V}} \end{pmatrix}^T\right) = \mathbb{E}\left(\begin{pmatrix} \bar{\boldsymbol{U}} \otimes \bar{\boldsymbol{Y}} \otimes \bar{\boldsymbol{U}}^T & \bar{\boldsymbol{U}} \otimes \bar{\boldsymbol{Y}} \otimes \bar{\boldsymbol{V}}^T \\ \bar{\boldsymbol{V}} \otimes \bar{\boldsymbol{Y}} \otimes \bar{\boldsymbol{U}}^T & \bar{\boldsymbol{V}} \otimes \bar{\boldsymbol{Y}} \otimes \bar{\boldsymbol{V}}^T \end{pmatrix}\right) \tag{A.49}$$

$$= \begin{pmatrix} \mathbb{E}(\bar{\boldsymbol{U}} \otimes \bar{\boldsymbol{Y}} \otimes \bar{\boldsymbol{U}}^T) & \mathbb{E}(\bar{\boldsymbol{U}} \otimes \bar{\boldsymbol{Y}} \otimes \bar{\boldsymbol{V}}^T) \\ \mathbb{E}(\bar{\boldsymbol{V}} \otimes \bar{\boldsymbol{Y}} \otimes \bar{\boldsymbol{U}}^T) & \mathbb{E}(\bar{\boldsymbol{V}} \otimes \bar{\boldsymbol{Y}} \otimes \bar{\boldsymbol{V}}^T) \end{pmatrix}. \tag{A.50}$$

From lemma 6.7 it follows that

$$\kappa_3(\boldsymbol{Y}) = \begin{pmatrix} \mathbf{0}_{(n-4)(n-1),n-4} & \mathbf{0}_{(n-4)(n-1),3} \\ \mathbf{0}_{3(n-1),n-4} & \mathbb{E}(\bar{\boldsymbol{V}} \otimes \bar{\boldsymbol{Y}} \otimes \bar{\boldsymbol{V}}^T) \end{pmatrix}. \tag{A.51}$$

Let

$$\hat{L}^{-1} = \begin{pmatrix} \hat{L}_{11}^{-1} & \mathbf{0}_{n-4,3} \\ \hat{L}_{21}^{-1} & \hat{L}_{22}^{-1} \end{pmatrix} \tag{A.52}$$

denote a partitioning of the inverse lower Cholesky factor with $\hat{L}_{11}^{-1} \in \mathbb{R}^{(n-4)\times(n-4)}$, $\hat{L}_{21}^{-1} \in \mathbb{R}^{3\times(n-4)}$, $\hat{L}_{22}^{-1} \in \mathbb{R}^{3\times 3}$ and $\mathbf{0}_{k,l}$ is the $(k \times l)$ zero matrix. Then the third standardized multivariate cumulant can be written as

$$\begin{aligned} \kappa_3(\boldsymbol{Z}) &= (\hat{L}^{-1} \otimes \hat{L}^{-1})\kappa_3(\boldsymbol{Y})\hat{L}^{-T} \\ &= \begin{pmatrix} \hat{L}_{11}^{-1} \otimes \hat{L}^{-1} & \mathbf{0}_{(n-4)(n-1),3(n-1)} \\ \hat{L}_{21}^{-1} \otimes \hat{L}^{-1} & \hat{L}_{22}^{-1} \otimes \hat{L}^{-1} \end{pmatrix} \kappa_3(\boldsymbol{Y}) \begin{pmatrix} \hat{L}_{11}^{-1} & \mathbf{0}_{n-4,3} \\ \hat{L}_{21}^{-1} & \hat{L}_{22}^{-1} \end{pmatrix}^T \\ &= \begin{pmatrix} \hat{L}_{11}^{-1} \otimes \hat{L}^{-1} & \mathbf{0}_{(n-4)(n-1),3(n-1)} \\ \hat{L}_{21}^{-1} \otimes \hat{L}^{-1} & \hat{L}_{22}^{-1} \otimes \hat{L}^{-1} \end{pmatrix} \begin{pmatrix} \mathbf{0}_{(n-4)(n-1),n-4} & \mathbf{0}_{(n-4)(n-1),3} \\ \mathbf{0}_{3(n-1),n-4} & \mathbb{E}(\bar{\boldsymbol{V}} \otimes \bar{\boldsymbol{Y}} \otimes \bar{\boldsymbol{V}})\hat{L}_{22}^{-T} \end{pmatrix} \\ &= \begin{pmatrix} \mathbf{0}_{(n-4)(n-1),n-4} & \mathbf{0}_{(n-4)(n-1),3} \\ \mathbf{0}_{3(n-1),n-4} & (\hat{L}_{22}^{-1} \otimes \hat{L}^{-1})\mathbb{E}(\bar{\boldsymbol{V}} \otimes \bar{\boldsymbol{Y}} \otimes \bar{\boldsymbol{V}}^T)\hat{L}_{22}^{-T} \end{pmatrix}. \end{aligned} \tag{A.53}$$

Then according to eq. (6.83), the skewness matrix is

$$\begin{aligned} M &= \kappa_3(\boldsymbol{Z})^T\kappa_3(\boldsymbol{Z}) \\ &= \begin{pmatrix} \mathbf{0}_{n-4,n-4} & \mathbf{0}_{n-4,3} \\ \mathbf{0}_{3,n-4} & G \end{pmatrix}, \end{aligned} \tag{A.54}$$

where the submatrix $G \in \mathbb{R}^{3\times 3}$ is

$$\begin{aligned} G &= \left((\hat{L}_{22}^{-1} \otimes \hat{L}^{-1})\mathbb{E}(\bar{\boldsymbol{V}} \otimes \bar{\boldsymbol{Y}} \otimes \bar{\boldsymbol{V}}^T)\hat{L}_{22}^{-T}\right)^T (\hat{L}_{22}^{-1} \otimes \hat{L}^{-1})\mathbb{E}(\bar{\boldsymbol{V}} \otimes \bar{\boldsymbol{Y}} \otimes \bar{\boldsymbol{V}}^T)\hat{L}_{22}^{-T} \\ &= \hat{L}_{22}^{-1}\mathbb{E}(\bar{\boldsymbol{V}} \otimes \bar{\boldsymbol{Y}} \otimes \bar{\boldsymbol{V}}^T)^T(\hat{L}_{22}^{-T} \otimes \hat{L}^{-T})(\hat{L}_{22}^{-1} \otimes \hat{L}^{-1})\mathbb{E}(\bar{\boldsymbol{V}} \otimes \bar{\boldsymbol{Y}} \otimes \bar{\boldsymbol{V}}^T)\hat{L}_{22}^{-T}. \end{aligned} \tag{A.55}$$

This expression can be further simplified by noticing that the matrix $\mathbb{E}(\bar{\boldsymbol{V}} \otimes \bar{\boldsymbol{Y}} \otimes \bar{\boldsymbol{V}}^T)$ contains moments involving the components $\bar{U}_1, \ldots, \bar{U}_{n-4}$ of the random vector $\bar{\boldsymbol{U}}$. According to lemma 6.7, these moments are zero and can therefore be omitted so that

$$G = \hat{L}_{22}^{-1} \kappa_3(\boldsymbol{V})^T (\hat{L}_{22}^{-T} \otimes \hat{L}_{22}^{-T})(\hat{L}_{22}^{-1} \otimes \hat{L}_{22}^{-1}) \kappa_3(\boldsymbol{V}) \hat{L}_{22}^{-T}, \tag{A.56}$$

which completes the proof. □

Lemma 6.9. *Let $\boldsymbol{X} \sim \mathcal{SN}_n(\boldsymbol{\mu}, \Sigma, \boldsymbol{\lambda})$ be a n-dimensional skew-normal random vector and let $c_j := Cov(X_{n-1}, X_j) - Cov(X_n, X_j)$ for all $1 \leq j \leq n$. If $c_{n-3}\lambda_{n-2} - c_{n-2}\lambda_{n-3} \neq 0$, then there exists an invertible linear transformation, which maps $\boldsymbol{X} = (X_1, \ldots, X_n)^T$ to a n-dimensional random vector $\boldsymbol{W} = (W_1, \ldots, W_n)^T$ with*

$$W_i = \begin{cases} X_i + q_i X_{n-2} + r_i X_{n-3} & \text{for } 1 \leq i \leq n-4 \\ X_i & \text{for } n-3 \leq i \leq n, \end{cases} \tag{6.126}$$

such that $\boldsymbol{W}$ has a restricted skew-normal distribution.

Proof. In order to satisfy the conditions of the restricted skew-normal distribution, the coefficients q_i and r_i must be chosen such that the system of linear equations

$$\mathrm{Cov}(X_{n-1}, X_i + q_i X_{n-2} + r_i X_{n-3}) - \mathrm{Cov}(X_n, X_i + q_i X_{n-2} + r_i X_{n-3}) = 0 \tag{A.57}$$

$$q_i \lambda_{n-2} + r_i \lambda_{n-3} + \lambda_i = 0. \tag{A.58}$$

is satisfied for all $i \in \mathbb{N}$ with $1 \leq i \leq n-4$. The solution is

$$q_i = -\frac{c_{n-3}\lambda_i - c_i \lambda_{n-3}}{c_{n-3}\lambda_{n-2} - c_{n-2}\lambda_{n-3}} \tag{A.59}$$

$$r_i = -\frac{c_i \lambda_{n-2} - c_{n-2}\lambda_i}{c_{n-3}\lambda_{n-2} - c_{n-2}\lambda_{n-3}}, \tag{A.60}$$

where $c_i = \mathrm{Cov}(X_{n-1}, X_i) - \mathrm{Cov}(X_n, X_i)$. A unique solution always exists under the precondition $c_{n-3}\lambda_{n-2} - c_{n-2}\lambda_{n-3} \neq 0$ of the lemma.

It remains to be shown that the transformation is invertible. The linear transformation in this lemma can be written as $\boldsymbol{W} = A\boldsymbol{X}$, where A is a $n \times n$ square matrix of the form

$$A = \begin{pmatrix} I_{n-4} & C \\ \mathbf{0}_{4,n-4} & I_4 \end{pmatrix}, \tag{A.61}$$

where $\mathbf{0}_{4,n-4}$ is the $4 \times (n-4)$ zero matrix and

$$C = \begin{pmatrix} q_1 & r_1 & 0 & 0 \\ q_2 & r_2 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ q_{n-5} & r_{n-5} & 0 & 0 \\ q_{n-4} & r_{n-4} & 0 & 0 \end{pmatrix}. \tag{A.62}$$

The inverse of any block matrix

$$\begin{pmatrix} B_1 & B_2 \\ B_3 & B_4 \end{pmatrix} \tag{A.63}$$

with square submatrices B_1 and B_4, $\det(B_1) \neq 0$ and $\det(S_{B_1}) \neq 0$ is

$$\begin{pmatrix} B_1^{-1} + B_1^{-1}B_2S_{B_1}^{-1}B_3B_1^{-1} & -B_1^{-1}B_2S_{B_1}^{-1} \\ -S_{B_1}^{-1}B_3B_1^{-1} & S_{B_1}^{-1} \end{pmatrix} \tag{A.64}$$

where $S_{B_1} = B_4 - B_3B_1^{-1}B_2$ denotes the Schur complement of B_1 [Choi09]. Setting $B_1 = I_{n-4}$, $B_2 = C$, $B_3 = \mathbf{0}_{4,n-4}$ and $B_4 = I_4$ implies $S_{B_1} = I_4$ and after simplifications, the inverse of A is

$$A^{-1} = \begin{pmatrix} I_{n-4} & -C \\ \mathbf{0}_{4,n-4} & I_4 \end{pmatrix}. \tag{A.65}$$

As a consequence, the inverse linear transformation can be obtained by switching the signs of the coefficients q_i and r_i for all $i \in \mathbb{N}$ with $1 \leq i \leq n-4$. □

Additional Result Tables

▼ **Table B.1 —** Average results for construction and Monte-Carlo simulation of subcircuits $\bar{\mathcal{S}}$ and $\mathcal{S}$, compared to Monte-Carlo Simulation of complete circuit considering subcircuit size, accuracy and runtime for subcircuit construction and simulation.

		subcircuit $\bar{\mathcal{S}}$					representative subcircuit $\mathcal{S}$					
circuit name	T_{clk} [ps]	#sen.crit. paths	|cone| [%]	$\lvert\bar{\mathcal{S}}\rvert$ [%]	P_{err} [%]	SU	#sen.crit. paths	|cone| [%]	$\lvert\mathcal{S}\rvert$ [%]	#iter	P_{err} [%]	SU
	1163.6	11.1	40.92	0.28	7.67	127.3	127.9	49.72	1.25	23.3	0.11	25.4
p35k	1266.8	8.9	37.63	0.24	4.73	137.5	114.7	48.82	1.15	21.8	0.05	27.0
	1412.7	6.0	32.00	0.19	1.51	127.3	81.4	47.30	0.98	19.4	0.01	21.6
	889.3	19.2	10.23	0.81	27.39	63.1	145.4	14.50	3.62	61.3	0.02	11.4
p45k	968.4	11.7	7.97	0.50	11.63	85.7	91.8	11.49	2.96	64.3	0.01	13.0
	1084.8	3.3	3.42	0.16	0.68	114.4	21.2	7.33	1.28	39.1	0.01	14.2
	5805.2	2.9	2.69	0.10	2.03	202.5	198.4	3.99	1.08	29.9	0.03	17.2
p77k	6377.2	2.1	2.28	0.07	1.89	218.4	140.0	3.59	0.94	28.0	0.01	18.5
	7137.5	1.2	1.45	0.05	0.92	207.6	76.0	3.58	0.73	21.8	0.00	15.8
	1213.3	72.8	16.66	1.74	43.55	41.1	680.3	39.87	13.71	233.4	0.01	3.7
p78k	1325.4	23.8	6.52	0.68	9.85	92.7	218.4	17.70	5.54	159.9	0.01	8.0
	1485.2	3.5	1.18	0.13	0.39	257.2	22.5	3.40	0.89	46.5	0.01	21.9
	1018.5	118.6	21.72	2.17	75.13	27.3	1612.1	24.31	12.43	176.7	0.02	3.6
p81k	1109.7	60.2	18.51	1.25	61.58	45.8	869.9	22.92	11.02	250.9	0.01	3.7
	1238.6	14.5	8.65	0.31	3.80	144.8	177.6	18.79	4.99	270.8	0.00	7.2
	1400.1	28.7	9.10	0.71	62.44	72.4	429.8	11.73	4.71	160.9	0.01	7.6
p100k	1533.9	14.1	6.49	0.39	30.22	106.6	210.2	9.07	3.11	144.9	0.00	10.4
	1710.6	3.2	2.42	0.11	1.15	178.0	33.4	4.76	0.96	63.1	0.02	22.4
	787.9	293.4	11.43	1.07	52.38	47.5	1001.2	16.98	2.90	95.6	0.01	11.1
p267k	856.2	132.9	6.40	0.53	31.38	75.3	508.4	10.43	1.62	77.7	0.00	15.6
	951.6	29.2	1.99	0.15	1.74	102.5	143.2	3.51	0.49	38.3	0.01	16.7
	1023.5	254.5	17.26	1.59	78.49	36.0	2134.2	22.37	7.86	220.0	0.01	4.6
p330k	1116.9	107.4	12.43	0.81	57.99	62.6	1012.9	17.18	5.77	253.9	0.00	5.8
	1246.4	18.3	3.75	0.15	1.32	156.0	121.5	8.58	1.60	188.9	0.01	13.9

(a) Average results over all test vector-pairs, generated for the detection of path delay faults

		subcircuit $\bar{\mathcal{S}}$					representative subcircuit $\mathcal{S}$					
circuit name	T_{clk} [ps]	#sen.crit. paths	|cone| [%]	$\lvert\bar{\mathcal{S}}\rvert$ [%]	P_{err} [%]	SU	#sen.crit. paths	|cone| [%]	$\lvert\mathcal{S}\rvert$ [%]	#iter	P_{err} [%]	SU
p35k	1412.7	11.2	39.65	0.26	3.07	129.8	61.3	40.74	0.72	12.7	0.03	32.5
p45k	1084.8	7.2	3.61	0.22	4.22	128.0	19.7	6.21	1.02	29.1	0.02	24.3
p77k	7137.5	6.7	3.32	0.15	2.02	170.3	77.2	5.88	0.90	21.1	0.02	22.3
p78k	1485.2	9.2	1.39	0.17	8.14	231.5	30.6	3.34	0.84	40.3	0.02	23.0
p81k	1238.6	13.0	8.54	0.29	6.25	152.9	125.8	17.83	4.79	266.4	0.01	7.5
p100k	1710.6	7.2	1.64	0.11	5.87	183.8	27.0	2.69	0.51	30.9	0.02	29.4
p267k	951.6	21.7	1.25	0.09	1.70	157.3	57.6	2.37	0.27	22.9	0.01	30.0
p330k	1246.4	25.3	3.96	0.16	4.76	157.6	126.8	8.55	1.58	189.8	0.02	14.0

(b) Average results over all test vector-pairs, generated for the detection of small delay faults

▼ **Table B.2 —** Average results of simplified probabilistic sensitization analysis of critical target paths over all test vector-pairs

circuit name	T_{clk} [ps]	sensitization criterion #rob	#nr	#fs	NR-test invalidation P_{ci}	P_{id}	P_{in}	$P_{wi\,|\,in}$	T_{sim} [ms]	T_{diag} [ms]
	1163.64	0.00	7.79	3.32	0.50	1.00	0.60	0.16	1.05	118.46
p35k	1266.85	0.00	6.12	2.78	0.45	1.00	0.63	0.16	1.04	115.88
	1412.74	0.00	4.02	1.96	0.35	1.00	0.64	0.17	1.04	107.77
	889.279	0.00	12.42	6.77	0.85	0.97	0.52	0.19	1.04	141.02
p45k	968.447	0.00	7.71	4.03	0.74	0.97	0.51	0.18	1.03	133.98
	1084.84	0.00	2.24	1.10	0.34	0.95	0.42	0.18	1.03	124.47
	5805.18	0.00	1.61	1.28	0.10	1.00	0.97	0.27	2.36	1487.69
p77k	6377.17	0.00	1.09	1.03	0.08	1.00	0.97	0.27	2.35	1167.63
	7137.51	0.00	0.53	0.68	0.04	1.00	0.95	0.31	2.36	746.36
	1213.31	0.00	58.56	14.24	0.99	1.00	0.59	0.21	5.45	793.89
p78k	1325.36	0.00	19.26	4.59	0.98	1.00	0.58	0.20	5.50	679.63
	1485.22	0.00	2.81	0.65	0.58	1.00	0.54	0.18	5.48	622.47
	1018.51	0.00	89.86	28.70	0.99	0.99	0.70	0.24	3.40	640.68
p81k	1109.69	0.00	46.11	14.09	0.99	1.00	0.67	0.23	3.40	477.42
	1238.59	0.00	11.26	3.25	0.97	1.00	0.65	0.18	3.51	338.22
	1400.08	0.00	24.81	3.91	0.97	1.00	0.80	0.18	2.87	414.81
p100k	1533.9	0.00	12.54	1.57	0.92	1.00	0.79	0.16	2.83	352.58
	1710.59	0.00	2.99	0.22	0.64	1.00	0.74	0.13	2.84	302.82
	787.908	0.06	226.29	79.82	0.99	0.84	0.33	0.10	5.16	894.77
p267k	856.186	0.00	107.03	34.99	0.95	0.89	0.39	0.10	5.14	824.83
	951.598	0.00	28.68	5.99	0.67	0.94	0.53	0.11	5.13	772.77
	1023.52	0.23	196.62	69.44	0.99	0.90	0.51	0.25	8.87	1581.22
p330k	1116.89	0.00	85.99	29.04	0.99	0.97	0.55	0.24	8.79	1363.01
	1246.38	0.00	15.52	5.51	0.95	0.94	0.46	0.20	8.95	1229.80

(a) Path delay fault tests for longest paths in benchmark circuit

circuit name	T_{clk} [ps]	sensitization criterion #rob	#nr	#fs	NR-test invalidation P_{ci}	P_{id}	P_{in}	$P_{wi\,|\,in}$	T_{sim} [ms]	T_{diag} [ms]
p35k	1412.74	0.00	4.48	1.93	0.32	0.97	0.59	0.15	0.96	101.96
p45k	1084.84	0.00	1.16	0.69	0.25	0.98	0.48	0.20	1.02	118.52
p77k	7137.51	0.00	0.73	0.56	0.06	1.00	0.93	0.31	2.33	688.15
p78k	1485.22	0.00	2.81	0.66	0.51	0.91	0.43	0.16	6.02	611.46
p81k	1238.59	0.00	6.66	1.75	0.95	1.00	0.62	0.24	3.40	318.62
p100k	1710.59	0.00	1.19	0.47	0.26	0.94	0.63	0.15	2.77	287.31
p267k	951.598	0.00	10.46	4.16	0.39	0.86	0.32	0.08	4.84	741.11
p330k	1246.38	0.00	15.28	4.62	0.94	0.97	0.47	0.19	8.76	1189.22

(b) Delay tests for marginally detectable small delay faults used in section 7.4

▼ **Table B.3 —** Runtime and error of approximating the delay fault detection probability by the target paths delay fault probability approximation of the non-incremental algorithm

circuit	$\lvert\mathcal{T}\rvert$	$\lvert\Pi\rvert$	δ 10^{-2}	$\lvert\delta\rvert$ 10^{-2}	T_{PA} [s]	T_{MC} [s]	SU
p35k	1	9.47	-2.6	3.3	0.0182	22	1204
	5	45.23	-0.4	1.4	0.0879	54	612
	10	88.96	-0.1	1.2	0.1842	88	475
	20	171.56	0.1	1.0	0.4134	156	377
p45k	1	5.40	-1.8	2.5	0.0125	22	1732
	5	24.15	-0.3	1.4	0.0595	56	948
	10	48.23	0.1	1.0	0.1244	96	776
	20	90.78	0.2	0.8	0.2599	176	677
p77k	1	5.87	-0.8	1.4	0.0159	39	2472
	5	24.64	-0.1	1.0	0.0779	112	1435
	10	47.73	-0.0	0.9	0.1603	198	1234
	20	93.71	0.0	1.0	0.3477	374	1077
p78k	1	8.31	-4.3	5.1	0.0310	86	2772
	5	39.94	-0.4	1.5	0.1554	280	1803
	10	75.58	0.1	1.1	0.3079	516	1676
	20	139.83	0.3	1.0	0.6273	990	1579
p81k	1	11.41	-1.8	2.5	0.0289	80	2753
	5	53.80	-0.1	0.9	0.1510	197	1307
	10	105.40	0.1	0.7	0.3176	339	1067
	20	205.73	0.1	0.7	0.7116	623	876
p100k	1	5.57	-1.9	2.6	0.0199	59	2937
	5	25.63	-0.4	1.4	0.0989	160	1614
	10	51.62	-0.2	1.1	0.2037	280	1376
	20	108.83	0.0	1.1	0.4499	527	1171
p267k	1	18.50	-0.5	1.5	0.0708	130	1843
	5	88.53	0.1	0.9	0.4071	362	889
	10	176.48	-0.8	1.7	0.7743	622	803
	20	333.83	-2.1	2.9	1.5168	1141	752
p330k	1	25.13	-1.6	2.4	0.0938	200	2131
	5	120.61	-0.4	1.4	0.5108	580	1136
	10	238.90	-1.1	2.1	0.9914	1037	1046
	20	479.62	-2.3	3.4	2.0844	1958	940

▼ **Table B.4 —** Runtime and absolute error of approximating the delay fault detection probability by the target paths delay fault probability approximation of the incremental algorithm, after insertion or removal of a test vector-pair

NXP circuit	$\lvert\mathcal{T}\rvert$	Reference	INSERT(test vector-pair)								REMOVE(test vector-pair)		
		Monte-Carlo	sensitization analysis (A)			dist. ext. (B,C)		$\hat{\Psi}$ (D)	*Complete Update*		$\hat{\Psi}$ (D)	*Complete Update*	
		T_{MC} [ms]	$\lvert\Pi\rvert$	$\lvert\delta\rvert$ 10^{-2}	T_{SA} [ms]	$\lvert\varepsilon\rvert$ 10^{-2}	T_{DE} [ms]	T_{NI} [ms]	$\lvert\epsilon\rvert$ 10^{-2}	**Speedup**	T_{NI} [ms]	$\lvert\epsilon\rvert$ 10^{-2}	**Speedup**
p35k	1	21 936	9.47	3.3	5.71	0.0	0.24	0.00	3.3	**3684**	0.00	3.3	**12 681 890**
	5	53 785	45.23	1.4	3.11	0.2	0.91	1.33	1.4	**10 055**	1.33	1.4	**40 459**
	10	87 524	88.96	1.2	2.95	0.2	1.71	13.32	1.2	**4866**	13.32	1.2	**6570**
	20	155 919	171.56	1.0	2.91	0.2	3.25	26.65	1.0	**4753**	26.65	1.0	**5850**
p45k	1	21 629	5.40	2.5	6.05	0.0	0.23	0.00	2.6	**3444**	0.00	2.6	**12 493 342**
	5	56 427	24.15	1.4	5.01	0.2	0.62	1.37	1.4	**8058**	1.37	1.4	**41 190**
	10	96 496	48.23	1.0	5.09	0.3	1.38	12.09	1.0	**5198**	12.09	1.0	**7982**
	20	176 030	90.78	0.8	4.98	0.3	1.56	24.13	0.9	**5739**	24.13	0.9	**7296**
p77k	1	39 213	5.87	1.4	9.16	0.1	0.33	0.00	1.5	**4129**	0.00	1.5	**25 492 799**
	5	111 734	24.64	1.0	8.26	0.9	1.01	1.22	1.6	**10 647**	1.22	1.6	**91 468**
	10	197 825	47.73	0.9	8.09	1.2	1.87	10.95	1.7	**9460**	10.95	1.7	**18 064**
	20	374 394	93.71	1.0	8.15	1.5	3.66	20.01	1.9	**11 764**	20.01	1.9	**18 710**
p78k	1	85 996	8.31	5.1	20.95	0.1	0.14	0.00	5.1	**4077**	0.00	5.1	**45 563 949**
	5	280 091	39.94	1.5	20.33	0.2	0.47	1.48	1.5	**12 571**	1.48	1.5	**188 726**
	10	516 012	75.58	1.1	20.21	0.2	0.83	13.69	1.1	**14 857**	13.69	1.1	**37 680**
	20	990 453	139.83	1.0	19.92	0.2	1.42	27.08	0.9	**20 457**	27.08	0.9	**36 580**
p81k	1	79 622	11.41	2.5	13.64	0.1	0.24	0.00	2.5	**5734**	0.00	2.5	**40 373 912**
	5	197 363	53.80	0.9	12.73	0.1	0.70	1.32	0.9	**13 370**	1.32	0.9	**149 254**
	10	338 725	105.40	0.7	12.69	0.1	1.27	12.34	0.7	**12 882**	12.34	0.7	**27 447**
	20	623 081	205.73	0.7	12.70	0.1	2.39	24.34	0.7	**15 802**	24.34	0.7	**25 595**
p100k	1	58 532	5.57	2.6	13.88	0.0	0.13	0.00	2.6	**4176**	0.00	2.6	**33 935 442**
	5	159 661	25.63	1.4	12.98	0.1	0.55	1.34	1.4	**10 738**	1.34	1.4	**119 177**
	10	280 357	51.62	1.1	12.90	0.2	0.95	12.12	1.2	**10 796**	12.12	1.2	**23 136**
	20	526 962	108.83	1.1	13.00	0.2	2.10	23.71	1.1	**13 577**	23.71	1.1	**22 222**
p267k	1	130 499	18.50	1.5	39.02	0.1	1.28	0.00	1.5	**3238**	0.00	1.5	**66 870 545**
	5	361 989	88.53	0.9	39.10	0.3	5.32	1.35	1.0	**7909**	1.35	1.0	**268 324**
	10	621 799	176.48	1.7	39.04	1.2	10.24	12.64	0.9	**10 041**	12.64	0.9	**49 179**
	20	1 141 305	333.83	2.9	38.94	2.6	16.14	25.22	0.9	**14 212**	25.22	0.9	**45 247**
p330k	1	199 812	25.13	2.4	49.43	0.2	2.13	0.00	2.5	**3875**	0.00	2.5	**86 237 893**
	5	580 496	120.61	1.4	50.03	0.6	11.26	1.40	1.2	**9258**	1.40	1.2	**413 190**
	10	1 037 266	238.90	2.1	50.36	1.4	24.17	13.64	1.0	**11 764**	13.64	1.0	**76 026**
	20	1 958 461	479.62	3.4	50.38	2.8	52.12	27.63	1.0	**15 050**	27.63	1.0	**70 880**

▼ **Table B.5 —** Runtime T and error ϵ for the approximation of the distribution of $\max(X_1, \ldots, X_n)$, where $(X_1, \ldots, X_n) \sim \mathcal{N}_n(\boldsymbol{\mu}, \Sigma)$ with random $\boldsymbol{\mu}$ and random Σ

n	$\bar{\rho}$	$\lvert\epsilon_n\rvert$	$\lvert\epsilon_{sn}\rvert$	$\frac{\lvert\epsilon_{sn}\rvert-\lvert\epsilon_n\rvert}{\lvert\epsilon_n\rvert}$	T_n[ms]	T_{sn}[ms]
4	$0.00 \leq \bar{\rho} < 0.33$	0.0162	0.0015	-90.7%	0.007	0.031
	$0.33 \leq \bar{\rho} < 0.66$	0.0089	0.0010	-89.2%	0.007	0.030
	$0.66 \leq \bar{\rho} < 1.00$	0.0070	0.0008	-89.1%	0.007	0.030
8	$0.00 \leq \bar{\rho} < 0.33$	0.0205	0.0021	-89.7%	0.011	0.078
	$0.33 \leq \bar{\rho} < 0.66$	0.0102	0.0013	-86.8%	0.011	0.079
	$0.66 \leq \bar{\rho} < 1.00$	0.0091	0.0015	-84.0%	0.011	0.079
16	$0.00 \leq \bar{\rho} < 0.33$	0.0219	0.0026	-88.2%	0.022	0.157
	$0.33 \leq \bar{\rho} < 0.66$	0.0105	0.0018	-82.7%	0.022	0.157
	$0.66 \leq \bar{\rho} < 1.00$	0.0095	0.0022	-76.8%	0.022	0.158
32	$0.00 \leq \bar{\rho} < 0.33$	0.0242	0.0029	-88.2%	0.037	0.322
	$0.33 \leq \bar{\rho} < 0.66$	0.0117	0.0023	-80.1%	0.038	0.325
	$0.66 \leq \bar{\rho} < 1.00$	0.0092	0.0030	-67.8%	0.038	0.325
64	$0.00 \leq \bar{\rho} < 0.33$	0.0264	0.0030	-88.7%	0.074	0.759
	$0.33 \leq \bar{\rho} < 0.66$	0.0128	0.0027	-78.5%	0.075	0.757
	$0.66 \leq \bar{\rho} < 1.00$	0.0096	0.0036	-62.9%	0.075	0.757
128	$0.00 \leq \bar{\rho} < 0.33$	0.0278	0.0031	-89.0%	0.147	2.402
	$0.33 \leq \bar{\rho} < 0.66$	0.0132	0.0031	-76.9%	0.148	2.333
	$0.66 \leq \bar{\rho} < 1.00$	0.0099	0.0041	-59.1%	0.148	2.331
256	$0.00 \leq \bar{\rho} < 0.33$	0.0284	0.0031	-89.1%	0.347	10.601
	$0.33 \leq \bar{\rho} < 0.66$	0.0133	0.0032	-75.8%	0.348	9.945
	$0.66 \leq \bar{\rho} < 1.00$	0.0100	0.0044	-56.1%	0.350	9.938
512	$0.00 \leq \bar{\rho} < 0.33$	0.0286	0.0031	-89.0%	0.795	64.849
	$0.33 \leq \bar{\rho} < 0.66$	0.0130	0.0033	-74.9%	0.796	59.213
	$0.66 \leq \bar{\rho} < 1.00$	0.0098	0.0046	-53.7%	0.808	59.214

(a) Small variations of random correlation coefficients $\rho(X_i, X_j)$

n	$\bar{\rho}$	$\lvert\epsilon_n\rvert$	$\lvert\epsilon_{sn}\rvert$	$\frac{\lvert\epsilon_{sn}\rvert-\lvert\epsilon_n\rvert}{\lvert\epsilon_n\rvert}$	T_n[ms]	T_{sn}[ms]
4	$0.00 \leq \bar{\rho} < 0.33$	0.0459	0.0195	-57.4%	0.007	0.034
	$0.33 \leq \bar{\rho} < 0.66$	0.0265	0.0110	-58.3%	0.007	0.034
	$0.66 \leq \bar{\rho} < 1.00$	0.0168	0.0081	-51.8%	0.007	0.034
8	$0.00 \leq \bar{\rho} < 0.33$	0.0605	0.0337	-44.3%	0.011	0.075
	$0.33 \leq \bar{\rho} < 0.66$	0.0339	0.0208	-38.8%	0.011	0.075
	$0.66 \leq \bar{\rho} < 1.00$	0.0208	0.0129	-37.8%	0.012	0.075
16	$0.00 \leq \bar{\rho} < 0.33$	0.0644	0.0385	-40.2%	0.019	0.148
	$0.33 \leq \bar{\rho} < 0.66$	0.0330	0.0229	-30.6%	0.019	0.149
	$0.66 \leq \bar{\rho} < 1.00$	0.0216	0.0148	-31.5%	0.019	0.150
32	$0.00 \leq \bar{\rho} < 0.33$	0.0580	0.0314	-45.9%	0.033	0.315
	$0.33 \leq \bar{\rho} < 0.66$	0.0285	0.0200	-29.7%	0.033	0.315
	$0.66 \leq \bar{\rho} < 1.00$	0.0207	0.0138	-33.1%	0.034	0.318
64	$0.00 \leq \bar{\rho} < 0.33$	0.0487	0.0234	-51.9%	0.064	0.739
	$0.33 \leq \bar{\rho} < 0.66$	0.0227	0.0163	-28.1%	0.064	0.739
	$0.66 \leq \bar{\rho} < 1.00$	0.0193	0.0125	-35.5%	0.065	0.743
128	$0.00 \leq \bar{\rho} < 0.33$	0.0399	0.0175	-56.2%	0.151	2.317
	$0.33 \leq \bar{\rho} < 0.66$	0.0173	0.0133	-23.3%	0.151	2.319
	$0.66 \leq \bar{\rho} < 1.00$	0.0183	0.0111	-39.6%	0.153	2.327
256	$0.00 \leq \bar{\rho} < 0.33$	0.0316	0.0136	-57.1%	0.351	9.912
	$0.33 \leq \bar{\rho} < 0.66$	0.0133	0.0110	-16.9%	0.352	9.912
	$0.66 \leq \bar{\rho} < 1.00$	0.0178	0.0106	-40.2%	0.355	9.929
512	$0.00 \leq \bar{\rho} < 0.33$	0.0302	0.0117	-61.4%	0.810	59.094
	$0.33 \leq \bar{\rho} < 0.66$	0.0134	0.0094	-29.9%	0.809	59.101
	$0.66 \leq \bar{\rho} < 1.00$	0.0175	0.0110	-37.0%	0.817	59.129

(b) Large variations of random correlation coefficients $\rho(X_i, X_j)$

▼ **Table B.6 —** Runtime T and error ϵ for the approximation of the distribution of the maximum delay of n critical target paths, sensitized by 1, 5, 10 and 20 test vector-pairs.

circuit	n	$\lvert\epsilon_n\rvert$	$\lvert\epsilon_{sn}\rvert$	$\frac{\lvert\epsilon_{sn}\rvert-\lvert\epsilon_n\rvert}{\lvert\epsilon_n\rvert}$	T_n[ms]	T_{sn}[ms]
p35k	4 - 15	0.0124	0.0065	-48.0%	0.013	0.086
	16 - 31	0.0259	0.0152	-41.2%	0.025	0.221
	32 - 63	0.0415	0.0256	-38.3%	0.049	0.525
	64 - 127	0.0669	0.0427	-36.1%	0.100	1.358
	128 - 255	0.0932	0.0681	-26.9%	0.182	3.770
	256 - 511	0.1102	0.0883	-19.8%	0.390	12.007
p45k	4 - 15	0.0207	0.0126	-39.1%	0.011	0.073
	16 - 31	0.0412	0.0283	-31.1%	0.025	0.216
	32 - 63	0.0618	0.0465	-24.8%	0.048	0.508
	64 - 127	0.0726	0.0551	-24.1%	0.091	1.183
	128 - 255	0.0815	0.0609	-25.2%	0.197	4.632
	256 - 511	0.0996	0.0805	-19.1%	0.491	23.597
	512 - 1023	0.0727	0.0787	8.3%	1.339	120.152
p77k	4 - 15	0.0146	0.0100	-31.8%	0.011	0.068
	16 - 31	0.0249	0.0171	-31.4%	0.025	0.211
	32 - 63	0.0355	0.0244	-31.1%	0.050	0.527
	64 - 127	0.0464	0.0335	-27.7%	0.101	1.373
	128 - 255	0.0625	0.0457	-26.9%	0.200	4.710
	256 - 511	0.0896	0.0723	-19.3%	0.432	21.162
	512 - 1023	0.1202	0.1066	-11.3%	1.171	103.090
p78k	4 - 15	0.0127	0.0072	-43.4%	0.010	0.063
	16 - 31	0.0361	0.0240	-33.6%	0.025	0.226
	32 - 63	0.0563	0.0424	-24.6%	0.047	0.511
	64 - 127	0.0758	0.0614	-19.1%	0.082	1.028
p81k	4 - 15	0.0159	0.0114	-28.4%	0.012	0.084
	16 - 31	0.0352	0.0264	-25.0%	0.024	0.217
	32 - 63	0.0661	0.0489	-26.1%	0.046	0.491
	64 - 127	0.0624	0.0485	-22.3%	0.093	1.230
	128 - 255	0.0639	0.0533	-16.6%	0.201	4.716
	256 - 511	0.0954	0.0834	-12.6%	0.392	17.669
p100k	4 - 15	0.0130	0.0082	-37.0%	0.011	0.070
	16 - 31	0.0290	0.0208	-28.3%	0.025	0.227
	32 - 63	0.0404	0.0288	-28.6%	0.047	0.508
	64 - 127	0.0588	0.0450	-23.5%	0.092	1.220
	128 - 255	0.0789	0.0578	-26.8%	0.179	3.799
	256 - 511	0.0800	0.0591	-26.1%	0.389	17.480
p267k	4 - 15	0.0180	0.0103	-42.6%	0.011	0.073
	16 - 31	0.0380	0.0254	-33.1%	0.024	0.211
	32 - 63	0.0545	0.0411	-24.6%	0.047	0.493
	64 - 127	0.0719	0.0555	-22.7%	0.098	1.317
	128 - 255	0.0935	0.0755	-19.2%	0.194	4.408
	256 - 511	0.1036	0.0867	-16.3%	0.461	24.767
	512 - 1023	0.1558	0.1307	-16.1%	1.572	157.739
	1024 - 2048	0.2380	0.2237	-6.0%	5.496	1296.518
p330k	4 - 15	0.0136	0.0079	-42.3%	0.012	0.077
	16 - 31	0.0319	0.0214	-32.7%	0.026	0.224
	32 - 63	0.0518	0.0369	-28.8%	0.048	0.507
	64 - 127	0.0774	0.0562	-27.4%	0.097	1.263
	128 - 255	0.0838	0.0609	-27.3%	0.207	4.890
	256 - 511	0.1157	0.0926	-20.0%	0.464	24.221
	512 - 1023	0.0681	0.0605	-11.1%	1.600	156.022
	1024 - 2048	0.0959	0.0929	-3.1%	6.517	1195.454

▼ **Table B.7 —** Runtime T and error ϵ for the approximation of the distribution of the maximum delay of n critical target paths, sensitized by 1, 5, 10 and 20 test vector-pairs, using covariance matrix scaling proposed in section 7.5.3

circuit	n	$\lvert\epsilon_n\rvert$	$\lvert\epsilon_{sn}\rvert$	$\frac{\lvert\epsilon_{sn}\rvert-\lvert\epsilon_n\rvert}{\lvert\epsilon_n\rvert}$	T_n[ms]	T_{sn}[ms]
p35k	4 - 15	0.0124	0.0057	-54.3%	0.012	0.084
	16 - 31	0.0259	0.0117	-54.9%	0.024	0.214
	32 - 63	0.0415	0.0164	-60.5%	0.048	0.513
	64 - 127	0.0669	0.0232	-65.4%	0.098	1.346
	128 - 255	0.0932	0.0337	-63.8%	0.179	3.811
	256 - 511	0.1102	0.0415	-62.3%	0.383	12.300
p45k	4 - 15	0.0207	0.0112	-45.9%	0.011	0.072
	16 - 31	0.0412	0.0202	-50.8%	0.024	0.211
	32 - 63	0.0618	0.0280	-54.8%	0.047	0.500
	64 - 127	0.0726	0.0311	-57.2%	0.090	1.174
	128 - 255	0.0815	0.0288	-64.7%	0.196	4.701
	256 - 511	0.0996	0.0368	-63.1%	0.488	23.904
	512 - 1023	0.0727	0.0242	-66.7%	1.289	120.964
p77k	4 - 15	0.0146	0.0090	-38.3%	0.011	0.068
	16 - 31	0.0249	0.0115	-54.1%	0.024	0.206
	32 - 63	0.0355	0.0159	-55.3%	0.049	0.518
	64 - 127	0.0464	0.0185	-60.1%	0.100	1.366
	128 - 255	0.0625	0.0230	-63.1%	0.198	4.815
	256 - 511	0.0896	0.0270	-69.9%	0.426	21.788
	512 - 1023	0.1202	0.0577	-52.0%	1.136	103.990
p78k	4 - 15	0.0127	0.0061	-51.5%	0.010	0.063
	16 - 31	0.0361	0.0155	-57.1%	0.024	0.220
	32 - 63	0.0563	0.0227	-59.8%	0.047	0.503
	64 - 127	0.0758	0.0296	-60.9%	0.081	1.022
p81k	4 - 15	0.0159	0.0099	-37.7%	0.012	0.084
	16 - 31	0.0352	0.0184	-47.6%	0.024	0.213
	32 - 63	0.0661	0.0287	-56.6%	0.046	0.483
	64 - 127	0.0624	0.0263	-57.9%	0.092	1.222
	128 - 255	0.0639	0.0302	-52.8%	0.200	4.737
	256 - 511	0.0954	0.0484	-49.2%	0.390	17.841
p100k	4 - 15	0.0130	0.0073	-44.2%	0.011	0.071
	16 - 31	0.0290	0.0139	-52.3%	0.025	0.223
	32 - 63	0.0404	0.0157	-61.1%	0.047	0.502
	64 - 127	0.0588	0.0212	-63.9%	0.092	1.213
	128 - 255	0.0789	0.0279	-64.7%	0.178	3.868
	256 - 511	0.0800	0.0460	-42.6%	0.410	17.934
p267k	4 - 15	0.0180	0.0091	-49.6%	0.011	0.072
	16 - 31	0.0380	0.0180	-52.7%	0.024	0.205
	32 - 63	0.0545	0.0259	-52.5%	0.046	0.483
	64 - 127	0.0719	0.0321	-55.3%	0.096	1.304
	128 - 255	0.0935	0.0436	-53.3%	0.191	4.461
	256 - 511	0.1036	0.0439	-57.6%	0.455	25.141
	512 - 1023	0.1558	0.0699	-55.1%	1.554	160.024
	1024 - 2048	0.2380	0.1383	-41.9%	5.262	1287.566
p330k	4 - 15	0.0136	0.0070	-48.4%	0.011	0.073
	16 - 31	0.0319	0.0146	-54.3%	0.025	0.216
	32 - 63	0.0518	0.0201	-61.2%	0.047	0.496
	64 - 127	0.0774	0.0273	-64.8%	0.094	1.252
	128 - 255	0.0838	0.0312	-62.8%	0.203	4.919
	256 - 511	0.1157	0.0513	-55.7%	0.449	24.558
	512 - 1023	0.0681	0.0378	-44.5%	1.536	156.395
	1024 - 2048	0.0959	0.0470	-50.9%	5.014	1148.219

Bibliography

[Aftab09] S.-A. Aftabjahani and L. Milor. Fast Variation-Aware Statistical Dynamic Timing Analysis. In *World Congress on Computer Science and Information Engineering*, pages 488–492. Los Angeles, CA, USA, mar 2009. [page 46]

[Agarw07] K. Agarwal and S. Nassif. Characterizing process variation in nanometer CMOS. In *Proc. Design Automation Conf. (DAC)*, pages 396–399. San Diego, CA, USA, jun 2007. [pages 24 and 104]

[Ahmed06] N. Ahmed, M. Tehranipoor, and V. Jayaram. Timing-based delay test for screening small delay defects. In *Proc. Design Automation Conf. (DAC)*, pages 320–325. San Francisco, CA, USA, jul 2006. [page 10]

[Arism13] J. Arismendi. Multivariate truncated moments. *Journal of Multivariate Analysis*, 117:41–75, 2013. [page 134]

[Aseno08] A. Asenov, A. Cathignol, B. Cheng, K. P. McKenna, A. R. Brown, A. L. Shluger, D. Chanemougame, K. Rochereau, and G. Ghibaudo. Origin of the asymmetry in the magnitude of the statistical variability of n- and p-channel poly-Si gate bulk MOSFETs. *IEEE Electron Device Letters*, 29(8):913–915, 2008. [page 21]

[Ash00] R. B. Ash and C. Doleans-Dade. *Probability and Measure Theory*. Academic Press, second edition, 2000. ISBN 978-0-12-065202-0. [page 36]

[Azzal99] A. Azzalini and A. Capitanio. Statistical applications of the multivariate skew normal distribution. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 61(3):579–602, 1999. [page 133]

[Azzal13] A. Azzalini and A. Capitanio. *The Skew-Normal and Related Families*. Cambridge University Press, 2013. ISBN 978-1-139-24889-1. [pages 49, 76, 77, 80, 81, and 82]

[Balas91] B. K. Balasubramanian, M. I. Beg, and R. B. Bapat. On Families of Distributions Closed under Extrema. *Sankhyā: The Indian Journal of Statistics, Series A (1961-2002)*, 53(3):375–388, 1991. [page 15]

[Becke10] B. Becker, S. Hellebrand, I. Polian, B. Straube, W. Vermeiren, and H.-J. Wunderlich. Massive statistical process variations: A grand challenge for testing nanoelectronic circuits. In *Int. Conf. on Dependable Systems and Networks Workshops (DSN-W)*, pages 95–100. Chicago, IL, USA, jun 2010. [pages 6 and 11]

[Bench09] C. Bencher, H. Dai, and Y. Chen. Gridded design rule scaling: taking the CPU toward the 16nm node. In *Proc. of SPIE Advanced Lithography*. San Jose, CA, USA, feb 2009. [page 2]

[Birnb51] Z. W. Birnbaum and P. L. Meyer. On the effect of truncation in some or all coordinates of a multinormal population. Technical report, Washington University and Seattle Laboratory of Statistical Research, nov 1951. [page 73]

[Blaau08] D. Blaauw, K. Chopra, A. Srivastava, and L. Scheffer. Statistical Timing Analysis: From Basic Principles to State of the Art. *IEEE Trans. Computer-Aided Design*, 27(4):589–607, 2008. [pages vii, 20, 23, and 24]

[Bose07] S. Bose and V. D. Agrawal. Delay Test Quality Evaluation Using Bounded Gate Delays. In *Proc. VLSI Test Symp. (VTS)*, pages 23–28. Berkeley, CA, USA, may 2007. [page 44]

[Bushn00] M. L. Bushnell and V. D. Agrawal. *Essentials of electronic testing for digital, memory and mixed-signal VLSI circuits*. Kluwer Academic Publishers, 2000. ISBN 978-0-306-47040-0. [page 4]

[Chakr00] S. Chakrabarti, S. Das, D. Das, and B. Bhattacharya. Synthesis of symmetric functions for path-delay fault testability. *Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, 19(9):1076–1081, 2000. [page 111]

[Chakr99] S. Chakraborty, D. Dill, and K. Yun. Min-max timing analysis and an application to asynchronous circuits. *Proceedings of the IEEE*, 87(2):332–346, 1999. [page 44]

[Chakr12] S. Chakravarty, N. Devta-Prasanna, A. Gunda, J. Ma, F. Yang, H. Guo, R. Lai, and D. Li. Silicon evaluation of faster than at-speed transition delay tests. In *Proc. VLSI Test Symp. (VTS)*, pages 80–85. Hyatt Maui, HI, USA, apr 2012. [page 11]

[Chang96] J.-Y. Chang and E. McCluskey. Detecting delay flaws by very-low-voltage testing. In *Proc. Int. Test Conf. (ITC)*, pages 367–376. Washington, D.C., USA, oct 1996. [page 11]

[Chang98] T.-Y. J. Chang. *Voltage Screens Early-Life Failures In CMOS Integrated Circuits*. Ph.D. thesis, Stanford University, 1998. [page 11]

[Cheng93] K. Cheng and H. Chen. Delay testing for non-robust untestable circuits. In *Proc. Int. Test Conf. (ITC)*, pages 954–961. Baltimore, MD, USA, oct 1993. [page 8]

[Cheng94] K.-T. Cheng and H.-C. Chen. Generation of high quality non-robust tests for path delay faults. In *Proc. Design Automation Conf. (DAC)*, pages 365–369. San Diego, CA, USA, jun 1994. [page 9]

[Cheng08] L. Cheng. Non-Gaussian statistical timing analysis using second-order polynomial fitting. In *Proc. of Asia and South Pacific Design Automation Conf.*, pages 298–303. Seoul, South Korea, jan 2008. [page 50]

[Choi09] Y. Choi. New form of block matrix inversion. In *IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, pages 1952–1957. Singapore, jul 2009. [page 144]

[Chopr06] K. Chopra, B. Zhai, D. Blaauw, and D. Sylvester. A New Statistical Max Operation for Propagating Skewness in Statistical Timing Analysis. In *Proc. Int. Conf. Computer-Aided Design (ICCAD)*, pages 237–243. San Jose, CA, USA, nov 2006. [page 49]

[Clark61] C. E. Clark. The greatest of a finite set of random variables. *Operations Research*, 9(2):145–162, 1961. [pages 48, 72, 100, 129, and 130]

[Czado11] C. Czado and T. Schmidt. *Mathematische Statistik*. Springer Berlin Heidelberg, 2011. ISBN 978-3-642-17261-8. [page 37]

[Czutr12] A. Czutro, M. E. Imhof, J. Jiang, A. Mumtaz, M. Sauer, B. Becker, I. Polian, and H.-J. Wunderlich. Variation-Aware Fault Grading. In *Proc. IEEE Asian Test Symp. (ATS)*, pages 344–349. Niigata, Japan, nov 2012. [pages 11 and 46]

[Daasc07] W. R. Daasch, M. Ward, and J. Van Slyke. Silicon evaluation of longest path avoidance testing for small delay defects. In *Proc. Int. Test Conf. (ITC)*. Santa Clara, CA, USA, oct 2007. [page 10]

[DeGro12] M. H. DeGroot and M. J. Schervish. *Probability and Statistics*. Pearson Education, fourth edition, 2012. ISBN 978-0-321-50046-5. [pages 26, 27, 28, 29, 32, 33, 34, and 36]

[Dervi91] B. I. Dervisoglu and G. E. Stong. Design for Testability Using Scanpath Techniques for Path-Delay Test and Measurement. In *Proc. Int. Test Conf. (ITC)*, pages 365–374. Nashville, TN, USA, oct 1991. [page 3]

[Devad92] S. Devadas and K. Keutzer. Validatable nonrobust delay-fault testable circuits via logic synthesis. *IEEE Trans. Computer-Aided Design*, 11(12):1559–1573, 1992. [page 8]

[Devga03] A. Devgan and C. Kashyap. Block-based static timing analysis with uncertainty. In *Proc. Int. Conf. Computer-Aided Design (ICCAD)*, pages 607–614. San Jose, CA, USA, nov 2003. [page 49]

[Dietr12] M. Dietrich and J. Haase. *Process Variations and Probabilistic Integrated Circuit Design*. Springer New York, 2012. ISBN 978-1-4419-6620-9. [page 21]

[Egger11] S. Eggersglüß and R. Drechsler. As-Robust-As-Possible test generation in the presence of small delay defects using pseudo-Boolean optimization. In *Proc. Design, Automation and Test in Europe (DATE)*. Grenoble, France, mar 2011. [page 9]

[Eiche77] E. B. Eichelberger and T. W. Williams. A logic design structure for LSI testability. In *Proc. Design Automation Conf. (DAC)*, pages 462–468. New Orleans, LA, USA, jun 1977. [page 3]

[Erb14] D. Erb, K. Scheibler, M. Sauer, S. M. Reddy, and B. Becker. Circuit parameter independent test pattern generation for interconnect open defects. In *Proc. IEEE Asian Test Symp. (ATS)*, pages 131–136. Hangzhou, China, nov 2014. [page 10]

[Fonse10] R. A. Fonseca, L. Dilillo, A. Bosio, P. Girard, S. Pravossoudovitch, A. Virazel, and N. Badereddine. Analysis of resistive-bridging defects in SRAM core-cells: A comparative study from 90nm down to 40nm technology nodes. In *IEEE European Test Symp. (ETS)*, pages 132–137. Prague, Czech Republic, may 2010. [page 11]

[Forza09] C. Forzan and D. Pandini. Statistical static timing analysis: A survey. *Integration, the VLSI Journal*, 42(3):409–435, 2009. [pages 20 and 23]

[Franc10] C. Franceschini and N. M. R. Loperfido. A skewed GARCH-type model for multivariate financial time series. In *Mathematical and Statistical Methods for Actuarial Sciences and Finance*, pages 143–152. Springer-Verlag Italia, 2010. ISBN 978-88-470-1481-7. [pages 35, 81, and 87]

[Frans04] S. Franssila. *Introduction to microfabrication*. John Wiley & Sons Ltd, 2004. ISBN 978-0-470-85106-7. [page 1]

[Fuchs91] K. Fuchs, F. Fink, and M. Schulz. DYNAMITE: an efficient automatic test pattern generation system for path delay faults. *IEEE Trans. Computer-Aided Design*, 10(10):1323–1335, 1991. [pages 110 and 112]

[Fuchs94] K. Fuchs, M. Pabst, and T. Roessel. RESIST: a recursive test pattern generation algorithm for path delay faults considering various test classes. *Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, 13(12):1550–1562, 1994. [page 110]

[Gajsk83] D. D. Gajski and R. H. Kuhn. Guest Editors' Introduction: New VLSI Tools. *Computer*, 16(12):11–14, 1983. [page 38]

[Genz92] A. Genz. Numerical Computation of Multivariate Normal Probabilities. *Journal of Computational and Graphical Statistics*, 1(2):141–149, 1992. [pages xi, 66, 71, 116, and 119]

[Genz04] A. Genz. Numerical computation of rectangular bivariate and trivariate normal and t probabilities. *Statistics and Computing*, 14(3):251–260, 2004. [pages xi, 66, 71, 74, 116, 119, and 135]

[Goel13] S. K. Goel and K. Chakrabarty. *Testing for Small-Delay Defects in Nanoscale CMOS Integrated Circuits*. CRC Press, 2013. ISBN 978-1-4398-2942-4. [pages vii, 2, and 10]

[Golub13] G. H. Golub and C. F. V. Loan. *Matrix Computations*. Johns Hopkins University Press, fourth edition, 2013. ISBN 978-1-4214-0794-4. [pages 17, 19, 20, and 66]

[Green12] W. H. Greene. *Econometric analysis*. Pearson Education Limited, seventh edition, 2012. ISBN 978-0-273-75356-8. [page 30]

[Gut09] A. Gut. *An Intermediate Course in Probability*. Springer New York, second edition, 2009. ISBN 978-1-4419-0162-0. [page 38]

[Hanso05] S. Hanson, D. Blaauw, and D. Sylvester. Analysis and mitigation of variability in subthreshold design. In *Proc. Int. Symp. on Low Power Electronics and Design (ISLPED)*, pages 20–25. San Diego, CA, USA, aug 2005. [page 47]

[Hao93] H. Hao and E. McCluskey. Very-low-voltage testing for weak CMOS logic ICs. In *Proc. Int. Test Conf. (ITC)*, pages 275–284. Baltimore, MD, USA, oct 1993. [page 11]

[Hapke11] F. Hapke, J. Schloeffel, W. Redemund, A. Glowatz, J. Rajski, M. Reese, J. Rearick, and J. Rivers. Cell-aware analysis for small-delay effects and production test results from different fault models. In *Proc. Int. Test Conf. (ITC)*. Anaheim, CA, USA, sep 2011. [page 131]

[Hopsc10] F. Hopsch, B. Becker, S. Hellebrand, I. Polian, B. Straube, W. Vermeiren, and H.-J. Wunderlich. Variation-Aware Fault Modeling. In *Proc. IEEE Asian Test Symp. (ATS)*, pages 87–93. Shanghai, China, dec 2010. [pages 5 and 47]

[Ingel09] U. Ingelsson, B. Al-Hashimi, S. Khursheed, S. Reddy, and P. Harrod. Process Variation-Aware Test for Resistive Bridges. *IEEE Trans. Computer-Aided Design*, 28(8):1269–1274, 2009. [page 11]

[Ingel11] U. Ingelsson and B. M. Al-Hashimi. Investigation into voltage and process variation-aware manufacturing test. In *Proc. Int. Test Conf. (ITC)*. Anaheim, CA, USA, sep 2011. [pages 5 and 45]

[Intel15] Intel Corporation. Intel Math Kernel Library 11.3 - Developer Reference, 2015. [page 103]

[Ishiu89] N. Ishiura, M. Takahashi, and S. Yajima. Time-symbolic simulation for accurate timing verification of asynchronous behavior of logic circuits. In *Proc. Design Automation Conf. (DAC)*, pages 497–502. Las Vegas, NV, USA, jun 1989. [page 43]

[Ishiu90] N. Ishiura, Y. Deguchi, and S. Yajima. Coded Time-Symbolic Simulation Using Shared Binary Decision Diagrams. In *Proc. Design Automation Conf. (DAC)*, pages 130–135. Orlando, FL, USA, jun 1990. [page 44]

[ITRS12] ITRS. International Technology Roadmap for Semiconductors. http://www.itrs.net/ITRS2012. [pages x, 1, and 2]

[Iyeng88a] V. Iyengar, B. Rosen, and I. Spillinger. Delay test generation. I. Concepts and coverage metrics. In *Proc. Int. Test Conf. (ITC)*, pages 857–866. Washington, D.C. USA, sep 1988. [page 5]

[Iyeng88b] V. Iyengar, B. Rosen, and I. Spillinger. Delay test generation. II. Algebra and algorithms. In *Proc. Int. Test Conf. (ITC)*, pages 867–876. Washington, D.C. USA, sep 1988. [page 5]

[Jayar13] D. Jayaraman and S. Tragoudas. A method to determine the sensitization probability of a non-robustly testable path. In *Proc. Int. Symp. on Quality Electronic Design (ISQED)*, pages 676–681. Santa Clara, CA, USA, mar 2013. [page 44]

[Jha92] N. Jha, I. Pomeranz, S. Reddy, and R. Miller. Synthesis of multi-level combinational circuits for complete robust path delay fault testability. In *Int. Symp. on Fault-Tolerant Computing (FTCS)*, pages 280–287. Boston, MA, USA, jul 1992. [page 111]

[Jung12] J. Jung, T. Kim, and S. Member. Variation-Aware False Path Analysis Based on Statistical Dynamic Timing Analysis. *Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, 31(11):1684–1697, 2012. [page 44]

[Kang13] C. Y. Kang, C. Sohn, R.-H. Baek, C. Hobbs, P. Kirsch, and R. Jammy. Effects of Layout and Process Parameters on Device / Circuit Performance and Variability for 10nm Node FinFET Technology. In *IEEE Symposium on VLSI Technology*, pages T90–T91. Kyoto, Japan, jun 2013. [page 2]

[Kenne01] M. C. Kennedy and A. O'Hagan. Bayesian calibration of computer models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(3):425–464, 2001. [page 20]

[Kiure09] A. D. Kiureghian and O. Ditlevsen. Aleatory or epistemic? Does it matter? *Structural Safety*, 31(2):105–112, 2009. [page 23]

[Kollo05] T. Kollo and D. von Rosen. *Advanced Multivariate Statistics with Matrices*. Springer Netherlands, 2005. ISBN 978-1-4020-3418-3. [page 88]

[Konuk00] H. Konuk. On invalidation mechanisms for non-robust delay tests. In *Proc. Int. Test Conf. (ITC)*, pages 393–399. Atlantic City, NJ, USA, oct 2000. [pages vii, 8, and 9]

[Kopp08] J. Kopp. Efficient numerical diagonalization of hermitian 3x3 matrices. *International Journal of Modern Physics C*, 19(3), 2008. [page 18]

[Krsti95] A. Krstic and K.-T. Cheng. Generation of high quality tests for functional sensitizable paths. In *Proc. VLSI Test Symp. (VTS)*, pages 374–379. Princeton, NJ, USA, may 1995. [pages 110 and 111]

[Kruse04] B. Kruseman, A. K. Majhi, G. Gronthoud, and S. Eichenberger. On hazard-free patterns for fine-delay fault testing. In *Proc. Int. Test Conf. (ITC)*, pages 213–222. Charlotte, NC, USA, oct 2004. [page 11]

[Kuhn08] K. Kuhn, C. Kenyon, A. Kornfeld, M. Liu, A. Maheshwari, W.-k. Shih, S. Sivakumar, G. Taylor, P. VanDerVoorn, and K. Zawadzki. Managing Process Variation in Intel's 45nm CMOS Technology. *Intel Technology Journal*, 12:93–110, 2008. [pages 21 and 104]

[Kuhn09] K. J. Kuhn. CMOS Scaling Beyond 32nm: Challenges and Opportunities. In *Proc. Design Automation Conf. (DAC)*, pages 310–313. San Francisco, CA, USA, jul 2009. [page 21]

[Kuhn10] K. J. Kuhn. CMOS transistor scaling past 32nm and implications on variation. In *IEEE/SEMI Advanced Semiconductor Manufacturing Conference (ASMC)*, pages 241–246. San Francisco, CA, USA, jul 2010. [page 2]

[Kuruv13] V. Kuruvilla, D. Sinha, J. Piaget, C. Visweswariah, and N. Chandrachoodan. Speeding up computation of the max/min of a set of gaussians for statistical timing analysis and optimization. In *Proc. Design Automation Conf. (DAC)*. Austin, TX, USA, jun 2013. [page 48]

[Lee05a] B. Lee, H. Li, L.-C. Wang, and M. S. Abadir. Hazard-aware statistical timing simulation and its applications in screening frequency-dependent defects. In *Proc. Int. Test Conf. (ITC)*, pages 91–100. Austin, TX, USA, nov 2005. [page 11]

[Lee05b] B. Lee, L.-C. Wang, and M. S. Abadir. Reducing pattern delay variations for screening frequency dependent defects. In *Proc. VLSI Test Symp. (VTS)*, pages 153–160. Palm Springs, CA, USA, may 2005. [pages 13 and 47]

[Leung12a] G. Leung and C. O. Chui. Variability impact of random dopant fluctuation on nanoscale junctionless FinFETs. *IEEE Electron Device Letters*, 33(6):767–769, 2012. [pages vii, x, 21, 22, and 23]

[Leung12b] G. Leung, L. Lai, P. Gupta, and C. O. Chui. Device- and circuit-level variability caused by line edge roughness for sub-32-nm FinFET technologies. *IEEE Transactions on Electron Devices*, 59(8):2057–2063, 2012. [pages x, 2, 22, and 23]

[Li10] Y. Li, C. H. Hwang, T. Y. Li, and M. H. Han. Process-variation effect, metal-gate work-function fluctuation, and random-dopant fluctuation in emerging CMOS technologies. *IEEE Transactions on Electron Devices*, 57(2):437–447, 2010. [pages 21, 24, and 104]

[Lin87] C. Lin and S. Reddy. On Delay Fault Testing in Logic Circuits. *IEEE Trans. Computer-Aided Design*, 6(5):694–703, 1987. [page 110]

[Liou01] J.-J. Liou, K.-T. Cheng, S. Kundu, and A. Krstic. Fast statistical timing analysis by probabilistic event propagation. In *Proc. Design Automation Conf. (DAC)*, pages 661–666. Las Vegas, NV, USA, jun 2001. [page 49]

[Loper13] N. M. R. Loperfido. Skewness and the linear discriminant function. *Statistics & Probability Letters*, 83(1):93–99, 2013. [page 89]

[Louhi02] S. Louhichi. Rates of Convergence in the CLT for Some Weakly Dependent Random Variables. *Theory of Probability and its Applications*, 46(2):297–315, 2002. [page 63]

[Luca15] G. D. Luca and N. Loperfido. Modelling multivariate skewness in financial returns: a SGARCH approach. *The European Journal of Finance*, 21(13-14):1113–1131, 2015. [pages 35 and 87]

[Ma11] Y. Ma, J. Sweis, C. Bencher, Y. Deng, H. Dai, H. Yoshida, B. Gisuthan, J. Kye, and H. J. Levinson. Double patterning compliant logic design. In *Proc. of SPIE: Design for Manufacturability through Design-Process Integration V*. San Jose, CA, USA, feb 2011. [page 2]

[Matti09] R. Mattiuzzo, D. Appello, and C. Allsup. Small delay defect testing. *Test and Measurement World*, pages 37–41, jun 2009. [page 10]

[Mille12] S. L. Millers and D. G. Childers. *Probability and Random Processes with Application to Signal Processing and Communications*. Elsevier Academic Press, second edition, 2012. ISBN 978-0-12-386981-4. [page 35]

[Montg13] D. C. Montgomery and G. C. Runger. *Applied Statistics and Probability for Engineers*. John Wiley & Sons Inc, sixth edition, 2013. ISBN 978-1-118-53971-2. [pages 25 and 30]

[Mori94] T. F. Móri, V. K. Rohatgi, and G. J. Székely. On Multivariate Skewness and Kurtosis. *Theory of Probability & Its Applications*, 38(3):547–551, 1994. [page 87]

[Nadar08] S. Nadarajah and S. Kotz. Exact Distribution of the Max/Min of Two Gaussian Random Variables. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 16(2):210–212, 2008. [pages 15 and 68]

[nan11] Nangate 45nm Open Cell Library, aug 2011. [page 103]

[Needh98] W. Needham, C. Prunty, and E. H. Yeoh. High volume microprocessor test escapes, an analysis of defects our tests are missing. In *Proc. Int. Test Conf. (ITC)*, pages 25–34. Washington, D.C., USA, oct 1998. [page 11]

[Nigh00] P. Nigh and A. Gattiker. Test method evaluation experiments and data. In *Proc. Int. Test Conf. (ITC)*, pages 454–463. Atlantic City, NJ, USA, oct 2000. [page 47]

[Peng10] K. Peng, M. Yilmaz, M. Tehranipoor, and K. Chakrabarty. High-quality pattern selection for screening small-delay defects considering process variations and crosstalk. In *Proc. Design, Automation and Test in Europe (DATE)*, pages 1426–1431. Dresden, Germany, mar 2010. [page 45]

[Peng13] K. Peng, M. Yilmaz, K. Chakrabarty, and M. Tehranipoor. Crosstalk- and process variations-aware high-quality tests for small-delay defects. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 21(6):1129–1142, 2013. [page 45]

[Polia11] I. Polian, B. Becker, S. Hellebrand, H.-J. Wunderlich, and P. Maxwell. Towards Variation-Aware Test Methods. In *IEEE European Test Symp. (ETS)*, pages 219–225. Trondheim, Norway, may 2011. [page 11]

[Qian10] X. Qian and A. D. Singh. Distinguishing Resistive Small Delay Defects from Random Parameter Variations. In *Proc. IEEE Asian Test Symp. (ATS)*, pages 325–330. Shanghai, China, dec 2010. [page 13]

[Rabae03] J. M. Rabaey, A. Chandrakasan, and B. Nikolic. *Digital integrated circuits: a design perspective*. Prentice-Hall International Inc, second edition, 2003. ISBN 978-0-13-090996-1. [pages vii and 40]

[Radha11] G. S. Radhakrishnan and S. Ozev. Adaptive Modeling of Analog/RF Circuits for Efficient Fault Response Evaluation. *Journal of Electronic Testing (JETTA)*, 27(4):465–476, 2011. [page 13]

[Resni14] S. I. Resnick. *A Probability Path*. Birkhäuser Boston, 2014. ISBN 978-0-8176-8408-2. [page 30]

[Ryan14] P. G. Ryan, I. Aziz, W. B. Howell, T. K. Janczak, and D. J. Lu. Process Defect Trends and Strategic Test Gaps. In *Proc. Int. Test Conf. (ITC)*. Seattle, WA, USA, oct 2014. [page 10]

[Sauer12] M. Sauer, A. Czutro, I. Polian, and B. Becker. Small-Delay-Fault ATPG with Waveform Accuracy. In *Proc. Int. Conf. Computer-Aided Design (ICCAD)*, pages 30–36. San Jose, CA, nov 2012. [page 110]

[Sauer14] M. Sauer, I. Polian, M. E. Imhof, A. Mumtaz, E. Schneider, A. Czutro, H.-J. Wunderlich, and B. Becker. Variation-aware deterministic ATPG. In *IEEE European Test Symp. (ETS)*. Paderborn, Germany, may 2014. [pages 11 and 46]

[Savir93] J. Savir and S. Patil. Scan-based transition test. *Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, 12(8):1232–1241, 1993. [page 3]

[Savir94] J. Savir and S. Patil. On broad-side delay test. *Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, 2(3):368–372, 1994. [page 3]

[Schne15] E. Schneider, S. Holst, M. A. Kochte, X. Wen, and H.-J. Wunderlich. GPU-Accelerated Small Delay Fault Simulation. In *Proc. Design, Automation and Test in Europe (DATE)*, pages 1174–1179. Grenoble, France, mar 2015. [page 45]

[Schue13] K. Schuegraf, M. C. Abraham, A. Brand, M. Naik, and R. Thakur. Semiconductor Logic Technology Innovation to Achieve Sub-10 nm Manufacturing. *IEEE Journal of the Electron Devices Society*, 1(3):66–75, 2013. [page 2]

[Segur02] J. Segura, A. Keshavarzi, J. Soden, and C. Hawkins. Parametric failures in CMOS ICs-a defect-based analysis. In *Proc. Int. Test Conf. (ITC)*, pages 90–99. Baltimore, MD, USA, oct 2002. [page 10]

[Smith85] G. L. Smith. Model for delay faults based upon paths. In *Proc. Int. Test Conf. (ITC)*, pages 342–349. Philadelphia, PA, USA, oct 1985. [page 5]

[Sriva05] A. Srivastava, D. Sylvester, and D. Blaauw. *Statistical Analysis and Optimization for VLSI : Timing and Power*. Springer, 2005. ISBN 978-0-387-25738-9. [pages 20, 21, 23, 24, 27, and 31]

[Sumik12] N. Sumikawa, L.-C. Wang, and M. S. Abadir. An experiment of burn-in time reduction based on parametric test analysis. In *Proc. Int. Test Conf. (ITC)*. Anaheim, CA, USA, nov 2012. [page 11]

[Talli61] G. Tallis. The moment generating function of the truncated multi-normal distribution. *Journal of the Royal Statistical Society. Series B (Methodological)*, 23(1):223–229, 1961. [pages 73 and 134]

[Tang14a] Q. Tang, J. Rodriguez, A. Zjajo, M. Berkelaar, and N. Van Der Meijs. Statistical transistor-level timing analysis using a direct random differential equation solver. *Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, 33(2):210–223, 2014. [page 45]

[Tang14b] Q. Tang, A. Zjajo, M. Berkelaar, and N. van der Meijs. Considering Crosstalk Effects in Statistical Timing Analysis. *Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, 33(2):318–322, 2014. [page 130]

[Tang14c] X. Tang, A. Xu, W. Li, and Z. Yang. Fault Models of CMOS Gates: An Empirical Study Based on Mutation Analysis. In *International Conference on Dependable, Autonomic and Secure Computing*, pages 115–120. Dalian, China, aug 2014. [page 131]

[Tseng00] C.-W. Tseng, E. J. McCluskey, X. Shao, J. Wu, and D. M. Wu. Cold delay defect screening. In *Proc. VLSI Test Symp. (VTS)*, pages 183–188. Montreal, QC, Canada, apr 2000. [page 11]

[Tsuki11] S. Tsukiyama and M. Fukui. A new statistical maximum operation for Gaussian mixture models and its evaluations. In *European Conference on Circuit Theory and Design (ECCTD)*, pages 45–48. Linkoping, Sweden, aug 2011. [page 50]

[Ul Ha11] F. Ul-Hassan, W. Vanderbauwhede, and F. Rodriguez-Salazar. Timing yield analysis of pipelined circuits under device variability. In *Proc. of International Symposium on Signals, Circuits and Systems (ISSCS)*. Iasi, Romania, jun 2011. [page 47]

[Vijay14] M. Vijaykumar and V. Vasudevan. Statistical static timing analysis using a skew-normal canonical delay model. In *Proc. Design, Automation and Test in Europe (DATE)*. Dresden, Germany, mar 2014. [page 50]

[Viswe04] C. Visweswariah, K. Ravindran, K. Kalafala, S. G. Walker, and S. Narayan. First-order incremental block-based statistical timing analysis. In *Proc. Design Automation Conf. (DAC)*, page 331. San Diego, CA, USA, jun 2004. [page 31]

[Wagne13] M. Wagner and H.-J. Wunderlich. Efficient Variation-Aware Statistical Dynamic Timing Analysis for Delay Test Applications. In *Proc. Design, Automation and Test in Europe (DATE)*. Grenoble, France, mar 2013.

[Wagne14] M. Wagner and H.-J. Wunderlich. Incremental Computation of Delay Fault Detection Probability for Variation-Aware Test Generation. In *IEEE European Test Symp. (ETS)*. Paderborn, Germany, may 2014.

[Wang06] L.-T. Wang, C.-W. Wu, and X. Wen. *VLSI Test Principles and Architectures: Design for Testability*. Morgan Kaufmann Publishers Inc., 2006. ISBN 978-0-12-370597-6. [pages 3 and 5]

[Wang09] Z. Wang and D. M. Walker. Compact Delay Test Generation with a Realistic Low Cost Fault Coverage Metric. In *Proc. VLSI Test Symp. (VTS)*, pages 59–64. Santa Cruz, CA, USA, may 2009. [page 45]

[Wang13] S. Wang, G. Leung, A. Pan, C. O. Chui, and P. Gupta. Evaluation of digital circuit-level variability in inversion-mode and junctionless FinFET technologies. *IEEE Transactions on Electron Devices*, 60(7):2186–2193, 2013. [page 2]

[Wothk93] W. Wothke. Nonpositive definite matrices in structural modeling. In *Testing Structural Equation Models*, pages 256–293. Sage Publications, 1993. ISBN 978-0-8039-4507-4. [page 66]

[Xie09] L. Xie, A. Davoodi, K. K. Saluja, and A. Sinkar. False path aware timing yield estimation under variability. In *Proc. VLSI Test Symp. (VTS)*, pages 161–166. Santa Cruz, CA, USA, may 2009. [page 44]

[Yalci97] H. Yalcin and J. P. Hayes. Event propagation conditions in circuit delay computation. *ACM Transactions on Design Automation of Electronic Systems*, 2(3):249–280, 1997. [page 39]

[Yamag13] T. J. Yamaguchi, J. S. Tandon, S. Komatsu, and K. Asada. A novel test structure for measuring the threshold voltage variance in MOSFETs. In *Proc. Int. Test Conf. (ITC)*. Anaheim, CA, USA, sep 2013. [page 21]

[Ye10] Y. Ye, S. Gummalla, C.-C. Wang, C. Chakrabarti, and Y. Cao. Random variability modeling and its impact on scaled CMOS circuits. *Journal of Computational Electronics*, 9(3):108–113, 2010. [page 104]

[Yilma08] M. Yilmaz, K. Chakrabarty, and M. Tehranipoor. Test-Pattern Grading and Pattern Selection for Small-Delay Defects. In *Proc. VLSI Test Symp. (VTS)*, pages 233–239. San Diego, CA, USA, apr 2008. [page 45]

[Zhang05] L. Zhang, W. Chen, Y. Hu, J. A. Gubner, and C. C.-P. Chen. Correlation-preserved non-Gaussian statistical timing analysis with quadratic timing model. In *Proc. Design Automation Conf. (DAC)*, pages 83–88. Anaheim, CA, USA, jun 2005. [page 131]

[Zhao06] W. Zhao and Y. Cao. New generation of predictive technology model for sub-45 nm early design exploration. *IEEE Transactions on Electron Devices*, 53(11):2816–2823, 2006. [page 2]

[Zhong11] S. Zhong, S. Khursheed, B. M. Al-Hashimi, S. M. Reddy, and K. Chakrabarty. Analysis of Resistive Bridge Defect Delay Behavior in the Presence of Process Variation. In *Proc. IEEE Asian Test Symp. (ATS)*, pages 389–394. New Delhi, India, nov 2011. [page 31]

Curriculum Vitae of the Author

Marcus Wagner studied computer science at the Friedrich Schiller University (FSU) in Jena and received his diploma degree in Computer Science (Dipl.-Inf.) in October 2007. He attained some industrial work experience while working with the company MAZeT GmbH in 2008.

In April 2009 he joined the institute of computer architecture and computer engineering (Institut für Technische Informatik, ITI) at the University of Stuttgart, under the direction of Prof. Dr. rer. nat. habil. Hans-Joachim Wunderlich. During this time, he contributed to the RealTest project and was working as a teacher assistant.

Publications of the Author

- M. Wagner and H.-J. Wunderlich. Incremental Computation of Delay Fault Detection Probability for Variation-Aware Test Generation. In *IEEE European Test Symp. (ETS)*. Paderborn, Germany, may 2014
- M. Wagner and H.-J. Wunderlich. Efficient Variation-Aware Statistical Dynamic Timing Analysis for Delay Test Applications. In *Proc. Design, Automation and Test in Europe (DATE)*. Grenoble, France, mar 2013

Declaration

All the work contained within this thesis, except where otherwise acknowledged, was solely the effort of the author. At no stage was any collaboration entered into with any other party.

_____

Marcus Wagner